The Role of Quality Assurance in Software Development Projects: Project Failures and Business Performance

Dr. Emmanuel Ichu
Adjunct Instructor
Tarrant County College
Garland, Texas, USA
E-mail: omeogaa@gmail.com

Dr. Rao Nemani
Adjunct Instructor
The College of St. Scholastica
Duluth, Minnesota, USA
E-mail: RNemani@css.edu

Abstract

Despite rapid advances in all facets of technology, the software industry still struggles with the formidable task of developing software applications that meet quality standards, and budget constraints. The need for software to be error-free remained a challenge to the IT industry. Therefore, the primary purpose of this study is to answer why executives reluctant to allocate resources to quality assurance (QA) process during the process of the system development life cycle (SDLC)? This research used a qualitative study design to investigate to what extent the involvement of QA during the SDLC process reduced software project failures. The data were analyzed using inductive methods and was intended to be generalized to the entire IT software development population. Participants were selected from major cities of the 50 United States that fit study criteria. The research indicated a sense that inclusion of QA in all phases of SDLC was an excellent phenomenon.

1. Introduction

Despite rapid advances in all facets of technology, the software industry still struggles with the formidable task of developing software applications that meet quality standards, time pressure, and budget constraints [1]. Information systems/information technology (IS/IT) in today’s complex and dynamic environment is very important and its development has continued to grow; however, IT faces the challenge of how to successfully develop its product with established methodologies [1] [4] [6] [16] [23]. Butthman [4] added that costs do not result only from producing and fixing failures; a high amount of costs come from ensuring that good products are produced. The Department of Commerce, Economics, and Statistics Administration [11] in a study focused on the value of software development, showed that Investment in software development increased from $82 billion in 1995 to $149 billion in 1999.

Quality assurance (QA) in this study was described as a function with responsibility to ensure that software meets its intended requirements—functional, date, staff, budget, etc. A similar function included was software testing. It was designed to prove that a system does meet its requirement. Interchangeably, QA and software testing was used as this study explored their inclusion in the system development life cycle (SDLC). Schenk, Vitalari, and Davis [33] indicated that errors found early in the design process could have rippling effects, and these errors were expensive and time consuming to correct after software project completion. Ewusi-mensah [13] indicated that 52.7% of IT projects completed are 189% over budget at an additional cost of $59 billion. Furthermore, Hardgrave et al. [16] pointed at the need to integrate people in the software development effort by referring to Pfleeger’s comments [30] which reminded the IT industry, especially the fields of software engineering, to better understand the role of people in the adoption process, and how it related to drawing upon social science models. Hardgrave et al. [16] stated in their study that there was need to improve software development with methodologies; however, these methodologies were not a panacea for all software development problems. The use of methodologies was seen as a positive direction towards achieving quality software products.

According to McDonagh and Coghlan [23] software project failures could be attributed to what was reported as conflicts between business and IT executives. Business executives seemed to be allocating fewer resources as they focused more on economics while IT executives focused on technical advancement with available technology. Ashrafi [1] defined the QA group as “professionals
founded to promote the Quality Assurance profession through proliferation and advancement of high professional standards” (p. 678). This group embraced the framework as a guideline to judge the quality of a software system. Glen [15] indicated that QA views IT projects through the lens of status. Glen [15] further indicated that QA’s role was to constantly determine the functional status of the product in development through objective tests and shared that information with the rest of the team to support fact-based decision making. Pyhäjärvi and Rautiainen [31] indicated the role of QA as reporting defects and verifying that they were resolved by the developer before migration, unless management decided not to fix the defect before going to production depending on its severity.

The need for software to be error-free remained a challenge to the IT industry. Dey, Kinch, and Ogunlana [12] stressed the importance of success of software development, which depended on functionality, quality, and timeliness. Dey et al. [12] also indicated that unless developed successfully to perform its desired function, the purpose of software development was defeated. Charette [5] urged that there were many facets to the root cause of software project failures with emphasis on two of the primary causes namely, poor planning and a lack of training, especially in QA. Osterweil [37] explained that a software failure occurred when a piece of the software failed to perform as users required and expected. Bessin [3] noted that “the U.S. economy spent almost $60 billion annually due to software defects cost” (p. 1). Having mentioned these extraordinary loses to the U.S. economy due to software failures, there is a need to identify how to eliminate or reduce these losses significantly.

