

## An Empirical Investigation on the Quality Assurance Practices in Software Industries: Bangladesh Perspective

Nasrin Sultana<sup>1</sup>, Dr. Mahbulul Syeed<sup>2</sup> and Kaniz Fatema<sup>3</sup>

<sup>1</sup>American International University-Bangladesh, Dhaka, Bangladesh.

<sup>2</sup>Professional Member, ACM, mahbulul.syeed@gmail.com.

<sup>3</sup>Professional Member, ACM, kaniz.fatema@gmail.com.

**ABSTRACT** – Software companies often follow a planned and systematic process to ensure that the end product is error free and performs as per the requirements, often termed as Software Quality Assurance (SQA) process. However, it has been reported that these companies (especially, mid-size and start-up companies) in developing countries often cut corners to the SQA process due to time and budget limit, and to rush into the market, compromising the quality. This research empirically investigated the SQA process followed in the software companies in Bangladesh. A survey with nine established companies is carried out with qualitative and quantitative data analysis. According to the assessment more than half of the companies fall short in adhering SQA practices within their SDLC. The rest, either fully or partially comply to it. This study further proposed an SQA model based on inspection process and provide guidelines for continuous quality improvement. The model and guidelines are derived to best support these software companies to improve SQA practices.

### ARTICLE HISTORY

Revised: 8 May 2020

Accepted: 27 July 2020

### KEYWORDS

*Software Quality*

*SDLC*

*SQA model*

*Software Companies*

*Inspection*

## INTRODUCTION

Over the past 15 years, software applications have proliferated from sophisticated scientific data processing computations to control and assist our everyday life activities (Galín D., 2004) (Naik K. et. al. 2008). This includes for instance, using a microwave, operating a tv, air-condition, driving a car, making a phone call, using credit card payments and many more. The impact is so much so that we don't even realize that some form of software is running behind each of our ordinary actions (Galín, D., 2004) (Singh B. et. al., 2016).

However, such reliance on software systems often attributed to their safety, security and performance rating that people often wants to make sure of (Falessi D. et. al., 2012). Because, software is a pivotal component of almost every safety-rated system that people use or is a part-of (e.g., automobile, aircraft, industry automation, and many more applications). Therefore, consequences of having undetected defects in these systems may be catastrophic (Maxim B.R. et.al. 2016). In relation to software safety, a catastrophic consequence could be anything between physical hazards to economic hardship (Falessi D. et. al., 2012) (Breu R. et. al., 2014). Therefore, a software system assuring all safety, security, performance and usability parameters can be interpreted as a quality software product.

Comprehending apposite literature on the topic, software quality should adequately address the following two concerns, (a) ensures the identification and redemption of potential hazards that may affect software negatively and cause an entire system to fail (Maxim B.R. et.al. 2016), and (b) the degree to which the software features meet the needs of customers to maximize user satisfaction (Galín D., 2004) (Naik K. et. al. 2008).

Software companies often follow a planned and systematic process to ensure that the end product is error free and performs as per the requirements, often termed as Software quality assurance (SQA) process (Galín D., 2004) (Maxim B.R. et.al. 2016)(Almeida D. et. al., 2003). Statistics shows that for the last two decades, testing as part of SQA process accounts for about 50% of the total time and more than 50% of the total money expended in a software development project. However, it has been reported that software companies (especially, mid-size and start-up companies) often cut corners to this process due to time and budget limit, and to rush into the market.

This research, thus targeted to empirically investigate the SQA process followed in the software companies based in Bangladesh, a developing country which is making its firm footprints in the world software market (Galín D., 2004). The core focus is to investigate the extent to which the software firms adheres to SQA practices within their SDLC process and disseminate best practiced guideline for improving SQA process. Additionally, a light-weight inspection based SQA process has been proposed that will assist these mid-size companies to adopt and improve their SQA process efficiently and cost effectively.

The remainder of this paper is organized as follows, Section 2 details the background and rationale to initiate this study followed by discussion on related works in Section 3, Section 4 describes the key focus and the approach followed

in conducting the research. Survey findings along with analysis are reported in Section 5. Proposal of the SQA model and the guideline are discussed in Section 6. Finally concluding remarks are drawn in Section VI.

## BACKGROUND AND MOTIVATION

Quality should be an inherent property of every man-made artefact that people desires to consume, and software products are no exception to that (Galın D., 2004) (Naik K. et. al. 2008). During the last couple of decades, with a rapid expansion of global competition, outsourcing, along with increasing customer expectations for software products, the concept of quality assurance came to the forefront. Developing quality products to meet the consumer expectations with tighter schedules and competitive budget becomes a decisive factor for the software companies to survive (Maxim B.R. et.al. 2016).

