# A Mixed Method Approach to investigate the Antecedents of Software Quality and Information Systems Success in Canadian Software Development Firms

**Delroy A. Chevers**
**Mona School of Business and Management, University of the West Indies, Kingston, Jamaica**
delroy.chevers@uwimona.edu.jm

**Abstract:** For years software developers globally have struggled in their attempts to deliver high quality software products. In an effort to overcome the problem, there is a widely held view that people, process maturity and technology are the main determinants of software quality and information systems (IS) success. The researcher conducted a survey in Canada among software developers, which confirmed the notion that people, process maturity and technology are determinants of software quality and also that software quality and user perception are determinants of information systems success. In an attempt to gain deeper insights, a series of interviews were conducted among Canadian developers to identify other contributing factors to software quality and information systems success. The interviews revealed that organization climate, user training and education, expectation management and technical support can influence the quality and success of the delivered software product. Such insights can enhance the competitiveness of these firms.

**Keywords:** Software quality, information systems success, organizational climate, process maturity, technical support, Canada

## 1. Introduction

For years many firms have been spending millions of dollars on software development projects to gain a competitive advantage (Benamati and Lederer, 2010; Montoni, Cerdeiral, Zanetti, and Da Rocha, 2008; Ponelis and Holmner, 2015), but in most cases they are not realizing the intended benefits. This is so because the software engineering community has been struggling in their attempts to deliver high quality software products (Niazi, 2012). High quality software product is defined as those with the requisite features that can add value to a business (Duggan, 2003). 71 percent of all delivered software products do not possess the requisite features to add value to respective businesses (Hastie and Wojewoda, 2015).

Scholars believe that process maturity, technology and people are major determinants of software product quality (Biro and Messnarz, 2009; Espinosa-Curiel, Rodriguez-Jacobo, and Fernandez-Zepeda, 2013). The competence and commitment by the people (developers, users and project managers) involved in the developmental process is critical to the effective development of software products. Likewise, the technology used and the maturity of the firm can positively impact the quality and success of the delivered software product.

Process maturity is defined as the degree to which the process is defined, managed, measured and continually improved (Dooley, Subra, and Anderson, 2001). It is believed that process maturity is the 'glue' that ties the triad (process, technology and people) together (SEI, 2005). The main tenet of this view is that higher levels of process maturity will increase the likelihood of producing higher-quality software products (Paulk, Weber. C.V., Curtis, and Chrissis, 1995). This view has led to the birth of software process improvement (SPI) programs (Humphrey, 1989). As a result, many software development firms have been adopting SPI programs in an effort to overcome the problem of delivering low quality software products.

Studies have demonstrated the benefits of SPI programs which include; reduced project cycle time, reduced project cost, improved staff productivity, improved customer satisfaction and improved software quality (Harter, Slaughter, and Krishnan, 2000; Krishnan and Keller, 1999; Sulayman, Urquhart, Mendes, and Seidel, 2012). The delivery of high-quality software products have been shown to enhance the likelihood of delivering successful software projects (Mudzana and Maharaj, 2017), that is those that are delivered on time, within budget and add business value.

Reference this paper: Chevers, D. A, 2018. A Mixed Method Approach to investigate the Antecedents of Software Quality and Information Systems Success in Canadian Software Development Firms. *The Electronic Journal Information Systems Evaluation,* 21(2), pp. 109-130, available online at www.ejise.com

However, despite the benefits, the adoption of SPI programs is low (Laporte and O'Connor, 2016). The main contributing factors to the low adoption is the belief that SPI implementation is costly, time consuming, disruptive and cumbersome (Laporte and O'Connor, 2016; Niazi, Babar, and Verner, 2010). Despite these beliefs, firms should endeavour to adopt SPI programs in an effort to deliver high quality software products since software process improvement is seen as a strategic resource that can provide a competitive advantage (Sulayman et al., 2012)

It is worth noting that there is no guarantee that the delivery of high quality software products will automatically lead to software project success. This is so because some technically sound and high quality software products are not being used due to users' perception (Newman and Robey, 1992). User perception regarding the ease of use, usefulness or threat of power loss can cause the non-use of a quality software product, which leads to the delivery of unsuccessful software projects. However, the delivery of high-quality software products usually increases the competitiveness of firms (Iversen and Ngwenyama, 2006). As a result, it can be argued that software quality and user perception can influence the success of software development projects.

This study adds to the literature on developing ways to implement successful software projects (Niazi, 2012). Additionally, the research focuses primarily on Canada since very few studies have been conducted in this country despite Canadian firms conducting more frequent training seminars on the importance of software engineering, and specifically software testing. These training seminars have been intended mainly to improve the quality of the delivered software product (Garousi and Zhi 2013).

As a result, the ultimate objective of this study is to ascertain the views of Canadian software developers regarding the state of software product quality and success in Canada. The research questions are therefore:

1. What are the views of Canadian software developers regarding the determinants of software product quality?
2. What are the views of Canadian software developers regarding the determinants of software project success?

A mixed method approach was taken because this approach is viewed as an appealing alternative to fulfil the diverse needs to investigative research (Razali et al., 2016). A mixed method sequential explanatory research design was taken (Creswell, 2009). A survey was conducted to evaluate the determinants of software quality and software project success in Canada. Subsequently, interviews were conducted with Canadian software developers to seek deeper insights regarding the survey findings and the identification of other factors that could impact software quality and software project success.

It is hoped that the findings of the research can provide useful insights to executives desirous of developing and delivering high-quality software products. This can lead to firms winning global contracts, which by extension can lead to economic development.

## 2. Literature Review

Software development is considered a human intensive activity (Clarke and O'Connor, 2013), because it involves the interaction of many individuals with various experiences, expectations and knowledge. As a result, software development is considered complex (Casey and Richardson, 2009; Clarke and O'Connor, 2013). This complexity has manifested itself in a high failure rate of software development projects (Standish Group, 2015) with the  current failure rate of these projects at 71% (Hastie and Woj 2015). According to the (Standish Group, 2013), a failed project is one that is cancelled prior to completion or delivered but never used. Many high-quality software are not used because they are perceived as not being useful and easy to use (Newman and Robey, 1992).

The delivery of high quality software products is ranked highly among IS executives (Luftman et al., 2012), because unused or underutilized software products can cost firms millions of dollars (Markus and Keil, 1994). Furthermore, the delivery of poor quality software products can negatively impact firms' performance as well as their competitiveness (Iversen and Ngwenyama, 2006).

For years the software engineering community has been experimenting with various software development methods in an attempt to deliver higher-quality software products. This evolution has seen the emergence of the traditional waterfall, incremental, cleanroom, component-based, user-centered and agile methods. Software engineering is a discipline that is concerned with all aspects of software development from the early stages of system specification through to maintaining the system after it has gone into use (Sommerville, 2011). Its goal is to delivery software products on time, on budget, with acceptable performance and with correct operation (Sommerville, 2011).

However, the discipline has been criticized as being inadequate for modern software development (Sommerville, 2011). Software development is a complex activity (Clarke and O'Connor, 2013). As a result, a socio-technical approach should be taken in software development (Mumford, 2000). This takes into account technical, organizational, economic and social factors in software development and delivery (Mumford, 2000). Software in this study is defined as computer programs and associated documents (Sommerville, 2011).

It is widely accepted in the software engineering community that the three major determinants of software quality are people, technology and process maturity (Biro and Messnarz, 2009; Espinosa-Curiel et al., 2013). However, some researchers believe that process maturity is the most influential of the three determinants (Humphrey, 1989; SEI, 2010). One proponent of this school (SEI, 2005) in articulating the role of process states that "everyone realizes the importance of having a motivated, quality work force and the latest technology, but even the finest people cannot perform at their best when the process is not understood or operating at its best." (p. 9). This view has led to the birth of software process improvement (SPI) programs in which firms are assessed regarding the maturity of their process, which then leads to a plan of action for process improvement. There are five levels of maturity and firms seek to achieve level 5 where the highest level of competence exists and a culture of continuous improvement prevails. The literature supports the notion that insufficient attention to process maturity can negatively affect firms' ability to provide high-quality software products (Krishnan and Keller, 1999).

Studies have demonstrated the benefits of SPI programs (Dooley et al., 2001; Harter, Slaughter, and Krishnan, 1998; Krishnan and Keller, 1999). However, the adoption of these programs are low (Espinosa-Curiel et al., 2013; O'Connor and Laporte, 2014) with 67% of firms that attempt to adopt any form of SPI program usually abandon the effort before realizing its potential benefits (Krasner, 2001). The contributing factors to the low adoption and abandonment is the belief that the implementation of these programs are costly, time consuming, disruptive and cumbersome (Buchman and Bramble, 1995; Laporte and O'Connor, 2016; Ngwenyama and Norbjerg, 2010; Staples et al., 2007).