1.1 Background of the Study

QA is determined by organizational size, organizational culture, application criticality, time to market, and the software development process [29]. Testing methods had evolved from the early years of haphazard testing processes performed by the development team to include formalized processes, testing teams, and defect tracking [7] [36]. Early testing processes were based on the waterfall model, which involved each stage of the SDLC broken into discrete phases (National Institute of Standards and Technology, 2002). In the waterfall model, testing software is broken into multiple phases of development, namely; unit test, application readiness test, application function test, application regression test, application performance/stress test, integration test, interface transaction, and solution end to end test [3]. Many testing approaches such as test planning, requirements and design analysis, test requirements and objectives, test plans, test cases, test scripts/procedures, test implementation, test execution and results, defect/bug reporting, and tracking results reporting (test metrics) have been created from the original waterfall systems development model that incorporated business requirements and functional requirements into formalized documents such as test plans and test objectives.

It is the responsibility of IT leaders to minimize the organizational risk of software defects released into the production environment and at the same time manages costs and time-to-market as agreed upon with stakeholders [16]. Defects found in production are more expensive to fix and can cause catastrophic problems to an organization or an individual, including the loss of life [18]. According to Cule, Schmidt, Lyytinen, and Keil [9] “Information Technology and Information System (IT/IS) projects are on the increase and they have continued to fail” (p. 1). Despite the continuous failures, IT executives have continued to engage resources to these projects [27]. According to McDonagh and Coghlan [23] corporate conflict between the business and IT executives has continued to impact software projects.

1.2 Problem and Purpose of Study

Charette [5] reported that organizations and governments spent an estimated $1 trillion on IT hardware, software, and services worldwide. According to the Standish Group study conducted in 1995 [as cited in 19] the U.S. government and businesses spent approximately $81 billion on canceled software projects, and another $59 billion for budget overruns. In another related study, Michaels [25] indicated that it was hard to determine the real cost of failed software projects; however, in the United States alone it was estimated to be upwards of $75 billion a year in rework costs and abandoned systems. The survey claimed that in the United States, only about one sixth of all projects were completed on time and within budget, nearly one third of all projects were canceled outright, and well over half were considered “challenged.” Of the challenged or canceled projects, the average project was 189% over budget, 222% behind schedule, and contained only 61% of the originally specified features.

This research study explored to what extent the inclusion of QA in all phases of the SDLC would increase organizational economic status, and conceptual and economic perspective. The data collection process for this study included personal interviews with practitioners and consultants who were asked to describe the causes of software
project failures with which they have been acquainted and how the QA department was involved during the project from 2003 to 2008.

1.3 Conceptual Framework

According to Miles and Huberman [26] a “conceptual framework explained, either graphically or in narrative form, the key factors, constructions or variables—and the presumed relationships among them” (p. 18). Figure 1 shows the conceptual framework for the present study.

![Figure 1. Conceptual framework](image)

In the conceptual framework, McAllister [22] indicated that the development of information systems involved knowing what to create and also how to create it. He indicated that requirements gathering were often identified as the most difficult part of bringing an information system into existence. The system analyst’s ability to develop accurate, complete, and clear information requirements was paramount for successful systems building [33]. Davidson [10] added that requirements determination during information systems delivery was a complex organizational endeavor, involving political sense making and communicative processes. For a successful and meaningful project, the theoretical framework for the study related to the management theories from IT, and organizational performance in terms of the effects of QA and IT on organizational performance and the role of QA in software development implementation. In the conceptual framework, immediately after the completion of requirements gathering, all stakeholders should review them before the next step, which is presentation to management for budget approval. The budget should be made available to respective departmental heads to enable them to procure necessary resources or update them as required in readiness for the project. The developers initiated their unit testing, which involves developing code to create the software.

2. Assessment of Current Literature

There is wide agreement in literature for the difficulty of including QA in all phases of the SDLC, and allocating budget for QA head count. Although the development of information systems had been examined from multiple dimensions, little was known about why QA was not included in all phases of the SDLC and studies have not prioritized the factors involved. Most studies on improving software projects were citing quality of products, abandoning of projects, projects staying within budget, user requirements, and the like; however, QA inclusion had not been focused on as a technique for preventing software project failures. Non inclusion of QA in all phases was considered an omission of a powerful tool in the prevention of errors going to production, which in turn renders the software project a failure.