Traditional practice of improving quality mostly centered around the end phase of Software Development Life Cycle (SDLC) through the detection and correction of defects (Galın D., 2004). However, contemporary approach relies on ensuring highest standard in every phase of the SDLC (from requirements analysis to all the way to the final product delivery) in order to ship a quality product that meets customer expectation (Naik K. et. al. 2008) (Maxim B.R. et.al. 2016) (Breu R. et. al., 2014) (Almeida D. et. al., 2003) (Tekinerdogan, B. et. al., 2015). This research investigates the SQA practice based on the above definition. Following aspects of an SQA process is explored (Galın D., 2004) (Naik K. et. al. 2008) in the context of software companies in Bangladesh.

- Paying much attention to customer's requirements.
- Making efforts to continuously improve quality.
- Integrating measurement processes with product design and development.
- Pushing the quality concept down to the lowest level of the organization.
- Developing a system-level perspective with an emphasis on methodology and process.
- Eliminating waste through continuous improvement.

## RELATED WORK

An ample set of research on software quality assurance has been conducted over the past couple of decades. Diverse aspects of the projects, practices, people, resources were analyzed to better understand the core issues related to software quality. Alongside, guidelines, methods and standards are proposed to improve the software quality and benchmark the standard of the quality assurance process. In this section, few of those results are discussed that forms the basis of this research.

The management of a software company plays a pivotal role in forming and maintaining the professional ecosystem for the developers to work and perform by ensuring standards and quality (Hribar, L. et.al. 2009) (Mahbubul Syeed M.M. et.al. 2015). Among others, management should facilitate with the adequate resources, logistics, training and socialization for the team members (Iftikhar A. et.al. 2011) (David L. P., 2003).

Quality assurance plan and practice embedded within software development process is the most important aspect of any quality improvement activity. SQA team managers are responsible and accountable to develop quality plan, its proper execution, and improvement. Several studies proposed *Inspection* method to be practiced for assessment, identification and remedy of defects in each technical artifact produced during the project lifetime (Yau et.al. 1990) (Iftikhar A. et.al. 2011) (Ahad, A., 2017).

Research also focuses on defining cost effective quality assurance process that could ensure customers satisfaction without exceeding the budget and project timing (Naik K. et.al. 2008). This study proposes agile methods to be the best candidates for this optimized quality assurance practices.

Studies to verify the quality assurance practices in software companies in developing countries are carried out by (Iftikhar A. et.al. 2011) (Muazzam Et Al. et.al. 2015) (Javed A. et.al. 2012). In (Iftikhar A. et.al. 2011) and (Muazzam Et Al. et.al. 2015) it is reported that companies in Pakistan often struggle in maintaining software quality to compete in world market. On the track, study in (Javed A. et.al. 2012) identified several key issues that underpinned the substandard quality of the software developed in Pakistan. Few of those issues are, Shortage of Time, Limited Budget, Less use of quality standards, Lack of specialists, Project durations, Compromise on quality due to less profit and Developer's attitude.

## RESEARCH APPROACH

The core focus of this research is three fold: (a) empirically investigate into the SQA practices within software companies located in Bangladesh, (b) analyze and comprehend the investigated data to draw a holistic (both merits and shortcomings) view on the current practices that lead to quality software development, and (c) offer a SQA model and guidelines based on the standards.

To achieve the first goal, we conducted a survey on 9 (Nine) well established software companies based in Dhaka city, the capital of Bangladesh. The survey consists of well formulated questionnaires directly connected to quality assurance process and practices (Galın D., 2004) (Maxim B.R. et.al. 2016) (Javed A. et. al., 2012) (Fagan M.E., 2001) (Ahad, A. et.al., 2017). Collected data is analyzed both quantitatively and qualitatively to draw insight and remarks in attaining the remaining two goals. In rest of this section, we detail this process,

The design of our survey is cross-sectional. We aimed to survey at a fixed point of time. The target audience of our survey is the software professionals (developers, testers, project managers) working in the selected companies.

In order to conduct the survey, we formulated open ended questionnaire. Table 1 lists the set of questionnaires used for this survey. A total of 17 (Seventeen) questions distributed in 8 (Eight) SQA parameters are asked. These categorical questions are elicited based on the guidelines prescribed for quality assurance in software development projects (Maxim B.R. et.al. 2016)(Parnas D. L. et.al. 2003) (Fagan M.E.,2001) (Chaar J. K. et. al. 1993).

The questions are open ended in nature with the provision to express their core belief in close ended form. To be specific, each question is associated with the following two parts to be answered, (a) multiple choice part with option categories: strongly disagree, disagree, strongly agree, agree, and (b) an open field to express opinion in relation to their observation. The questions are kept short, to the point and unambiguous. Each question focuses on one aspect of the SQA practices only. In formulating the questions, standard terms specific to SQA are used for greater clarity and understanding. A concrete guideline on expected answer is accompanied with each of the questions. This arrangement is done to get a detail perspective of the interviewee on a given topic of interest.

The first part of the answer (Multiple-choice) allowed us to get a generic perspective on a given aspect (e.g., is the project delivery time limit is pragmatic?) through quantitative investigation, and the second part (interviewee opinion) allows us to perform a qualitative reasoning on the observation. Then perform an evaluation against the standard to derive recommendations.