There are two distinct schools of software development approaches, with the ultimate goal being the delivery of high quality software products. The two approaches are agile and structured methods. Within the structured approach, process maturity is seen as the main determinant of software quality and firms' processes are evaluated by process assessment, which in turn leads to the determination of process capability and an action plan for process improvement (Bicego, 1996; Humphrey, 1989).

To be competitive and win global contracts software development firms must demonstrate that their software development process is mature (Espinosa-Curiel et al., 2013; Oktaba, 2006; Pino, Garcia, and Piattini, 2008; Sulayman et al., 2012). The rationale is that high levels of maturity can increase the likelihood of producing high quality software products (Humphrey, 1989).

The capability maturity model integration (CMMI) is a process improvement framework that is used to measure process maturity and provides a reference point for planning SPI initiatives (SEI, 2010). Globally, it is the most popular framework used in assessing the capability and maturity of firms (Laporte and O'Connor, 2016). Statistics regarding the assessment of firms' maturity shows only 1.7% being appraised at level 4 and 6.2% being appraised at level 5, with 63.3% (the large majority) being assessed at level 3 (Keller and Mack, 2013). This means that most firms are not operating at level 5, where process maturity practices are institutionalized. Another report shows a similar trend where 48% (the majority) of Canadian firms are appraised at level 3 and only 5% being appraised at level 4 (Carnegie Mellon, 2013).

Quite recently, many firms have been recognizing that the CMMI is becoming an effective medium for achieving agile performance – the second school of software development (Dalton, 2016). Based on this new development, there is a recent trend for CMMI firms to implement agile methods and vice versa (Dalton, 2016).

Within the agile school of software development, the emphasis is on the capability and creativity of the people involved in software development (Cockburn and Highsmith, 2001). The two most popular methods in the school are Scrum and Extreme programming (Dalton, 2016). These methods require users to actively participate in the development and delivery of the software product throughout the project life cycle (Dingsoyr, Nerur, Balijepally, and Moe, 2012). Firms are increasingly being pressured to develop new or enhanced software products quickly, and so agile methods focus on facilitating early and fast development and delivery of software products in an iterative manner (Turk, France, and Rumpe, 2005). The general tenet of this school as posited by (Cockburn and Highsmith, 2001) is that "if the people on the project are good enough, they can use almost any process and accomplish their assignment. If they are not good enough, no process will repair their inadequacy" (p. 131). The main objective of the methods within this school is to develop high-quality software products as quickly as possible with the hope of providing business value (Balijepally, Mahapatra, Nerur, and Price, 2009).

These methods seek to improve the relationship between developers and users and as a result, they are gaining popularity (Ahmed, Ahmad, Ehsan, Mirza, and Sarwar, 2010; Ceschi, Sillitti, and Succi, 2005). Not only do developers and users work closely but an iterative approach is taken in the development of software products. As such applicable and relevant software prototypes are quickly produced and tested and acceptance sought within short time spans or sprints (Dalton, 2016). These methods are believed to be more responsive to environmental changes, in an industry that is very volatile (Boehm, 2002).

Studies have found that most information systems projects fail mainly because of the interaction and conflict among people – project members (Barrett and Oborn, 2010; Ceschi et al., 2005). These conflicts increase the risk of project failure (Boehm, 2002). The risks that are applicable in this study are project and functionality risk (Sherer and Alter, 2004). Project risk is concerned with the project being on time, within budget and realizing intended quality specifications, while functionality risk is concerned with the delivery of quality specifications and adding business value (Sherer and Alter, 2004).

People are an organization's most important assets (Sommerville, 2011). There is the notion that software development can be improved through the participation of the people who will eventually use the delivered software product (Mumford, 1978). Given the power and authority vested in people in the agile school, the literature supports the notion that agile methods can promote the success of software development projects (Standish Group, 2015).

Based on the arguments put forth between the structured and agile approaches to software development, the difference is the structured seeks to create stability and less variability in the developmental process, while agile seeks to create an atmosphere of responsiveness in a volatile environment.

In an effort to achieve both stability and responsiveness many firms are embracing the combination of agile and structured approaches to software development (Dalton, 2016). It is said that the structured approach provides a road map of 'what' must be done, whereas the agile approach prescribes 'how' it must be done (Dalton, 2016). This new trend of combining the structured and agile approaches has been reaping various benefits and success. It was noted in the literature that Minacs IT Services, an outsourcing company with headquarters in the US, Canada and India, who embraced both structured and agile methods experienced a 30% - 40% increase in attaining a agile technique called sprint, a 30% increase in the number of user stories delivered in each sprint and a 40% increase in on time delivery (Dalton, 2016). Similarly, a global high technology company – Perficient Chennai – was able to reduce software defects by 70% by adopting both structured and agile methods (Dalton 2016). Hence, the integration of both approaches is a growing trend.

The trend of many firms embracing both structured and agile methods of software development aligns with the institutional theory of organizational growth. The theory focuses on firms' desire to grow and increase their capability and survive in a competitive environment by satisfying their stakeholders (DiMaggio and Powell, 1983). In their pursuit to survive and strive, firms develop value-creation skills and seek to become similar in an effort to fit into the industry. In trying to 'fit-in' firms normally conform and adopt established rules and practices within the industry, as well as mimic the best practices of industry leaders (DiMaggio and

Powell, 1983). In an effort to become similar, some firms even recruit employees from their competitor who possess certain competences like business analysis, software coding and/or project management skills. This recruitment and selection strategy is relevant because the retention rate of software developers is low (Westlund and Hannon, 2008). When experienced and knowledgeable software developers resign, they usually depart with critical knowledge of good business processes, procedures and practices that are essential in sustaining a competitive advantage (Westlund and Hannon, 2008).

In addition to the scarcity of knowledgeable software developers in some firms, there is the situation of SPI program abandonment. 67% of firms who attempt to implement SPI programs usually abandon the effort before realizing the potential benefits (Krasner, 2001). The strategies of adopting the best practices in the industry and recruiting experienced and knowledgeable developers from competitors are described as mimetic and normative isomorphism in institutional theory. Finally, this trend of firms embracing both structured and agile methods in software development is becoming popular because firms want to grow, develop and be similar.

A research model was developed based on the arguments in the literature regarding the relationship between the software quality and information systems success variables. It was stated that technology, people and process maturity are major determinants of software quality (Sulayman et al., 2012), and that software quality and user perception are determinants of IS success (Halilovic and Cicic, 2013; Petter, DeLone, and McLean, 2013; Zheng, Zhao, and Stylianou, 2013). Figure 1, the research model, shows the result of these associations in which technology, people and process maturity are seen as antecedents to software quality. And further, software quality and user perception are shown as antecedents to software project success.

Five hypotheses were derived and tested in this study. They include:

H1: The institutionalization of process maturity practices will have a positive impact on software quality  
H2: People skills and contribution will have a positive impact on software quality  
H3: The efficient utilization of technology will have a positive impact on software quality  
H4: Software quality will have a positive impact on software project success  
H5: User perception will have a positive impact on software project success