2.1 Methodology

According to McAllister [22] information systems often failed to satisfy users’ expectations if developers misunderstand their requirements. McAllister further stated that ineffective communication is only one of many reasons why developers misunderstand users’ requirements. Instead of continuing to rely on assumptions about why requirements are misunderstood, this study sought to qualitatively identify the inclusion of QA as another factor that influenced misunderstandings about requirements and prioritize the perception of QA as value-added instead of a cost to software projects, as seen by some business executives. Consequently, this theory-building study might improve requirement determination processes in the future with the inclusion of QA in all phases of the SDLC. This was best accomplished using a qualitative study design [24]. The study investigated to what extent the involvement of QA during the SDLC process would reduce software project failures, influence organizational performance, and also explored how management allocation of funds would increase QA involvement during the SDLC.

2.2 Research Questions

This study would provide answers to a research question “Why are business executives reluctant to allocating resources to quality assurance (QA) for performing testing throughout the entire process of the system development life cycle (SDLC)?” These research questions were converted into hypotheses.
Table 1 Research Question Hierarchy

<table>
<thead>
<tr>
<th>Type of question</th>
<th>Question or statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business dilemma</td>
<td>In the United States alone it is estimated to be upwards of $75 billion a year in rework costs and abandoned systems projects.</td>
</tr>
<tr>
<td>Management question</td>
<td>To what extent would the inclusion of Quality Assurance (QA) in all phases of the system development life cycle (SDLC) increase organizational economic status, conceptual, and economic perspective?</td>
</tr>
<tr>
<td>Research question</td>
<td>Why are business executives reluctant to allocate resources to quality assurance (QA) for performing testing throughout the entire process of the system development life cycle (SDLC)?</td>
</tr>
</tbody>
</table>

3. Data Analysis

3.1 Data Collection

In keeping with qualitative design, the participants of this study were purposefully selected [8] [24]. The sample of participants was best described as a typical sample [24] which reflected the average in a given population. The interviewees were selected from a total population consisting of selected software developing organizations spread across selected major cities from the 50 United States.

3.2 General Research Philosophy

The present study was guided by a philosophy of pragmatism. According to Miles [26] and Huberman (1994), “any method that works—that produced clear, verifiable, credible meanings from a set of qualitative data—is grist for our mill, regardless of its antecedents” (p. 3). This pragmatic approach was also discussed by Robson [32] who acknowledged a basic incompatibility between the historical extremes of positivists and constructivists. He characterized the pragmatic perspective as “whatever philosophical or methodological approach that works best for a particular research problem at issue” (p. 43). Regardless of the specific method, the pragmatist was concerned with defining the scope and intent of a study, ensuring reliability and validity, and using an adequate research structure. The pragmatist used a scientific method that included systematic procedures, a skeptical interpretation of results, and an ethical code of conduct. Consequently, the research approach described was chosen to answer the research questions without regard to epistemological preferences.

3.3 Demographic Survey Results

The demographic survey and person to person interviews were given to information technology professionals selected from various parts of the 50 United States. The demographic survey gathered the following information:

1. Gender and age.
2. Highest education level achieved.
3. Employment status, industry, and annual salary.

The survey results reported five women, representing 20% participating and 20 men, representing 80% male participants. The research evaluated the age of the professionals in the study. Results ranged from 25–34 years, representing 23%, 35–44 years, representing 23%, 45–54 years, representing 46%, and 55–64 years, representing 9%. The demographic survey also asked the respondents to identify their academic levels and the result showed 76% completed college degrees, 14% possessed Associate Degrees, and 10% held postgraduate degrees. The study sought to include participants’ employment status, the industry employed, and annual income.

The study collected data representing the annual income of respondents who reported a range of $60,000–74,000 representing 3%, $75,000–89,999 representing 10%, and $90,000 or more representing 60%. In an effort to evaluate the respondents’ participation on resource allocation decision making, the survey data collection found that 66% were involved while 34% were not involved in resource allocation decision making. The development approach was considered in the research because the SDLC method was used to determine QA participation.

The research also sought to determine at what stage QA would be integrated into the SDLC process. The result indicated approximately 67% of those surveyed responded that QA should be included in the process as soon the clients’ requirements are approved, 34% wanted to bring QA into the process before the design document was sent to the developers, 25% wanted QA to become involved as soon as the Unit test development starts, 17% wanted QA to become

ISSN:2229-6093
involved after Unit testing is completed, 15% wanted QA to become involved just before the product is ready for migration, and 15% said QA should get into the SDLC when user acceptance testing starts.