To execute this survey, an interactive google form is designed with the questionnaire. This form is accompanied with the detail guidelines to assist the interviewees and navigate through the questionnaire session. The form begins with a common section to record company profile followed by the SQA specific selection. Response is recorded in google sheet categorically which is then extracted and analysed. To complete the survey, approximately 10 to 15 minutes of dedicated time is required. The design of this survey is cross-sectional and are aimed at a fixed point of time. All the companies are contacted well before conducting the survey through official channel, and a *Non Disclosure Agreement (NDA)* was signed to maintain the secrecy and anonymity of the company specific information. The NDA also guarantee to some extent the accuracy of responses as the interviewee feels confident of not getting disclosed. Then the contact information (official emails and phone numbers) of the interviewees from each of the companies are collected. The survey form is sent over the email with clear guidelines and a follow up phone conversation is carried out in case of any clarification is required by the participant.

Collected data is then classified in google sheets based on the research theme. The quantitative data is used to generate graphs for getting general perspective and overall situation on the topic of interest. The qualitative data (i.e., the opinion collected from the respondents) is assessed to complement and reason behind the observations.

Before conducting the survey, a disclaimer is signed between the parties to keep all information and identities of the companies and the participants anonymous.

**Table 1.** Survey Questionnaire and Rationale

| SI No | SQA Parameter  | Survey Questions  | Rationale   |
|-------|--|---|---|
| 1     | Idealistic time limit                                | Did developers get less time for software development?<br>During the development period, do the developers feel the lack of resources?<br>Was the project plan followed by all project members? | To know the extent to which software projects are adhering to legitimate project timing and planning.                             |
| 2     | Dispute of Properties                                | Do individuals perform multiple tasks?<br>Are proper training and required experience considered for development and management activities?   | To verify whether the project resources are utilized and trained as per the standards.  |
| 3     | Project plan and standard are not following properly | Are standards followed in SDLC?<br>Is requirement phase critically analyzed?<br>Is the entire life cycle of software development, strongly supported by planning process?                       | To verify the extent to which the SDLC is adequately put into practice.   |
| 4     | Developer's attitude                                 | Is developer attitude being non-cooperative?<br>Do the project members give less importance in design and coding face?  | To assess if the conventional disputes between developers and testers exists?   |
| 5     | Lack of training and inexperienced team members      | Do the project members have adequate training for performing the assigned task?<br>Do regular training sessions hold for improving performance?   | To check whether the manpower is put together as per the exact need of the project or they just follow an opportunistic approach. |
| 6     | Imbalanced priority to different phases              | Do the allocation of timing and resource to individual SDLC phases suffers discrimination?  | To verify if the testing phase get any importance within the Project.   |
| 7     | Avoiding testing                                     | Are testing rules followed properly?  |   |

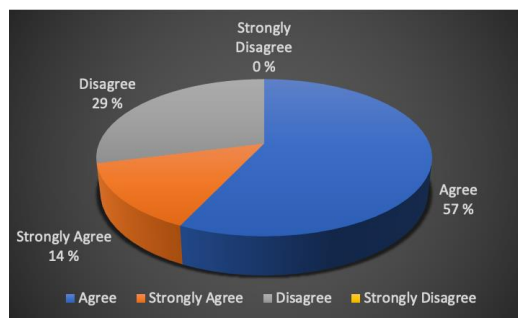
|   |                                      |  |
|---|--------------------------------------|--|
|   | Are product tested as per test plan? | To know if testing is neglected altogether.  |
| 8 | Overlooking project follow up        | Are size, cast and timing documents used for the tracking of the software?<br>Is checklist followed in whole SDLC? |

**RESULTS**

As per the survey outcome, several key issues related to SQA practice has emerged, an elaborated discussion on which is presented below.

**Idealistic Time Limit**

In practice, Project Time is defined as the time to complete a project that includes the complete SDLC process (Galin D., 2004). Developers are bound to deliver the project within this time frame. In reality it is often observed that the QA (Quality Assurance) team gets very limited time to test and review, as most of the project time is consumed in designing and developing it (Parnas D. L. et.al. 2003). This well accepted concern is also reflected in the survey outcome (Figure 1), where 61% respondent agrees (57% agrees and 14% strongly agrees) that the projects’ often have very limited time to develop quality software products.

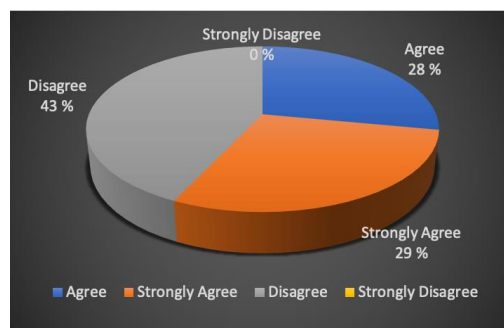


**Figure 1.** Response of Project Time limit.

This observation may be due to the fact that these software companies pay less attention on SQA when they start crossing the deadline or finding it hard to keep the project within budget (Naqvi N. et.al., 2012). Setting an unrealistic deadline may influence projects’ failure to build a quality software (Javed A. et. al., 2012), and many of the software companies in Bangladesh suffers from this issue.