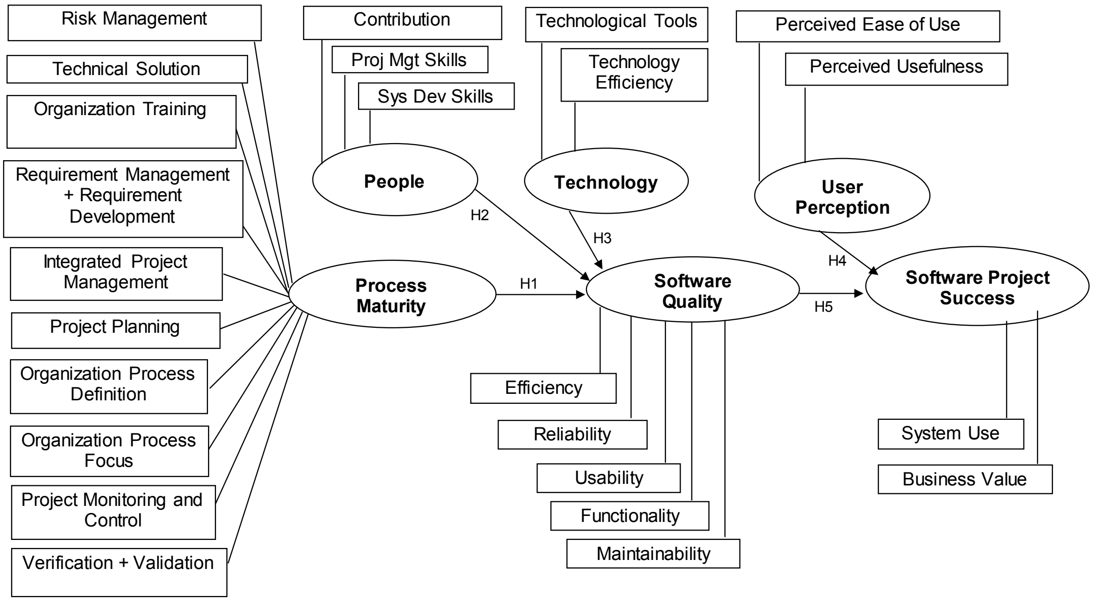

**Figure 1:** The Research Model

## 3. Methodology

### 3.1 The Survey

The study took a mixed method sequential explanatory research design (Creswell, 2009). The survey items for process maturity, user perception and success were taken from the literature, in which reliabilities were established (Lewis, 1995; McManus, 1999; SEI, 2010; Zubrow, Hayes, Siegel and Goldensen, 1994). The survey items for the people and technology constructs were newly developed. As a result, a pre-test was conducted on these items in order to establish reliability. Seventeen information systems researchers, methodologists and graduate students in information systems in Jamaica and Trinidad and Tobago were selected to assist with the pre-test. Upon completion of the pre-test, suggestions were made regarding the time to complete the survey and the identification of ambiguous or compound questions. The relevant adjustments were made based on the feedback. The survey items were anchored on a 7-point likert-type scale, with 1 being strongly disagree and 7 being strongly agree (see Appendix A – The survey instrument). Ethical approval was given by the Ethics Committee at Carleton University, Canada.

The sample frame was developed from the Canadian Company Capabilities Database, as well as personal referrals with the unit of analysis being the software development project. For the actual survey, an online approach was taken where questionnaires were emailed to potential respondents. A total of 344 email invitations were sent but only 61 responses were collected, with 6 being incomplete. Hence, only 55 responses were analysed which equates to a response rate of 16%.

Of the 55 respondents in the survey, eleven were females and forty-four males. These respondents were chief information officers (CIOs), analysts, programmers, developers, administrators and project managers. 22% of the survey respondents were CIOs, and the average years of service in the software industry was 15.2 years. Notably, the survey respondents were somewhat skewed towards more senior personnel in the industry with the vast majority occupying senior positions within the industry.

### 3.2 The Interview

Telephone interviews, email communication and face-to-face site visits were used in this study. The main focus of these interview sessions was software developers/project managers' views regarding the rationale of the survey findings. A total of seven interview questions were prepared as shown in Appendix B. All respondents were asked these seven questions, irrespective of the method used to collect the data (whether telephone, email or face-to-face). In addition, the purpose and findings of the survey, the established CMMI practices at all five maturity levels, as well as the definition of key terms like risk management, technical solution, etc. were given or explained to respondents, irrespective of the approach taken.

In the case of the face-to-face interviews, permission were sought to record the session, after which the responses were transcribed. A similar approach was taken with the email interviews in which respondents were asked to submit their responses in writing on the interview question sheet. These responses were later transcribed. For the telephone interviews, respondents were asked the same seven questions and their responses were transcribed. All responses were compiled for analysis.

Altogether a total of 64 invitations/requests were sent, but only 25 interviews were conducted (telephone, email and face-to-face). This gave a 39% response rate. The respondents were 23 male and 2 female Canadian software developers who worked in Ottawa, Toronto and Montreal. Proximity to the researcher and convenience were the basis for selecting these cities. The majority of the respondents held at least a bachelor degrees in computer science or information systems.

Content analysis was the technique used to conduct the analysis, in which various central themes were sought for each of the seven interview questions. Although content analysis do not deal with people directly, the technique studies recorded human communication (Downe-Wamboldt, 1992). And in this study it was important to analyze the central themes of human communication on issues relating to quality software products being delivered and the success of software development projects in a specific context - Canada.

The process taken to analyze the responses was:

- Identify the research questions
- Define how the variables can be measured/assessed
- Create a coding scheme
- Record the frequencies of codes
- Identify central themes

A mixed method approach was taken in this because based on the $R^2$ values, deeper insights were needed to ascertain the other factors (determinants) which could impact software product quality and software development projects. The results of the survey and the interviews are shown in the next section.

## 4. Findings and Discussion

### 4.1 The Survey:

Partial least squares (PLS) was used as the statistical tool to test the research model. It is a popular structural equation modelling (SEM) technique (Chin, 2010) and was used to evaluate both the measurement and structural models. The bootstrap re-sampling method of 500 re-samples was used to determine the path significance of the structural model. PLS-Graph was selected as the statistical tool based on its ability to handle small sample sizes (Chin, 2010).

Table 1 provides details on the descriptive statistics, as well as the reliability and validity of the survey items. In reviewing the mean scores, all readings were above the mid-point (on the 1 – 7 scale), except one of the process maturity practices - organization process focus which had a mean score of 3.764. In looking closer at the descriptive statistics of the process maturity practices, requirement management + requirement development was ranked the highest (mean = 5.041, SD = 1.914), with organization process focus being the lowest ranked (mean = 3.764; SD = 2.197).

**Table 1:** Descriptive Statistics, Composite Reliability and Average Variance Extracted

| **Variable (# of Items)** | **Description** | **Mean (n = 55)** | **SD (n = 55)** | **Composite Reliability (CR)** | **Average Variance Extracted (AVE)** |
|---|---|---|---|---|---|
| **Process Maturity** | | | | | |
| RSKM (4) | Risk Management | 4.245 | 1.864 | 0.960 | 0.858 |
| TS (3) | Technical Solution | 4.739 | 1.794 | 0.937 | 0.832 |
| OT (3) | Organization Training | 4.770 | 1.965 | 0.928 | 0.765 |
| RMRD (4) | Requirement Management + Requirement Development | 5.041 | 1.914 | 0.933 | 0.777 |
| IPM (3) | Integrated Project Management | 4.533 | 1.968 | 0.930 | 0.817 |
| PP (5) | Project Planning | 4.931 | 1.946 | 0.960 | 0.828 |
| OPD (3) | Organization Process Definition | 4.000 | 2.104 | 0.948 | 0.859 |
| OPF (3) | Organization Process Focus | 3.764 | 2.197 | 0.981 | 0.944 |
| PMC (4) | Project Monitoring and Control | 4.895 | 1.919 | 0.965 | 0.875 |
| VV (4) | Verification + Validation | 4.736 | 1.809 | 0.963 | 0.867 |
| People | | | | | |
| CONTR (3) | Contribution | 5.630 | 1.380 | 0.936 | 0.830 |
| PMSKL (2) | Project Management Skills | 5.336 | 1.570 | 0.925 | 0.860 |
| SDSKL (5) | Systems Development Skills | 5.451 | 1.582 | 0.938 | 0.755 |
| Technology | | | | | |
| TechT (2) | Technological Tools | 4.309 | 2.106 | 0.872 | 0.773 |
| TechEff (2) | Technology Efficiency | 4.200 | 2.149 | 0.887 | 0.796 |
| Software Quality | | | | | |
| MAINT (3) | Maintainability | 4.824 | 1.506 | 0.898 | 0.746 |
| RELIA (3) | Reliability | 4.945 | 1.936 | 0.920 | 0.794 |
| EFFIC (2) | Efficiency | 5.191 | 1.565 | 0.917 | 0.847 |
| FUNCT (3) | Functionality | 4.867 | 2.047 | 0.915 | 0.782 |
| USABI (3) | Usability | 5.242 | 1.298 | 0.959 | 0.888 |
| Perception | | | | | |
| PU (3) | Perceived Usefulness | 4.509 | 2.103 | 0.982 | 0.948 |
| PEU (3) | Perceived Ease Of Use | 4.624 | 1.949 | 0.959 | 0.887 |
| IS Success | | | | | |
| SysUse (3) | System Use | 5.006 | 1.856 | 0.824 | 0.610 |
| BusVal (3) | Business Value | 4.503 | 2.171 | 0.944 | 0.849 |

The overall results as shown in Table 1 with the mean scores for the Process Maturity construct ranging from 3.764 to 5.041, suggests that most software development firms in Canada have some form of software processes in place and these process maturity practices were reasonably established (scale 1-7). Nevertheless, based on the low loading of 0.6585 (< 0.70) for organization process focus in Table 2 gives rise to the need for greater awareness with respect to the understanding and institutionalization of this variable. As distilled by (Luftman et al., 2012; SEI, 2010) organization process focus is concerned with the deployment of process improvement based on a thorough understanding of the firm's strengths and weaknesses. A series of training seminars with key users defining the role and importance of each process maturity variables would be useful to increase the awareness, adoption and ultimate benefits of these variables. This strategy supports the belief regarding assisting users to become competent in the management of proposed organizational change (Mumford, 2000).