The development approach was considered in the research because the SDLC method was used to determine QA participation. The participants were asked to indicate the approaches they have used (Table 2) and 24% of respondents indicated they are following waterfall SDLC approach and about 60% of respondents are using a combination of incremental, iterative and agile SDLC approaches.

Table 2 Development Approaches Used

<table>
<thead>
<tr>
<th>SDLC approach</th>
<th>n</th>
<th>% of total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waterfall</td>
<td>6</td>
<td>24.0</td>
</tr>
<tr>
<td>Incremental</td>
<td>5</td>
<td>20.0</td>
</tr>
<tr>
<td>Iterative</td>
<td>5</td>
<td>20.0</td>
</tr>
<tr>
<td>Agile</td>
<td>5</td>
<td>20.0</td>
</tr>
<tr>
<td>Other</td>
<td>4</td>
<td>16.0</td>
</tr>
</tbody>
</table>

The study collected data that indicated that QA should be introduced to all software development organizations, if one does not exist. 100% of respondents wanted QA to be a part of all software development organizations. Table 3 displays a high percentage of acceptances of usage of QA in all phases of SDLC. The study, in an effort to explore a management dilemma, stated that in the United States alone, it is estimated to be upwards of $75 billion a year in rework costs and abandoned systems projects. Approximately 65% of the survey population responded to a question that asked why management should spend time and money in SDLC. In an effort to understand why some business process managers were reluctant on involving QA in all phases of SDLC, the study sought from practitioners their opinions and about 40% of respondents responded with different views. Respondents surveyed were asked what they thought were some common solutions to software development problems. Only 60% of those surveyed responded and provided some suggestions.

Table 3 QA Testing Involved in Software Projects

<table>
<thead>
<tr>
<th>Use QA in all software development</th>
<th>n</th>
<th>% of total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>18</td>
<td>72.0</td>
</tr>
<tr>
<td>No</td>
<td>5</td>
<td>20.0</td>
</tr>
<tr>
<td>No response</td>
<td>2</td>
<td>8.0</td>
</tr>
</tbody>
</table>

The study also sought the number of software projects that failed after migration within a 5-year period (2003–2008). Only 36% of participants responded and 15% of them reported the failures were as a result of end-user refusal to use the product due to their requirements not being met. In another similar question that tried to investigate the number of projects abandoned within a 5-year period, 76% responded.

In an attempt to understand the relationships between the developers and QA professionals, the researcher asked respondents to characterize their relationship. The data collected showed 83.3% had very good relationships between the developers and the QA professionals that were used in the development process. The research also sought to determine at what stage QA would be integrated into the SDLC process. The result indicated approximately 67% of those surveyed responded that QA should be included in the process as soon as the clients’ requirements are approved, 34% wanted to bring QA into the process before the design document was sent to the developers, 25% wanted QA to become involved as soon as the Unit test development starts, 17% wanted QA to become involved after Unit testing is completed, 15% wanted QA to become involved just before the product is ready for migration, and 15% said QA should get into the SDLC when user acceptance testing starts. The study collected data that indicated that QA should be included in the process as soon as the clients’ requirements are approved, 34% wanted QA to become involved as soon as the Unit test development starts, 17% wanted QA to become involved after Unit testing is completed, 15% wanted QA to become involved just before the product is ready for migration, and 15% said QA should get into the SDLC when user acceptance testing starts. The study also sought the number of software projects that failed after migration within a 5-year period (2003–2008). Only 36% of participants responded and 15% of them reported the failures were as a result of end-user refusal to use the product due to their requirements not being met. In another similar question that tried to investigate the number of projects abandoned within a 5-year period, 76% responded.