**Dispute on Properties or Resources**

Carrying through a software development project requires adequate resources, which includes but not limited to qualified and sufficient manpower, technology support, equipment, logistics, funding, and training (Galin D., 2004) (Maxim B.R. et.al. 2016). Therefore, deficit of resources often leads to sub-standard product development with reduced quality. Our survey on the track lead to an inconclusive result with slight bias towards inadequacy of required resources. As shown in Figure 2, 43% of the respondent denied of having any deficiency of resources, with 53 % (29% strongly agree and 28% partially agrees) with the concern of deficiency.



**Figure 2.** Response on Resource Disputes within the projects.

**Non-Standard Project plan & Process**

Success of a development project depends on the project planning (Maxim B.R. et.al. 2016) (Pankaj J., 2005) (Ahad, A. et.al., 2017) A well evaluated plan contributes heavily towards the projects’ success. A properly designed project plan helps the management team to deliver proper information and guideline to the right people at the right time. This in turn

creates a strong communication channel with a feedback mechanism among the concerned people. This helps to understand the customer requirements better, and design, develop the software properly.

However, small software companies in developing countries often fails to realize the necessity of project planning, communication and co-ordination within the team. They are always in a hurry to build the software without even knowing complete customer requirements (Pankaj J., 2005) (Naqvi N. et.al., 2012). The members rarely get a clear picture of what is expected from them as they work on the project. This approach might cause a serious problem in developing quality software. The survey result (Figure 3) on the track reported that 54% of the software companies in Bangladesh are reluctant in going for an extensive project planning and its' implementation during the project lifetime.

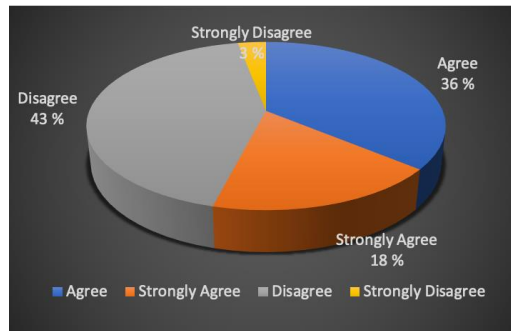


Figure 3. Response regarding Project planning and process management.

### Developer's Attitude

Poor communication and coordination is one the biggest challenges in software project management (Jackson, S.L., 2011). And while it involves the SQA team and the developers, the issue gets even worse. According to study (Javed A. et. al., 2012), a common allegation of the SQA employees is that developers' mostly do not show cooperating behavior. Developers' think that they are right on what has been developed, and that the SQA people suffers from lack project specific knowledge to assess the quality. Our survey along the track conforms to this observation (Figure 4) with 73% of the respondent agrees that coordination and support from the developers is a core concern in doing SQA activities.

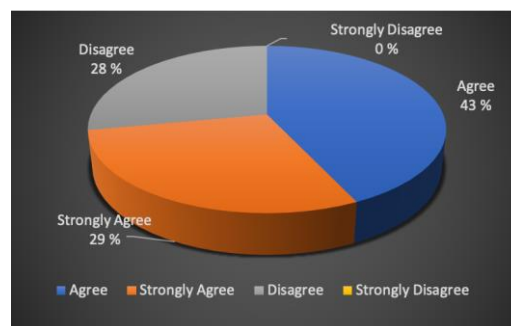


Figure 4. Response regarding developers' attitude towards SQA team.

### Lack of Training and Experience

Team members are often assigned in projects based on their availability, not based on their expertise and experience. This is a common practice in small or mid-size software companies where resource scarcity is a reality. Domain related experience and expertise are often overlooked for development and management activities due to time and budget constraints, and to maximizing profitability. Additionally, many small software companies don't have dedicated SQA team. Rather the same has been done by the developers who developed the system (Javed A. et. al., 2012). However, this practice leads to substandard product development due to several associated concerns., (a) developers are often not trained enough to extensively test the product, (b) new technologies and tools are not adopted and trained before putting them in use, (c) human biases to find errors in own code ( Pankaj J., 2005).

Approval and strong approval on these ill practices in software companies are reported by 57% of the respondent (Figure 5). Therefore, more than half of the companies in Bangladesh might suffer quality standard due to lack of training and experience. Additionally, 72% of the respondent agreed that testing is carried out by the people who do not have adequate technical and procedural knowledge on testing, rather they depend on their self-educated knowledge and judgement in conducting the task.

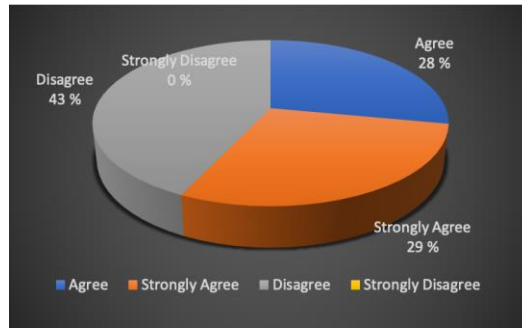


Figure 5. Feedback on training of developers.