Both software quality and software development project success constructs were found to be reasonably established in Canadian software development firms. The readings for software quality are high (mean =5.242, SD = 1.298) and low (mean = 4.824, SD = 1.506) and software development project success high (mean = 5.006, SD = 1.856) and low (mean = 4.503, SD = 2.171).

Table 1 also shows that both reliability and validity were established in analyzing the measurement model. The composite reliability readings ranged from 0.824 – 0.982, which is above the acceptable threshold of 0.70 (Chin, 2010). Likewise, the average variance extracted (AVE) scores ranged from 0.610 – 0.948, which is above the acceptable threshold of 0.50 (Fornell and Larcker, 1981). These two measures are indicators of the reliability and validity of the survey items, and all readings are above the threshold.

**Table 2:** Loadings of Variables

| Construct | Indicator Variable | Description | Loadings |
|---|---|---|---|
| Process Maturity | RSK (4) | Risk Management | 0.8526 |
| | TS (3) | Technical Solution | 0.9411 |
| | OT (3) | Organization Training | 0.7346 |
| | RM (4) | Requirements Management + Requirements Development | 0.8745 |
| | IPM (3) | Integrated Project Management | 0.8289 |
| | PP (5) | Project Planning | 0.9151 |
| | OPD (3) | Organization Process Definition | 0.7300 |
| | OPF (3) | Organization Process Focus | 0.6585 |
| | PMC (4) | Project Monitoring and Control | 0.9059 |
| | VV (4) | Verification + Validation | 0.8685 |
| People | CON (3) | Contribution | 0.9104 |
| | PMS (4) | Project Management Skills | 0.8991 |
| | SDS (3) | Systems Development Skills | 0.9327 |
| Technology | TT (2) | Technology Tools | 0.9732 |
| | TE (2) | Technology Efficiency | 0.9771 |
| Software Quality | MA (3) | Maintainability | 0.8393 |
| | REL (3) | Reliability | 0.8655 |
| | EFF (2) | Efficiency | 0.8351 |
| | FUN (3) | Functionality | 0.8566 |
| | USA (3) | Usability | 0.8021 |
| Perception | PU (3) | Perceived Usefulness | 0.9466 |
| | PEU (3) | Perceived Ease of Use | 0.9313 |
| Information Systems Success | SysUse (3) | System Use | 0.9120 |
| | BusVal (3) | Business Value | 0.9379 |

All variables in this study were modelled as reflective and as such their loadings are important. Table 2 shows that the item loadings were above the recommended threshold of 0.70 (Chin, 2010), except again for the process maturity practice – organization process focus with a loading of 0.6585. However, this variable was retained in the research model because it was derived from the literature.

In analysing the structural model, it is shown in Figure 2 that all five hypotheses (H1, H2, H3, H4 and H5) were supported. The construct 'People' had the greatest impact on software quality, when compared to process maturity and technology. With a $R^2$ of 0.76 it means that collectively, people, process maturity and technology accounted for 0.76 of the variance in software quality. Equally important is the finding that software quality and user perception accounted for 0.781 of the variance in software project success ($R^2$ = 0.781), with both software quality and user perception being significant at 99%. These findings are consistent with prior studies in which it was found that mature firms are more inclined to implement successful projects, people management is important for process improvement, and people involvement and people skills are critical in quality improvement (Batista and de Figueiredo, 2000; Johansen and Pries-Heje, 2007; Niazi and Babar, 2009; Rainer and Hall, 2002). In fact, software development can be improved through the participation and involvement of the people who will ultimately use the software (Mumford, 2000).

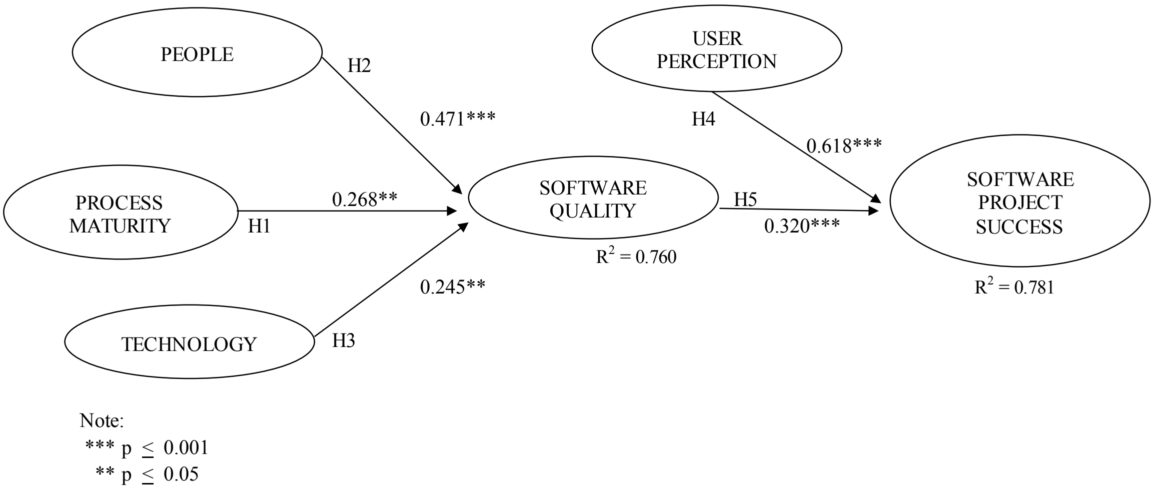

**Figure 2***:* The Hypothesized Model

## 4.2 The Interview:

*The responses to question #1 which seeks to explain why people skills and contribution had the greatest impact on software quality over technology and process maturity in Canada are as follows:*

The central theme to this question is that human resource skills, effective communication and decision making abilities are key factors that can enhance the development and delivery of high quality software products. It was further stated that in general people take along with their skills set, knowledge and experience into the developmental process and these traits are essential throughout the project life cycle. It was clearly stated by a few respondents that knowledge sharing among project team members and key stakeholders is more important than the adoption of the latest technology and the maturity of any development process. One respondent was quoted as saying “what is important is developer skills in software development to implement correctly and perform the relevant verification and validation process”. Studies have shown that social factors like cultural barriers, user rejection and resistance to change are more influential than technical and cost factors in the successful implementation of software projects (Merlo, Carneiro, and Araujo, 2016). In fact, the literature supports the notion that motivated and capable project team members are likely to implement successful process improvement initiatives with the potential benefits being realized (Montoni et al., 2008).

Furthermore, it was felt that if developers have good people and communication skills, they would be better able to understand users’ requirements and therefore develop more effective solutions, which by extension could manifest in the production of higher quality software products. The requirements engineering stage involves a clear understanding of the requirements and specifications of the intended software product (Kotonya and Sommerville, 1996). It is widely accepted that the most important stage in software development is the definition of the requirements, otherwise called requirements definition (Mumford, 1985). However, it is posited that the main problem in software development is the difficulty with requirements definition (Kotonya and Sommerville, 1996). As a result, it is imperative that software developers and users of the new software product, work closely together and clearly define and understand what they want the proposed software product to do (Mumford, 1985). If the requirements of the software are clearly defined, then users will have a better idea of what they want the new software to help them achieve from a personal level and also at the business value level.

However, the gap between the delivered software and the intended needs are usually high and its occurrence is usually common; and the cost to address this gap is very high (Kotonya and Sommeville, 1996). In some cases, the gap is so wide in which the delivered software do not deliver the expected support for the real work in the organization (Baxter and Sommerville, 2011), this can shatter users' expectation which by extension can result in non-use of the delivered software. Non-use leads to unsuccessful delivery of the software project. Hence, effort should be taken to manage the expectation of users in an attempt to increase the likelihood of success.

It was clearly stated by one respondent that generally in Canada, project team members are willing to contribute and take responsibility for the development and delivery of high quality software products. It was also pointed out that those developers who can interpret and integrate both the ways the technology can be applied as well as understand the needs of the user/client, will be more equipped to produce better results at the completion of the project. It was also believed that in the absence of established practices, good people tend to formulate their own tools and practices in an effort to maximize productivity during system development, especially in a very competitive industry.

*In terms of question #2 which seeks to address the finding that technology had the lowest impact on software quality in Canada. The following were revealed:*

Although new technologies are changing rapidly and becoming available quickly, the central theme for this question, is that technology is a secondary tool. The intellectual interpretation and application of the new technology is more important than the technology itself. It is the application of the technology which can impact the quality of the delivered software. In addition, respondents believed that many Canadian software development companies (in a developed country) can afford to purchase the latest technology. As a result, technology should not be seen as a strategic weapon that can provide a competitive advantage by these firms in Canada. Furthermore, it was claimed that individuals (developers, users and project managers) can easily learn how to navigate and utilize the latest technology, because such knowledge is usually explicit, that is, knowledge that can be codified into instructions that can be followed easily. In contrast, some knowledge is tacit, especially those of key users (Bano and Zowghi, 2015). Users usually possess specific knowledge about the business processes, the work practices, the tasks that they perform and the context that guide various practices (Bano and Zowghi, 2015). In many cases these rules and practices are tacit in nature and as such difficult to codify (Bano and Zowghi, 2015). Based on the tacit nature of such knowledge, they provide a competitive advantage. However, the instructions to navigate through newly developed technologies, can be mastered by the average individual.