In an attempt to understand the relationships between the developers and QA professionals, the researcher asked respondents to characterize their relationship. The data collected showed 83.3% had very good relationships between the developers and the QA professionals that were used in the development process. The research also sought to determine at what stage QA would be integrated into the SDLC process. The result indicated approximately 67% of those surveyed responded that QA should be included in the process as soon as the clients’ requirements are approved, 34% wanted QA to become involved as soon as the Unit test development starts, 17% wanted QA to become involved after Unit testing is completed, 15% wanted QA to become involved just before the product is ready for migration, and 15% said QA should get into the SDLC when user acceptance testing starts. The study collected data that indicated that QA should be included in the process as soon as the clients’ requirements are approved, 34% wanted QA to become involved as soon as the Unit test development starts, 17% wanted QA to become involved after Unit testing is completed, 15% wanted QA to become involved just before the product is ready for migration, and 15% said QA should get into the SDLC when user acceptance testing starts. The study collected data that indicated that QA should be included in the process as soon as the clients’ requirements are approved, 34% wanted QA to become involved as soon as the Unit test development starts, 17% wanted QA to become involved after Unit testing is completed, 15% wanted QA to become involved just before the product is ready for migration, and 15% said QA should get into the SDLC when user acceptance testing starts. The study collected data that indicated that QA should be included in the process as soon as the clients’ requirements are approved, 34% wanted QA to become involved as soon as the Unit test development starts, 17% wanted QA to become involved after Unit testing is completed, 15% wanted QA to become involved just before the product is ready for migration, and 15% said QA should get into the SDLC when user acceptance testing starts. The study collected data that indicated that QA should be included in the process as soon as the clients’ requirements are approved, 34% wanted QA to become involved as soon as the Unit test development starts, 17% wanted QA to become involved after Unit testing is completed, 15% wanted QA to become involved just before the product is ready for migration, and 15% said QA should get into the SDLC when user acceptance testing starts.
wanted to bring QA into the process before the design document was sent to the developers, 25% wanted QA to become involved as soon as the Unit test development starts, 17% wanted QA to become involved after Unit testing is completed, 15% wanted QA to become involved just before the product is ready for migration, and 15% said QA should get into the SDLC when user acceptance testing starts. The study collected data that indicated that QA should be introduced to all software development organizations, if one does not exist. 100% of respondents wanted QA to be a part of all software development organizations. Table 4 displays a high percentage of acceptances of usage of QA in all phases of SDLC.

Table 4 QA Usage in Software Development

<table>
<thead>
<tr>
<th>Use QA in all software development</th>
<th>n</th>
<th>% of total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>18</td>
<td>72.0</td>
</tr>
<tr>
<td>No</td>
<td>5</td>
<td>20.0</td>
</tr>
<tr>
<td>No response</td>
<td>2</td>
<td>8.0</td>
</tr>
</tbody>
</table>

The study, in an effort to explore a management dilemma, stated that in the United States alone, it is estimated to be upwards of $75 billion a year in rework costs and abandoned systems projects. Approximately 65% of the survey population responded to a question that asked why management should spend time and money in SDLC. Following are participants’ responses to a question that sought justification of time and money invested in SDLC when QA is involved. The research question for the study was why business executives were reluctant in allocating resources to QA for performing testing throughout the entire process of the SDLC? Since the researcher indicated that sub questions would be used as part of the interviewing process, approximately 56% of the population surveyed provided the following responses to a sub question that asked why budgeting was not done early on, so that enough money could be allocated for QA group inclusion during the SDLC process?

Table 5 Introduction of QA Processes

<table>
<thead>
<tr>
<th>Establish QA in all software development org</th>
<th>n</th>
<th>% of total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>19</td>
<td>76.0</td>
</tr>
<tr>
<td>No</td>
<td>3</td>
<td>12.0</td>
</tr>
<tr>
<td>No response</td>
<td>3</td>
<td>12.0</td>
</tr>
</tbody>
</table>

Table 5 indicates that IT professional used in the study were in support of QA established in all software development organizations, based on their responses to a sub question that asked how they support the introduction of QA processes in an existing organization without one. The research also sought to know why management assumed that problems won’t exist when developing an enterprise product with several modules and teams involved. The research also sought to know why management assumed that problems won’t exist when developing an enterprise product with several modules and teams involved. In an effort to understand why some business process managers were reluctant on involving QA in all phases of SDLC, the study sought from practitioners their opinions and about 40% responded. In an attempt to explore the personal views of IT practitioners in the inclusion of QA in all phases of the SDLC process, this study sought the opinions of the survey sample. Approximately 36% responded. The study also sought the number of software projects that failed after migration within a 5-year period (2003–2008). Only 36% of participants responded and 15% of them reported the failures were as a result of end-user refusal to use the product due to their requirements not being met. In another similar question that tried to investigate the number of projects abandoned within a 5-year period, 76% responded. Table 6 provides a result of the participants’ responses.