### Disproportionate Distribution of time among SDLC Stages

Extensive requirement analysis is one of the most crucial task of software development. Project success heavily depends on the rigor of requirement analysis, understanding and documentation (Pankaj J., 2005). However, due to time pressure, and budget constraints, management often jumps into the design and development well before understanding the requirements. Around 43% of the respondent (Figure 6) adhere to this issue that may affect the quality of the end product.

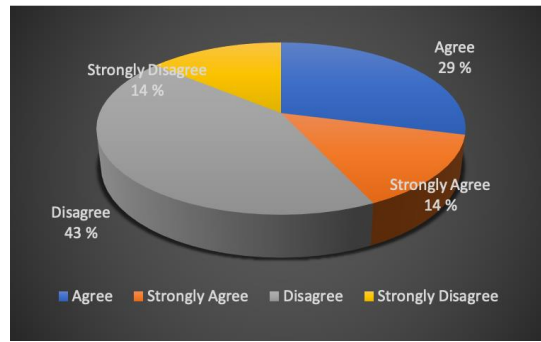


Figure 6. Response on discrimination between the stages of SDLC.

### Bypassing Testing

As per our survey result (Figure 7), around 47% of the respondents agreed of by-passing SQA practices in some way, whereas, 53% either disagree (or strongly disagree) of avoiding testing altogether. Therefore, this issue remains inconclusive. However, it is agreed by the majority that quality assurance has been neglected either entirely or partially. More often managers keep on postponing testing until late in the development. Many of the firms also tailored down the test plan, bypassing the testing rules, standard and practices which contributed to low product quality.

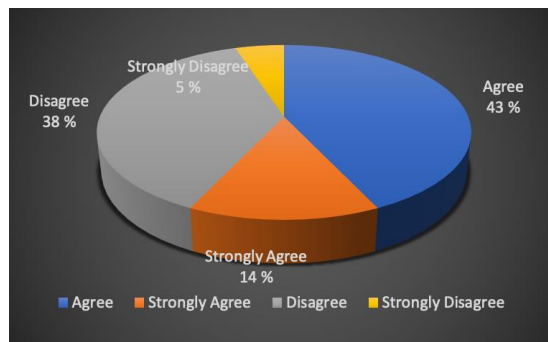


Figure 7. Feedback on avoiding testing.

### Overlooking Project Follow up

Tracking project progress and doing check-and-balance between the plan and actual work done with quality assessment is the key to a project success (Ericson C. A., 2005). In doing so, firms need to put into practice proper project planning, estimation of budget, resources and contingency planning and its proper execution.

The investigation on this track reveals inclination towards good project follow-ups. 57% of the respondents (Figure 8) reported that the projects in which they participated followed a standard method of tracking project progress to keep the work on track and ensure timely delivery.

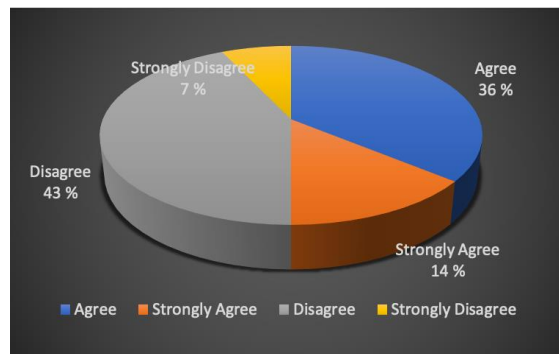


Figure 8. Response on project follow-up process.

## SYNTHESIS AND RECOMMENDATIONS

This study is carried out considering the quality assurance process presented in Section 2. This process emphasizes few core concerns of quality assurance, e.g., quality control in every stage of SDLC, continuous improvement and customer satisfaction. The survey questionnaire was set to investigate the same within the context of software companies in Bangladesh. According to the survey transcript (discussed in Section 5) more than half of the companies fall short in adhering SQA practices within their SDLC. The rest, either fully or partially comply to it.

Overall survey outcome strongly suggests that software companies in Bangladesh either need to enhance or adopt the SQA practices. Therefore, these companies need to furnish their practices, adopt the best prescribed SQA methods in literature and follow a guideline for continuous improvement along the path to reach the ultimate quality standard. In the rest of this section, a SQA model based on inspection process is proposed. Alongside, a detail guideline for continuous improvement is also prescribed.

### The Quality Assessment Model

The proposed SQA model is based on Inspection method (Fagan M.E., 2001) (Parnas D. L. et.al., 2003). According to this study, this model is light-weight, easy to learn and practice. The model is termed as *Quality Inspection Model (QIM)*. QIM should be planned and executed at the end of each SDLC phase. Therefore, QIM is involved in assessing the quality of the deliverable throughout the project life cycle (from requirement analysis to product delivery) and thus ensuring product quality. Figure 9 presents the QIM model in relation to a typical SDLC process.