*In responding to question #3 about why the variance explained by people, process maturity and technology on software quality [$R^2$] was 0.760 or 76%. The following were revealed:*

Even though the $R^2$ could be considered high, it is in alignment with the literature which states that people, process maturity and technology are the major determinants (antecedents) of software quality. Interestingly, other contributing factors (antecedents) were suggested by the respondents. These included the size of the firm, the size of the project, the size of the user-community, as well as various organizational climate and cultural factors such as top management support, the strength and dynamics of the project members, established business processes and finally the structure of the organization.

The level of competition, time to market, bargaining power of clients/users, demand for new products, demand for quality products by clients/users, social trends, and regulatory guidelines and changes were other less frequent factors that were suggested. Some of these suggested factors could be classified as control variables, while others could be viewed as antecedents to software quality. The designation of these variables were not defined in the responses, but the researcher modelled the organizational climate construct as an antecedent to software quality (as shown in Figure 3); with top management support, dynamics of the project members, established business processes and structure of the organization being the indicator variables. It was felt that the size of the firm, size of the project and size of the user-community be used as control variables in the study.

In addition, it was stated that the desire to meet stringent deadlines in many cases could result in little or no testing being done. This oversight could result in the development and delivery of poor quality software products. Furthermore, the sale cycle of emerging software for firms who develop software for sale, as well as cost cutting measures could also impact quality. However, it is felt that these factors could be at a minimum in recent times because:

1. Canada is paying a lot of attention to the importance of software testing and training (Garousi and Zhi, 2013)
2. Firms seek to become similar by adopting the best practices in the industry or adopt the best practices of a very successful firm within the industry. This practice is in alignment with the institutional theory (DiMaggio and Powell, 1983).

*Question #4 explored why perception had a greater impact on software project success than software quality in Canada. The following were revealed:*

There is the notion that many technically sound and high-quality software products are not being used due to user perception (Newman and Robey, 1992). The perception of users regarding the usefulness, ease of use or the potential of the software product to reduce or replace users' role and significance in the firm, can negatively impact utilization and the success of the project. The central theme was that no matter how high developers rate the quality of the delivered software, quality is relative to the users' requirements, desires and preferences. Many users gravitate to software that are useful, add business value and user-friendly with an excellent user interface, and not necessarily the best computer system as determined by the developer. As a result, it was recommended that the user community needs to participate in the design and development of the system more closely in an effort to reduce misunderstanding and increase the likelihood of delivering high-quality software products. This interaction of software developers and users should be encouraged from the beginning to the end of the system development life cyle, but especially at requirements definition. This approach is in alignment with the user-centered and agile software development methods.

*Question #5 looked at the fact that the variance explained by perception and software quality on software project success was 0.781 or 78%. The following were revealed:*

A $R^2$ of 0.781 means that user perception and software quality explains 78% of the variance in software development project success. Other contributing factors to software project success that were suggested were user training and education, technical support and the management of user/client expectations. In fact, it was claimed by one respondent that prolonged technical support by developers can enhance the likelihood of successful software projects. It was suggested that future research could model technical support and the management of user/client expectations as variables for the software development project success construct. The respondents did not specify whether the variables should be modelled as antecedents, mediating variables or moderating variables. However, the researcher felt that the suggested factors should be modelled as moderating variables. Moderation takes place when the effect of an independent variable on a dependent variable varies according to the level of a third variable (Edwards and Lambert, 2007). In this case the third variables are technical support and expectation management impacting software quality and software project success. As a result, these two variables could be incorporated amongst software quality and IS success in the revised research model (as shown in Figure 3).

A major problem being faced by many organizations is implementing and managing change to the delivered software product (Sommerville, 2011). The changing business environment means that quite regularly new requirements might emerge, as well as existing requirements might change. In response, the newly delivered software product must continually evolve in accordance to the changes in the business community, or else it will become outdated with no business value. In addition, clients are demanding faster delivery of software products, at lower cost and those that are more responsive to changing requirements. As a result, the current trend is to adopt an incremental or agile software development method. The highest priority of these methods is to satisfy the client/user through early and continuous delivery of valuable software (Ahmed et al., 2010).

The respondents also made reference to developers' level of education, training, and experience as an additional antecedent to the 'Software project success' construct. These factors (education, training and experience) are relevant because it is stated that software quality assurance and software process improvement courses are currently being taught in many Canadian universities (Laporte and O'Connor, 2016). A well trained and qualified employee is a valuable asset to an organization. This claim is embedded in the

philosophy of agile software development methods – “If the people on the project are good enough, they can use almost any process and accomplish their assignment. If they are not good enough, no process will repair their inadequacy” (Cockburn and Highsmith, 2001, p. 131). Based on these arguments it would be good if these new variables could be tested and validated in the revised research model.
*Question #6 sought answers to the question regarding why the loading on ‘Organization Process Focus’ was below the 0.700 threshold. The following were revealed:*

Organization process focus (OPF) centers around plans to deploy process improvements based on a thorough understanding of the strengths and weaknesses of the organization’s software development processes (SEI, 2010). It assists in the planning of process improvement, deploying of organizational resources and the sharing of project team members’ experiences. In essence, careful planning is required to ensure that process improvement is adequately managed. In an effort to seek clarity and a common understanding, the definition of the term – OPF – as distilled by SEI (2010) was shared with all respondents irrespective of the communication method employed (telephone interviews, face-to-face site visits or email).

The central theme emerging from this question was that organization process focus was a high level 3 CMMI practice. It was felt that the level 2 practices, which were more project management practices, can have a more direct impact on software quality. The level 2 CMMI practices included project management practices such as requirements management, software project planning and software project monitoring. However, one respondent was quoted as saying “too much focus on process kills innovation”.

In general, there is usually a reluctance by employees to adopt new practices whether at the project management or organizational levels. Resistance to change is a common phenomenon in organizations. In addition, this tendency by employees can be further compounded by top management, who sometimes lack the requisite understanding regarding what is required in implementing process improvement programs. Furthermore, many top executives are more focused on the bottom line rather than process improvement.

The literature supports the notion that the implementation of SPI programs can be costly, time consuming, disruptive and cumbersome (Espinosa-Curiel et al., 2013; O'Connor and Laporte, 2014). As a consequence many firms might sacrifice compliance to process maturity practices in an effort to improve their responsiveness to time-to-market. Time-to-market is a very important performance measure in the software industry (Jeong and Yoon, 2016). A measure that can be used to gain a competitive advantage.

*The final question (question #7) was open-ended and asked respondents to reflect and share any observations or insights which they garnered from the study. The following were revealed:*

The study supports the notion that software process improvement (SPI) program positively impacts software quality and software project success. However, it is important to note that a reasonable balance is needed among people, process maturity, technology and user perception because all factors were found to have a positive impact on software quality and software project success. The central theme emerging from this question is that firms must create a culture of knowledge sharing, conduct user training and education, perform technical support and manage the expectation of key stakeholders. It is felt that all these conditions are attainable because most Canadian firms place a lot of emphasis on training and development. It was observed by one respondent that the age of the firm, the maturity of the firm and the size of the firm should be used as control variables in an attempt to increase the richness of the research findings.

Figure 3 – the revised research model was derived based on the views presented by the respondents. A fourth construct called ‘organization climate' was added to the proposed revised research model. Hence, the key determinants of software quality are being proposed to be people, process maturity, technology, and organization climate. In addition, technical support and expectation management are being proposed as moderating variables between software quality and software project success while user training and education is being proposed as an antecedent to software project success. As a result, the overall antecedents to software project success are software quality, user perception and user training and education. It is hoped that this proposed research model will be validated in future studies.