Table 6 Abandoned Projects

<table>
<thead>
<tr>
<th>No. abandoned software projects</th>
<th>n</th>
<th>% of total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>32.0</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>48.0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>8.0</td>
</tr>
<tr>
<td>None</td>
<td>3</td>
<td>12.0</td>
</tr>
</tbody>
</table>

The research data collected from this study included suggestions sought from IT practitioners on how software development could be improved. The results of the analysis for this study have been presented. However, it should be noted that this study has opened many avenues to investigate QA and its inclusion in the SDLC. Thirty participants accepted to participate in the study but only 25 responded to the survey. Of these 25 respondents, 11 are full-time employees in large IT corporations, 5 are consultants from midsized organizations, with the rest of the respondents from small sized corporations that do internal software development for their own consumption. About 6 of the professionals in the large IT corporations are
involved in producing commercial software products that end-users purchase and customize or use as needed. The study had an adequate mix of IT professionals, hence; these conclusions lead to opportunities for future research.

Another interesting aspect in the data collected from the respondents was that they provided words that have concrete, vivid, meaningful flavour that proved more convincing facts about the inclusion of QA in all phases of the SDLC. It increased organizational economic status, conceptual, and economic perspective. The researcher would agree that good qualitative data helped in producing unanticipated conclusions. Finally, most of the respondents agree that QA is a critical component of the SDLC, but due to some technicalities and management view on return on investment, QA inclusion in all phases of SDLC had some pitfall as reported in an earlier section of the manuscript.

4. Discussion

The inclusion of QA in the SDLC could have contributed to the reduction of billions of dollars invested on IT software project rework. In such a case where QA was not present, IT executives would have struggled to come up with alternatives in order to keep organizational stability more or less competitive. The investigation for an alternative solution to the current way management is undertaken in order to combat the challenges of the inclusion of QA in the SDLC, lack of financial allocation, and the perceptions of IT executives concerning QA professionals, was supported by this study.

Within the study, Gall et al. [14] indicated that data could be obtained through public archival records, private archival records such as journals and calendars, and direct response, such as interviews and observation. Of the 25 respondents in this study, 11 were full-time employees in large IT corporations that produce software for external consumption, 5 were consultants from midsized firms, and the residual respondents were from small corporations that did internal software development for their own consumption. These respondents provided a wealth of data that complement Miles and Huberman’s [26] assertion that the “findings from qualitative studies have a quality of undeniability” (p. 1). In the various tables listed in this section, data collected supported the inclusion of QA in all phases of the SDLC and provided insight into why some management teams were reluctant in allocating resources for QA.

Based on the analysis of this study, there were several noteworthy outcomes uncovered during this research. In light of $75 billion spent in the U.S. economy for software product rework, 65% of the respondents to the study supported the notion that IT management should spend more money in the SDLC to prevent rework for the following reasons: (a) SDLC provides a systematic template for the successful testing, identification, and resolution of barriers to implementing a quality software release process; (b) money and time should be spent in the SDLC due to the well-known technology adage, “garbage in, garbage out,” thus avoiding the hurry to cut corners financially, typically leading to the likelihood of producing human and technological error; and (c) to have a high quality software product that could be sold down the road.

Interesting themes were seen in the entire study. One example arose from the question of how the SDLC process could be improved. About 65% responding pointed at better communication with QA as early as possible, include the addition of training and allowing more time for planning, design, and working close with end-users to ensure their requirements were understood before development starts. Following this theme was the relationship between QA and the developers. The study discovered a high percent of responses indicating good relationships existed between these key players of the SDLC.

The researcher’s original thinking was that the two groups were not being getting along very well, since QA is all about breaking the developers’ codes. A significant number of responses was observed when participants responded to what stage of the SDLC process QA should become involved. From analysis of these responses, the study revealed that IT practitioners wanted QA to be involved as soon as clients’ requirements were approved and before programmers started developing their unit test cases. Another theme of interest was related to when QA should be introduced to all software development organizations if one does not exist. Seventy-two percent of respondents wanted QA to be a part of all software development organizations.