To carryout QIM in every phase of SDLC, a core team is formed consisting of internal stakeholders and external domain expert. The team is kept minimal to reduce unnecessary overhead. The core focus of the inspection is to,

- Assess the quality of the project progress (in terms of technical report, documentation, code walk-through);
- Identify scope of improvements (e.g., misinterpretation, code quality, bug tracing, and others), and
- Prescribe action for improvement with specific deadline and checkpoint.

For this, the project manager and the management must have detail project plan with specific milestones. Individuals or team of experts are assigned to meet the milestones. At the end of every milestone a buffered time should be there to carry out the QIM to ensure milestone quality. Specification of the QIM process is detailed below:

**Roles in Inspection:** Following roles are defined for the inspection,

- Moderator:** Moderator is the planner of the inspection (Chaar J. K. et. al. 1993). In QIM, a moderator should be the project manager who has a detail knowledge of the project plan, milestones and task distribution. Therefore, he would be the best person to coordinate and lead the inspection session. In addition, he can cross verify the inspection outcome / findings with the involved parties (as presented next), document them and set the next course of actions in resolving the issues that are potential threat to quality.
- Document Owner:** This is the technical person (e.g., designer or developer) who owns the content to be inspected, e.g., design document or part of code module (Chaar J. K. et. al. 1993) (Parnas D. L. et.al. 2003). He is the technical representative from the development team who will explain, argue, defend the job done with the Principle Inspector. His sole motive is to support the inspector in a constructive way to make a thorough assessment of the content and find out as much issues as possible. By default, he is the person chosen by the Moderator to resolve the identified issue(s).
- Principle Inspector:** This is the professional external to the organization who is highly skilled, experienced and have in-depth technical know-how on the domain under consideration (Laitenberger O. 2002) (Chaar J. K. et. al. 1993). Therefore, for each stage of SDLC the appropriate principle inspector must be selected, e.g., requirement analyst, experienced architect, senior developer and so on. Being external to the organization and the team, his sole responsibility is to track down all possible flaws unbiased during the inspection session and offer a guideline for resolving those.

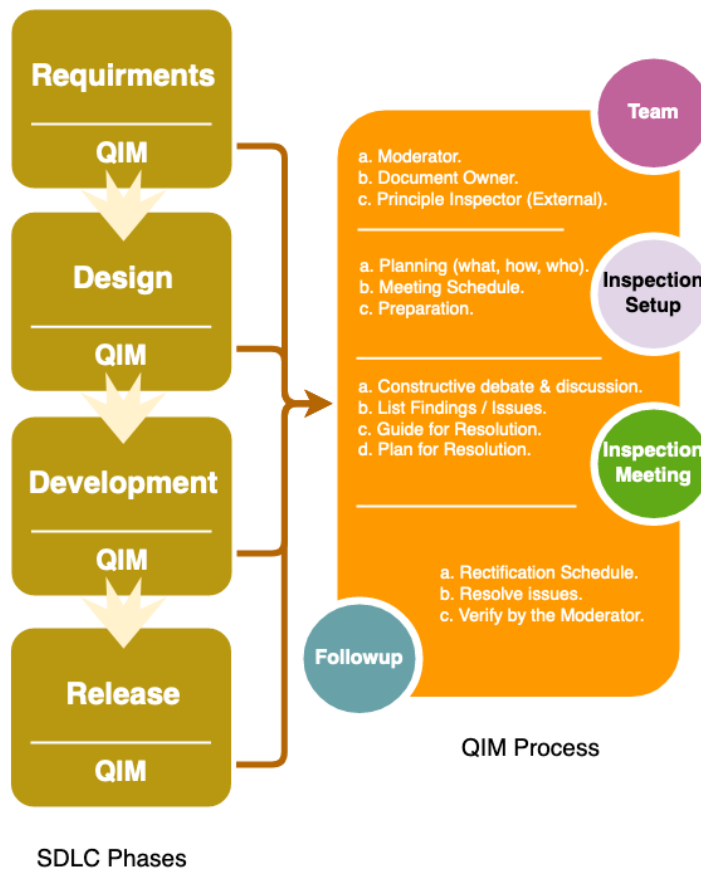


Figure 9. The QIM process within the context of SDLC phases.

**Inspection Activity:** Following activities are performed during an inspection.

- a) **Planning:** This phase is used to determine what to be inspected, setting passing criteria, schedule and the inspection team. The Moderator selects the document owner based on what to be inspected and a principle inspector. The moderator collects the content to be inspected and finalize the plan for meeting.
- b) **Meeting Overview:** The moderator with the help of the document owner prepare a meeting minutes detailing the agenda, content and schedule of the inspection and send it to all the meeting stake holders.
- c) **Preparation:** In this phase each team member individually prepares for the inspection. Each team member reviews its assign work product and note down the observations before the inspection meeting. The principle inspector carries out a detail investigation of the content.
- d) **Inspection Meeting:** This activity is led by the moderator. The key discussion is held between the external principle inspector and the document owner. Through a constructive debate and discussion all the issues are identified and documented for rectification. Alongside, a guideline for resolving the issues are determined.
- e) **Follow-up:** Based on the rectification plan, the document owner corrects the defects issues, and the Moderator verifies the same to ensure the quality.