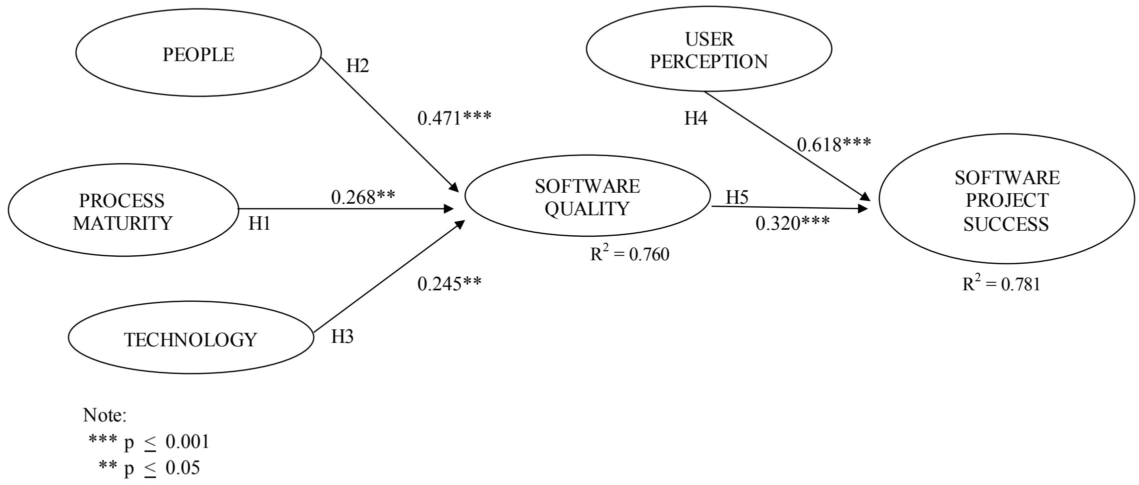

**Figure 3:** The Revised Research Model

## 5. Conclusion

The study confirm the notion that people, process maturity and technology are key determinants (antecedents) of software quality and also that user perception and software quality are key determinants of software development project success. The survey discovered that developer skills and contribution had the greatest impact on software quality, over process maturity and the application of the latest technology. Technology is believed by respondents in the interview session to be a secondary tool which can be acquired by most Canadian software development firms but the ability of developers and users to clearly articulate what is required and what should be delivered is critical in developing high quality software products. This argument supports the notion that requirements definition is a very important stage in the software development life cycle.

The importance of requirements definition has influenced the current trend in which firms are merging both agile and structured methods of software development. This merger is in alignment with the user-centered approach in software development, and the concept of early and continuous delivery of valuable software products. In response to this new trend of the merger of agile and structured methods and the fact that some knowledge is tacit, some organizations are recruiting knowledgeable developers and users from their competitors. This strategy is described as normative isomorphism in the institutional theory and if applied correctly, can allow firms to strive and gain a competitive advantage.

The survey also found that although both user perception and software quality had a positive impact on software development project success, user perception had the greater impact. This finding is consistent with the literature which states that many technically sound and high-quality software products are not being used (Newman and Robey, 1992). The perception of users regarding the usefulness, ease of use and value-adding capability, is a critical component in the adoption and utilization of the delivered software product.

On the other hand, the interview sessions revealed that organization climate could be considered a determinant of software quality. In this study, organization climate includes top management support, the strength and dynamics of the project members, established business processes, and the structure of the firm. The interviews also revealed that technical support and expectation management could be considered as moderating variables between software quality and software development project success, and user training and education be treated as a determinant of software project success. It is strongly believed that if the suggested factors are applied then higher-quality software products and more successful software project can be produced, which by extension can increase the competitiveness of these firms.

To be competitive and win global contracts, software development firms must demonstrate that their software development process is capable and mature. In fact, firms must be assessed regarding their process maturity level and found to be at level 3 and above to be qualified for global contracts, especially for US Government contracts. The mixed method approach taken in this study provides a means to get richer insights about the antecedents and moderating variables of process maturity, software quality and software project success.

Pressure by clients/users for quality software cause continuous process improvement to be a necessity. This drive is also important as Canadian software development firms seek to re-establish their capability and produce more successful projects. A practical implication of the study is the possibility of other researchers to validate the proposed research model and further refine it. Other factors such as culture and leadership style could be incorporated into the research model for future assessment. In addition, it is being suggested that demographic data like the age of the firm, the maturity of the firm and firm size be used as control variables in future research to provide greater insights.

The main limitation of this study is the small sample size, which limits the generalizability of the results. In an effort to overcome this limitation, future research should increase the sample size. It is hoped that this stream of research can contribute to the growth and development of the software development industry in Canada.

## References

Ahmed, A., Ahmad, S., Ehsan, N., Mirza, E., and Sarwar, S. Z., 2010. *Agile software development: Impact on productivity and quality.* Paper presented at the 5th IEEE International Conference on Management of Innovation and Technology, Singapore.

Balijepally, V., Mahapatra, R., Nerur, S., and Price, K., 2009. Are two heads better than one for software development? The productivity paradox of pair programming. *MIS Quarterly,* 33(1), pp.91-118.

Bano, M., and Zowghi, D., 2015. A systematic review on the relationship between user involvement and system success. *Information and Software Technology,* 58, pp.148-169.

Barrett, M., and Oborn, E., 2010. Boundary object use in cross-cultural software development teams. *Human Relations,* 63(8), pp.1199-1221.

Batista, J., and de Figueiredo, A. D., 2000. SPI in a very small team: A case with CMM. *Software Process Improvement and Practice,* 5, pp.243-250.

Baxter, G. and Sommerville, I., 2011. Socio-technical systems: From design methods to systems engineering. Interacting with Computers, 23, pp.4-17.

Benamati, J., and Lederer, A. L., 2010. Managing the impact of rapid change. *Information Resources Management Journal,* 23(1), pp.1-16.

Biro, M., and Messnarz, R., 2009. SPI experiences and innovation for global software development. *Software Process Improvement and Practice,* 14, pp.243-245.

Boehm, B., 2002. Get Ready for Agile Methods, with Care. *IEEE Computer,* 35(1), pp.64-69.

Buchman, C. D., and Bramble, L. K., 1995. Three-tiered software process assessment hierarchy. *Software Process Improvement and Practice,* 1(2), pp.99-106.

Carnegie Mellon., 2013. Published CMMI Appraisal Results of Canadian Firms: Staged representation. *CMMI Institute*.

Casey, V., and Richardson, I., 2009. Implementation of global software development: A structured approach. *Software Process Improvement and Practice,* 14, pp.247-262.

Ceschi, M., Sillitti, A., and Succi, G., 2005. Project management in plan-based and agile companies. *IEEE Software*, pp.21-27.

Chin, W. W., 2010. *How to write-up and report PLS analysis*. Berlin Heidelberg: Springer - Verlag.

Clarke, P., and O'Connor, R. V., 2013. An empirical examination of the extent of software process improvement in software SMEs. *Journal of Software: Evolution and Process*, pp.1-18.

Cockburn, A., and Highsmith, J., 2001. Agile Software Development: The People Factor. *Computer,* 34(11), pp.131-133.

Creswell, J. W., 2009. *Research design: Qualitative, quantitative, and mixed methods approaches* (Third Edition ed.). Los Angeles: SAGE.

Dalton, J., 2016. A guide to Scrum and CMMI: Improving agile performance with CMMI. *CMMI Institute*, pp.1-130.

DiMaggio, P., and Powell, W. W., 1983. The iron cage revisited: Institutional isomorphism and collective rationality in organizational fields. *American Sociological Review,* 482, pp.147-160.

Dingsoyr, T., Nerur, S., Balijepally, V., and Moe, N., 2012. A decade of agile methodologies: Towards explaining agile software development. *The Journal of system and software,* 85(6), pp.1213-1221.

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

Downe-Wamboldt, B., 1992. Content analysis: Method, applications, and issues. Health Care for Women International, 13 (3), pp.313-321.

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

Edwards, J.R. and Lambert, L.S., 2007. Methods for integrating moderation and mediation: A general analytical framework using moderated path analysis. Psychological Methods, 12 (1), pp.1-22.

Espinosa-Curiel, I. E., Rodriguez-Jacobo, J., and Fernandez-Zepeda, J. A., 2013. A framework for evaluation and control of the factors that influence the software process improvement in small organizations. *Journal of Software: Evolution and Process,* 25(4), pp.393-406.

Fornell, C., and Larcker, D., 1981. Evaluating structural equation models with unobservable variables and measurement error. *Journal of Marketing Research,* 18(1), pp.39-50.

Garousi, V., and Zhi, J., 2013. A survey of software testing practices in Canada. *Journal of Systems and Software,* 86(5), pp.1354-1376.

Halilovic, S., and Cicic, M., 2013. Antecedents of information systems user behaviour. *Behavior and Information Technology,* 32(4), pp.359-370.

Harter, D. E., Slaughter, S. A., and Krishnan, M. S., 1998. *The life cycle effects of software quality: A longitudinal analysis.* Paper presented at the The International Conference on Information Systems, Helsinki, Finland.

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

Hastie, S., and Wojewoda, S., 2015. Standish Group 2015 Chaos Report - QandA with Jennifer Lynch. *InfoQ*, pp.1-24.

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

Iversen, J., and Ngwenyama, O., 2006. Problems in measuring effectiveness in software process improvement: A longitudinal study of organizational change at Danske Data. *International Journal of Information Management,* 26(1), pp.30-43.