The study also explored another theme in the inclusion of QA in the SDLC process. The survey population revealed that (a) if design issues have not been settled, it would be difficult to define permanent test parameters; (b) that corporate perception of QA is comparable to what most people believe about life insurance—it’s a “nice to have” but not “mandatory”; (c) that project delays could be due to not having enough QA personnel to complete testing. One potential solution could be anticipating this early enough and have the budgetary allowance to hire more personnel or get contractors to make sure the project is not delayed; (d) that the later QA gets involved in the process
the more expensive the bug/issue becomes geometrically more expensive; and (e) that businesses are beginning to understand the risk involved with not having a robust QA process and teams and are now moving toward QA inclusion, both financially and organizationally.

These revelations required the IT executives’ attention in order to address worldwide software development failures, for which the U.S. economy has already suffered. According to the Standish Group study [35] conducted in 2005 [as cited in 17] the U.S. government and businesses spent approximately $81 billion on cancelled software projects, and another $59 billion for budget overruns. In another related study, Michaels [26] indicated that it was hard to determine the real cost of failed software projects. As this study continued to discuss other themes, it is worthwhile to mention that 40% of the survey population indicated that IT process managers did not want QA included in all phases of the SDLC for the following reasons: (a) they did not clearly understand the full benefit of complete testing; (b) cost and time involved in QA processes, with no real, tangible benefits perceived; (c) QA is time consuming; (d) customers were waiting for product as soon as possible; and (e) if design and development were not ready, there was not much to test. The researcher saw these five reasons for not including QA in all phases of the SDLC as a lack of understanding of the benefits of defect-free products by the process managers themselves.

5. Implications and Limitations

5.1 Implications

The research question was intended to discover how selected software development organizations perceived the inclusion of QA in all phases of the SDLC and how their inclusion would benefit the software industry for the future. The findings provided descriptions of current practices and perceived implications. A second intent of the study was to determine whether there were themes or patterns that emerged from the descriptive data that could be used to help encourage business executives to view QA as value-added, instead of cost to IT projects. The sampled software development organizations were located in metropolitan areas within the 50 United States that fit study criteria. The characteristics of typical software development organizations were similar throughout the country and would benefit from the discoveries acquired through this research study. Merriam [24] described this second intent as an interpretative study, which took the data a step beyond descriptive. It is common place in education research to find combination studies (descriptive/interpretative), as described here [24]. The design and purpose of this study fit the aforementioned characteristics for qualitative research design. The data collected were reported and described from the participant’s perspective rather than the perspective of the researcher. The researcher in this study was the primary instrument for data collection and analysis. The study was inductive in approach and the researcher used detailed accounts of current processes to reach conclusions based on the observations and responses to the interview questions from the selected IT practitioners.

5.2 Assumptions and Limitations

The study looked at 10 IT companies; five were developing software for internal consumption while five were producing software for commercial consumption. Although care was taken to select a sample that represents diverse software development profiles, individual IT leaders were sorted through random solicitation from software developing organizations. This approach resulted to a legitimate limitation due to the complexity of individual IT leaders as they offered different program options, locations, and differing client consumption. However, the goal of this qualitative study was to assess the need for inclusion of QA in all phases of the SDLC. This inclusion was explored to help in the reduction of software errors that were migrated to production. Another limitation of the study was that data collected for the study were gathered from respondents within each organization. Consequently, the data collected were based on the perceptions and interpretations of the one individual interviewed. This limitation exacerbated the interview because the interviewee was personally responsible within his/her organization for QA resource allocation decisions, preparation strategies, succession planning, and/or placement strategies.

6. Conclusion

The research appeared to indicate a sense that inclusion of QA in all phases of SDLC was an excellent phenomenon. Just like every other good intension, there were still opponents to the adoption of the inclusion. Seventy-two percent of the survey population was of the opinion that QA should be introduced in all phases of the SDLC. The remaining 28% fell into a different category. In a response from a face-to-face interview, one participant succinctly related the unstable economy as a global force against involving QA in all phases of the SDLC. Other trends that affected the inclusion of QA in all phases of the SDLC included (a) too much time spent in the process, (b) inconsistent methodologies, (c) over commitment...
Regarding the timeline to meet market demand, (d) incomplete/nonexistent documentation to enable QA to understand the end user requirements, and (e) too much focus on “getting it out the door.” According to the respondents of this study, only 12 projects (48%) were abandoned due to non-QA inclusion. Due to a high number of defects found, management decided not to migrate. This study showed that QA is recognized as a critical component of the SDLC. IS/IT in today’s complex and dynamic environment was very important. Its development will continue to grow; however, IT will face the challenge of how to successfully develop its product with established methodologies.

7. References