### Key Features of Quality Inspection Model (QIM)

It is to be noted that QIM is not a novel one per-se, rather it has been adopted in a tailored version from the established practices for conducting inspection. Essence of an effective quality inspection process has been kept intact. Therefore, this model will offer several benefits to startup and mid-range software companies in developing countries under the following capacity:

- a) QIM is light-weight in-terms-of resource, time and budget required. The process can be carried out with minimal number of people involved, with little or no additional budget spend and can be easily blended within the conventional SDLC process. The advantage is that with only core stakeholders in the meeting most effective outcome can be achieved.
- b) The process is easy to learn, adopt and practice. The learning curve is flat considering anyone having good knowledge of SDLC can practice it. Adoption within the organizational practice should be seamless, as no excessive adjustment is required from method perspective.
- c) Organizations having little or no monitoring on the standards, can deploy this method to gradually inject best practices to follow in every level of the organizational activities, establishing standard, and increase accountability.



- d) Help building the culture of quality assurance within the conventional practice of SDLC. Additionally, having an external reviewer will keep pushing the employees to produce their best.
- e) This model will assist in identification and resolution of the faults/defects while the system is under development. According to conventional wisdom, this will substantially reduce the cost of product production and maintenance, as defects detected at earlier stage of development are always easy to resolve.

### Adoption of Good Practices

Adoption of Good Practices In addition to the quality assurance model (QIM), the software companies should focus on the development of professional practices. It will enhance the overall quality of the development process, and hence the product quality.

**Attitude and Skill Development:** Software professionals (Developers, Testers, Project managers and the management) need to realize the fact that all the stakeholders in the team are leading towards a common goal. And that is, making a quality software product to achieve maximum customer satisfaction. Therefore, they are not rivalling in nature, rather complements each other. All parties need to take their responsibility at highest professional terms, assist each other in getting any issues (e.g., bugs, design defects) fixed within a healthy professional atmosphere and within time.

Project managers as well as the management must hold periodic meetings and social team building events to boost confidence, trust, and interpersonal communication and coordination among the team members (Pankaj J., 2005). Software companies regardless of their size should strive for having a dedicated SQA team to do the quality control. However, in reality they often overlook this part (mostly due to budget and time restrictions) and carry out partial testing with the developers themselves. However, this is a requirement for every organization to build up a SQA team with adequate experience and expertise on contemporary methods and tools to carry out quality assurance process effectively (Tuteja M. et. al., 2012).

For startup software companies in developing countries, like Bangladesh, can start the SQA practice with the adoption of QIM model. And as they grow in maturity, a dedicated SQA team can be formed. Additionally, training of the employees on contemporary tools, techniques, methods and processes are very crucial for a software company to sustain. Software world is a very rapidly evolving domain, and keeping the manpower align with this ever-evolving technology is a must for the tech companies to grow and mature.

**Project Planning and Execution:** Following issues in relation to planning and execution of a Software projects must be handled with great care. We are highlighting activities based on best practices prescribed in relevant literature (Pankaj J., 2005) (Fagan M.E.,2001) (Laitenberger O. 2002) (Galín D., 2004).

Defining a *pragmatic project Timeline* is the key factor for projects' success. A rigorous analysis of the requirement, and a detail project plan incorporating project phases, deadlines, risk analysis and mitigation plan must be done to decide on the project tentative timeline. This estimation will assist management to have a good grab on the project progress, quality control and budget.

Allocating *Resources* adequately for the project is another key factor to success. Resource for a project includes but not limited to manpower, budget, planning, schedule, risk management and monitoring. A detailed project planning as discussed above would support in allocating resources in an efficient and timely fashion to maximize quality output.

With a project up and running, the project manager and the management must run a mechanism to inspect the project progress and quality assurance on regular intervals. An integrated inspection process with the defined SDLC would be good practice to assuring quality. The proposed light weight QIM process can support the software companies in Bangladesh to initiate the process in a cost-effective way without allocating extra resources.

Finally, explicit and rigorous testing is a must for producing quality software. Therefore, having a testing phase within the SDLC with dedicated resources is required. Management should give attention on the track to improve software quality and increase organizational capability and reputation.

### CONCLUSION

Contemporary literature views SQA as a continuous process of ensuring quality in every stage of the SDLC process. This research empirically investigates the extent to which the prescribed practices are followed in software companies in Bangladesh. Analysis of the collected data reveals that more than half of the companies are reluctant in following any quality improvement process, leaving others with the scope of improvement in several key sectors.

This study further proposed a lightweight SQA process based on inspection method and considering the need and capacity of these companies. Additionally, best practices for continuous quality improvement are also prescribed. Future research in this direction requires imperial verification of this model for its effectiveness.

### REFERENCES

- [1] Yau, S.S., Wang, Y.-W., Huang, J.G., Lee, J.E. (1990). An integrated expert system framework for software quality assurance, Computer Software and Applications Conference. COMPSAC 90.Proceedings.
- [2] Hribar, L. , Burilovic, A., Huljenic, D. (2009) Implementation of the Software Quality Ranks method in the legacy product development environment” Telecommunications. ConTEL 2009. 10<sup>th</sup> International Conference on 8-10 June 2009.