Jeong, B. K., and Yoon, T. E., 2016. Improving IT process management through value stream mapping approach: A case study. *Journal of Information Systems and Technology Management,* 13(3), pp.389-404.

Johansen, J., and Pries-Heje, J., 2007. Success with improvement - requires the right roles to be enacted - in symbiosis. *Software Process: Improvement and Practice,* 12(6), pp.529-539.

Keller, K., and Mack, B., 2013. Maturity profile reports. *CMMI Institute*, pp.1-26.

Kotonya, G. and Sommerville,I., 1996. Requirements engineering with viewpoints. Software Engineering Journal, 11 (1), pp.5-18.

Krasner, H., 2001. Accumulating the body of evidence for the payoff of software process improvement. *Software Process Improvement, IEEE Computer*, pp.519-539.

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

Laporte, C. Y., and O'Connor, R., 2016. Systems and software engineering standards for very small entities: Accomplishments and overview. *IEEE Computer Society*, pp.84-87.

Luftman, J., Zadeh, H. S., Derksen, B., Santana, M., Rigoni, E. H., and Huang, Z., 2012. Key information technology and management issues 2011-2012: An international study. *Journal of Information Technology,* 27(3), pp.198-212.

Markus, M. L., and Keil, M., 1994. If we build it they will come: Designing information systems that users want to use. *Sloan Management Review,* 35(4), pp.11-25.

Merlo, E., Carneiro, T. C., and Araujo, C. A., 2016. Factors influencing software migration decision: Casestudies of Acerlormittal Tubarao and the City Government of Serra, Espirito Santo. *Journal of Information Systems and Technology Management,* 13(3), pp.515-532.

Montoni, M. A., Cerdeiral, C., Zanetti, D., and Da Rocha, A. R., 2008. *A knowledge management approach to support software process improvement implementation initiatives.* Paper presented at the European Conference on Software Process Improvement.

Mumford, E., 1978. A participative approach to the design of computer systems. Impact of Science on Society, 28 (3), pp.235-253.

Mumford, E., 1985. Defining system requirements to meet business needs: A case study example. The Computer Journal, 28 (2), pp.96-104.

Mumford, E., 2000. A socio-technical approach to systems design. Requirements Engineering, 5 (2), pp.125-133.

Newman, M., and Robey, D., 1992. A Social Process Model of User-Analyst Relationships. *MIS Quarterly,* 16(2), pp.249-266.

Ngwenyama, O., and Norbjerg, J., 2010. Software process improvement with weak management support: An analysis of the dynamics of intra-organizational alliances in IS change initiatives. *European Journal of Information Systems,* 19(3), pp.303-319.

Niazi, M., 2012. An exploratory study of software process improvement implementation risks. *Journal of Software: Evolution and Process,* 24(8), pp.877-894.

Niazi, M., and Babar, M. A., 2009. Identifying high perceived value practices of CMMI level 2: An empirical study. *Information and Software Technology,* 51, pp.1231-1243.

Niazi, M., Babar, M. A., and Verner, J. M., 2010. Software process improvement barriers: A cross-cultural comparison. *Information and Software Technology,* 52(11), pp.1204-1216.

O'Connor, R. V., and Laporte, C. Y., 2014. An innovative approach to the development of an international software process lifecycle standard for very small entities. *International Journal of Information Technologies and Systems Approach,* 7(1), pp.1-22.

Oktaba, H., 2006. *MoProSoft: A software process model for small enterprises.* Paper presented at the First International Research Workshop for Process Improvement in Small Settings, Pittsburgh, Carnegie Mellon University.

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

Petter, S., DeLone, W., and McLean, E., 2013. Information systems success: The quest for the independent variables. *Journal of Management Information Systems,* 29(4), pp.7-62.

Pino, F. J., Garcia, F., and Piattini, M., 2008. Software process improvement in small and medium software enterprises: A systematic review. *Software Quality Journal,* 16(2), pp.237-261.

Ponelis, S. R., and Holmner, M. A., 2015. ICT in Africa: Enabling a better life for all. *Information Technology for Development,* 21(1), pp.1-11.

Rainer, A., and Hall, T., 2002. Key success factors for implementation software process improvement: A maturity-based analysis. *The Journal of system and software,* 62, pp.71-84.

SEI., 2005. Capability Maturity Model Integration (CMMI) version 1.1 Overview. Retrieved on February 9, 2007. www.sei.cmu.edu/cmmi/adoption/pdf/cmmi-overview05.pdf, p.9

SEI., 2010. CMMI for Development, Version 1.3. *Carnegie Mellon University, Software Engineering Institute, CMU/SEI-2010-TR-033*.

Sherer, S.A. and Alter, S., 2004. Information systems risks and risk factors: Are they mostly about information systems? Communications of the Association for Information Systems, 14 (2), pp.29-64.

Sommerville, I., 2011. Software engineering, 9th Edition. Addition-Wesley, Boston, MA.

Standish Group, T., 2013. Chaos Manifesto 2013: Think big, act small. *The CHAOS Manifesto*, pp.1-48.

Standish Group, T., 2015. Standish Group 2015 Chaos Report. *The Standish Group*, pp.1-16.

Staples, M., Niazi, M., Jeffery, R., Abrahams, A., Byatt, P., and Murphy, R., 2007. An exploratory study of why organizations do not adopt CMMI. *The Journal of system and software,* 80(6), pp.883-895.

Sulayman, M., Urquhart, C., Mendes, E., and Seidel, S., 2012. Software process improvement success factors for small and medium Web companies: A qualitative study. *Information and Software Technology,* 54(5), pp.479-500.

Turk, D., France, R., and Rumpe, B., 2005. Assumptions Underlying Agile Software-Development Processes. *Journal of Database Management,,* 16(4), pp.62-87.

Westlund, S. G., and Hannon, J. C., 2008. Retaining talent: Assessing job satisfaction facets most significantly related to software developer turnover intentions. *Journal of Information Technology Management,* 19(4), pp.1-15.

Zheng, Y., Zhao, K., and Stylianou, A., 2013. The impacts of information quality and system quality on users' continuance intention in information-exchange virtual communities: An empirical investigation. *Decision Support Systems,* 56, pp.513-524.

## Appendix A: Survey Instrument - Software Quality and Success

1. Which of the following best describes your organization's industry?
□ Communications □ Education □ Finance □ Government □
Health
□ Hotel and Hospitality □ Information Tech. □ Insurance □ Manufacturing
□ Transportation □ Utilities □ Other (please specify) ______________________

2. Please specify the approximate number of employees in your organization
□ Less than 10 □ 10 – 50 □ 51 – 250 □ More than 250

3. Kindly state the name of the IS project you are describing (i.e. A recently (< 2yrs) completed project)
____________________________________________________________

4. In which country was the information system project undertaken?
□ Barbados □ Guyana □ Jamaica □ Trinidad
□ Canada □ U.S.A. □ U.K. □ Other (please specify) _______________

5. The number of team members in this project was:
□ Less than 3 □ 3 - 5 □ 6 - 10 □ 11 - 15 □ More than 15

6. The scope of this project can be classified as the:
□ Development of a new system □ Development of a new sub-system(s)
□ Modification of an existing system □ Modification of an existing sub-system(s)

7. The organization develops software for:
□ In-house use □ Sale □ Other (please specify) _________________________

8. What is your gender? □ Male □ Female

9. What activities were you responsible for on this project? (Please tick as many as are applicable)
□ Software Requirements □ Software Quality Assurance □ Software Design
□ Version Management □ Coding □ Software Process Improvement
□ Test and Integration □ User Training □ Data Entry □ Other _____________

10. Which role(s) best described your position on this project? (Please tick as many as are applicable)
□ Project Manager □ Team Leader □ Team Member
□ Software Analyst □ Software Developer □ Programmer
□ User □ IS Consultant □ Subject Matter Expert
□ IS Contractor □ Other (please specify) _________________________________________

11. How many years have you worked as an IS professional? _____ Year(s)

## Process Maturity Practices

| **Requirements Management and Requirements Development** are about analyzing and producing the system requirements and managing customer requirements. | Strongly Disagree | | | | | | | Strongly Agree |
|---|---|---|---|---|---|---|---|---|
| 12. Requirements were well established for this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 13. Detailed records were kept of the requirement changes in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 14. Requirement management and requirement development on this project were guided by organization policy | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 15. Performance measures (e.g. total number of requirements changes that were proposed, opened, approved or closed) were in place on this project for requirement management and requirement development | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

| **Project Planning** establishes and maintains the plans that define project activities | Strongly Disagree | | | | | | | Strongly Agree |
|---|---|---|---|---|---|---|---|---|
| 16. Detailed cost estimates were in place for managing this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 17. Detailed time estimates were in place for managing this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 18. All relevant tasks were identified for successful execution of this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 19. Adequate resources were in place for the planning of this project (e.g. funding, expertise, etc.) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 20. Performance measures (e.g. completion of milestones) were in place for planning this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

| **Project Monitoring and Control** provides an understanding of the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan | Strongly Disagree | | | | | | | Strongly Agree |
|---|---|---|---|---|---|---|---|---|
| 21. Corrective actions were always taken to manage variances (actual vs plan) in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 22. Changes to development plans were always signed off by relevant stakeholders in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 23. Organization policies were in place for guiding project monitoring and control in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 24. Performance measures were in place to assess project monitoring and control in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

| **Technical Solution** is about designing, developing and implementing solutions to user requirements | Strongly Disagree | | | | | | | Strongly Agree |
|---|---|---|---|---|---|---|---|---|
| 25. A well established plan was developed to address all requirements for this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 26. All needed resources were in place to execute the solution for all the requirements in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 27. Performance measures were in place to assess the effectiveness of technical solution for this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

| **Verification** describes the steps taken to ensure that the activities are performed in compliance with processes such as reviews, audits and software quality assurance while **Validation** is checking that the software process produced the intended results such as formal walkthroughs and inspections | Strongly Disagree | | | | | | | Strongly Agree |
|---|---|---|---|---|---|---|---|---|
| 28. Verification activities were effectively planned for this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 29. Validation activities were effectively planned for this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 30. Adequate resources were provided to perform verification activities in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 31. Adequate resources were provided to perform validation activities in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

| **Organization Process Definition** establishes and maintains a usable set of software development procedures and standards | Strongly Disagree | | | | | | Strongly Agree | |
|---|---|---|---|---|---|---|---|---|
| 32. The organization's software development procedures were adequately documented in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 33. The organization's software development procedures were clearly understood in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 34. Performance measures were in place to assess if key stakeholders understand the firm's software development processes and procedures covered by this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

| **Organization Process Focus** plans, implements and deploys process improvements based on a thorough understanding of the strengths and weaknesses of the organization's software development processes | Strongly Disagree | | | | | | Strongly Agree | |
|---|---|---|---|---|---|---|---|---|
| 35. There were established software development processes to be followed by key stakeholders in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 36. Adequate measures were taken to adjust the organization's software development processes based on performance indicators for this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 37. Performance measures were in place to assess the effectiveness of all organization software development processes in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

| **Organization Training** is about developing the skills and knowledge of project personnel so they can perform their roles effectively and efficiently | Strongly Disagree | | | | | | Strongly Agree | |
|---|---|---|---|---|---|---|---|---|
| 38. Project team members were adequately trained to perform their roles in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 39. Adequate resources were provided to facilitate training for this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 40. Performance measures were in place to assess the effectiveness of training in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

| **Integrated Project Management** is about managing the project in a manner that brings team members together in a coordinated manner | Strongly Disagree | | | | | | Strongly Agree | |
|---|---|---|---|---|---|---|---|---|
| 41. A well established policy document regarding the integration of all relevant project team members was in place for this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 42. High levels of co-operation were encouraged among all relevant project groups in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 43. Performance measures were in place to assess the integration of project team members in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

| **Risk Management** is about identifying potential problems before they occur so that risk-management activities can be planned and put into action as needed | Strongly Disagree | | | | | | Strongly Agree | |
|---|---|---|---|---|---|---|---|---|
| 44. Potential risks were clearly identified for this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 45. A clearly defined action plan was established for possible risks in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 46. All needed resources were in place to address potential risks in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 47. Performance measures were in place to assess the management of risk in this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

## People Skills

| **Competence** is about the depth of project management skills, software development methods, software engineering skills and faithful execution of software practices | Strongly Disagree | | | | | | Strongly Agree | |
|---|---|---|---|---|---|---|---|---|
| 48. High levels of project management skills (e.g. project estimating, planning, controlling) were displayed on this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 49. High levels of business process expertise were available during the execution of this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 50. High levels of system analysis and design expertise were available during the execution of this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 51. Adequate understanding of modelling techniques were displayed during the execution of this project (e.g. CASE tools, formal methods) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 52. Competent data analysts and database administrators were assigned to this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 53. Competent systems construction personnel were assigned to this project (e.g. programmers, networkers, etc.) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 54. Competent code testers and documentation personnel were assigned to this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

| **Contribution** is about the degree of contribution by the project team members towards the successful execution of the project | Strongly Disagree | | | | | | Strongly Agree | |
|---|---|---|---|---|---|---|---|---|
| 55. Project members contributed greatly to the success of this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 56. Project members participated equally in the execution of the project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 57. Project members contributed to the project achieving its stated objectives | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

## Technology

| **Technological Tools:** This looks at the extent to which technological tools are used to support the successful and efficient execution of the project | Strongly Disagree | | | | | | Strongly Agree | |
|---|---|---|---|---|---|---|---|---|
| 58. Project management software (e.g. Microsoft Project, Primavera, Timeline, Rational, or others) was used in the execution of this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 59. Formal methods (e.g. ASM and VDM) and computer aided software engineering (CASE) tools were used in the execution of this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 60. Using technological tools (such as project management software) increased the efficiency of this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 61. Using technological tools (such as formal methods, CASE tools, etc.) increased the efficiency of this project | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

## IS Quality

| **Functionality of the delivered system** | Strongly Disagree | | | | | | Strongly Agree | |
|---|---|---|---|---|---|---|---|---|
| 62. The delivered system features are suitable for achieving the business objectives | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 63. The system generally provides the support I need to do my job | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 64. I am not tempted to work around the system to perform system tasks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| **Reliability of the delivered system** | Strongly Disagree | | | | | | Strongly Agree | |
| 65. The delivered system is usually available when needed | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 66. I can easily recover from input errors | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 67. The system is usually restored quickly after a failure | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| **Usability of the delivered system** | Strongly Disagree | | | | | | Strongly Agree | |
| 68. The delivered system is easy to understand | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 69. The delivered system is easy to learn | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 70. The delivered system is easy to use | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| **Efficiency of the delivered system** | Strongly Disagree | | | | | | Strongly Agree | |
| 71. The response time of the system is generally acceptable | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 72. It is generally easy to use systems features | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| **Maintainability of the delivered system** | Strongly Disagree | | | | | | Strongly Agree | |
| 73. The delivered system provides diagnostics that help to identify causes of failures | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 74. Relatively little effort is expended to correct faults whenever system errors occur | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 75. Relatively little effort is needed to test system modifications | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

## User Perception

| **Perceived usefulness:** The degree to which a person believes that using a particular system would enhance his or her job performance | Strongly Disagree | | | | | | Strongly Agree | |
|---|---|---|---|---|---|---|---|---|
| 76. The delivered system has made it easier to do my job | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 77. The delivered system improves my job performance | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 78. The delivered system improves my decision making capabilities | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

| **Perceived ease of use:** The degree to which a person believes that using a particular system would be free of effort | Strongly Disagree | | | | | | Strongly Agree | |
|---|---|---|---|---|---|---|---|---|
| 79. Learning to use the delivered system is easy for me | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 80. I interact easily with the delivered system and I find it easy to understand | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 81. The system has online help/user manual that is easy to follow | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

## IS Success

| **System usage:** The extent to which the system is accepted and used, which is an event that can lead to individual and organizational impact | Strongly Disagree | | | | | | Strongly Agree | |
|---|---|---|---|---|---|---|---|---|
| 82. I believe the delivered system is serving its purpose | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 83. I frequently use the delivered system in the execution of my job | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 84. I am very knowledgeable about the features of the delivered system | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

| **Business value creation:** The degree to which the delivered system provides value to the organization that deploys it | Strongly Disagree | | | | | | Strongly Agree | |
|---|---|---|---|---|---|---|---|---|
| 85. The delivered system has increased my productivity | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 86. The delivered system has improved information flow in the firm | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |
| 87. The delivered system has enhanced customer service | 1 | 2 | 3 | 4 | 5 | 6 | 7 | n/a |

Thanks for your support

## Appendix B: Interview Questions on IS Quality and Success

1. Why do you think that people skills and contribution had the greatest impact on information systems (IS) quality over technology and process maturity in Canada?

2. Why do you think that technology had the lowest impact on information systems (IS) quality in Canada?

3. The variance explained by people, process maturity and technology on information systems (IS) quality [$R^2$] was 0.760; What other factors could impact IS quality in Canada?

4. Why do you think that perception had a greater impact on information systems (IS) success over information systems (IS) quality in Canada?

5. The variance explained by perception and IS quality on IS success [$R^2$] was 0.781; What other factors could impact IS success in Canada?

6. Why do you think the loading on ‘Organization process focus’ was below the 0.700 threshold?

7. What other observations or insights can you make regarding this study or the phenomena of software quality and success in Canada?