- [3] Iftikhar A., Muhammad S.A. (2011). Software Quality Assurance A Study Based on Pakistan's Software Industry, *PJETS Volume 1, No 2, 2011*, pp: 65:73.
- [4] Javed A., Muazzam et al. (2012). Software quality assurance improvement in software market, *Advanced Computing: An International Journal (ACIJ)*, Vol.3, No.2, March 2012, pp: 17-29.
- [5] Muazzam Et Al. (2015). How To Improve Software Quality Assurance In Developing Countries, *Advanced Computing: An International Journal (ACIJ)*, Vol.3, No.2, 2012, Pp: 15-27.
- [6] Galin D. (2004). *Software Quality Assurance - From theory to implementation*, Pearson Education Limited.
- [7] Naik K., P. Tripathy. (2008). *Software Testing and Quality Assurance, Theory and Practice*, John Wiley & Sons, Inc.
- [8] Maxim B.R., Kessentini M. (2016). An introduction to modern software quality assurance, Chapter 2, *Software Quality Assurance*, Elsevier, Pages 19 – 46.
- [9] Pankaj J. (2005). *Software Project Management in Practice*, Addison-Wesley Professional.
- [10] Almeida D., Camargo, Basseto A.B., Paz M.S. (2003). Best practices in code inspection for safety-critical software, *IEEE Software*.
- [11] Ericson C. A. (2005) *Hazard Analysis Techniques for System Safety*, John Wiley & Sons, Inc.
- [12] Mahbulul Syeed M.M., Lohman A., Mikkonen T. and Hammouda I. (2015). Pluggable Systems as Architectural Pattern: An Ecosystemability Perspective. In *Proceedings of the 2015 European Conference on Software Architecture Workshops (ECSAW '15)*. Association for Computing Machinery, New York, NY, USA, Article 42, 1–6.
- [13] Falessi D., Sabetzadeh M., L. Briand, Turella E., T. Coq and Panesar-Walawege R. K. (2012). Planning for Safety Standards Compliance: A Model-Based Tool-Supported Approach, in *IEEE Software*, vol. 29, no. 3, pp. 64-70.
- [14] Breu R., A. Kuntzmann-Combelles and Felderer M. (2014). New Perspectives on Software Quality [Guest editors' introduction], in *IEEE Software*, vol. 31, no. 1, pp. 32-38.
- [15] Tekinerdogan B., Grundy, J., Ali, N., Soley, R., Mistrik, E. (2015). *Software Quality Assurance*. United states; Morgan Kaufmann.
- [16] Galin, D. (2004). *Software Quality Assurance*. United kingdom; British Library Cataloguing-in-Publication Data
- [17] Singh B., Gautam, S. (2016). The Impact of Software Development Process on Software Quality. *International Conference on Computational Intelligence and Communication Networks (CICN)*; vol.10, pp:09-10
- [18] Tuteja M., Dubey, G. (2012). A research study on Importance of Testing and Quality assurance in software Development Life Cycle(SDLC) Models. *International Journal of soft computing and Engineering*; vol.2, pp-3
- [19] Naqvi N., Dr. Khan, A., Khan, A. (2012). Survey to improve software quality assurance in developing countries. *International Journal of Technology and Research*; Islamabad, Vol.3, pp:17-29
- [20] Javed A., Maqsood, M., Quazi, K., Shah. (2012). How To Improve Software Quality Assurance In Developing Countries. *Advanced Computing: An International Journal ( ACIJ )*, Vol.3, pp:2.
- [21] David L. P., Lawford M. (2003). *Inspection's Role in Software Quality Assurance*, McMaster University.
- [22] Ahad, A., Ullah, Z., Tariq, L., Niaz, S. (2017). Software Inspections and Their Role in Software Quality Assurance. *American Journal of Software Engineering and Applications*. Vol. 6, No. 4, 2017, pp. 105-110.
- [23] Jackson, S.L. (2011). *Research Methods and Statistics: A Critical Approach*, 4th edition, Cengage Learning, p.17
- [24] Fagan M.E. (2001) *Advances in Software Inspections*. In: Broy M., Denert E. (eds) *Pioneers and Their Contributions to Software Engineering*. Springer, Berlin, Heidelberg.
- [25] Parnas D. L. and Lawford M. (2003) The role of inspection in software quality assurance, in *IEEE Transactions on Software Engineering*, vol. 29, no. 8, pp. 674-676.
- [26] Chaar J. K., M. J. Halliday, I. S. Bhandari and Chillarege R. (1993) In-process evaluation for software inspection and test, in *IEEE Transactions on Software Engineering*, vol. 19, no. 11, pp. 1055-1070.
- [27] Laitenberger O. (2002). A survey of software inspection technologies, *Handbook of Software Engineering and Knowledge Engineering*, pp. 517-555.