IUG Journal of Natural and Engineering Studies
Vol.22,No.1, pp 167-186 **2014**, ISSN 1726-6807, http://www.iugaza.edu.ps/ar/periodical/

Evaluating Software Quality Assurance Practices In Gaza Strip

Alaa M. El-Halees
Faculty of Information Technology, The Islamic University of Gaza, Gaza, Palestine
alhalees@iugaza.edu.ps

Mohammed Alnajjar
Faculty of Information Technology, The Islamic University of Gaza, Gaza, Palestine
mohammed_el_najjar@hotmail.com

Abstract: It is important to understand the degree developers in Gaza Strip perform standard Software Quality Assurance (SQA) activities. This would allow customers of the software to better understand the quality risks, if there is any. Further, it will help the developers identify process improvement opportunities to produce higher quality software.

The purpose of this paper is to measure the effects of using software engineering practices on the productivity and the quality of the software. The study shows the variation of using these practices from doing project management, requirements gathering methods, documentations, using coding standard and testing. Also, it recommends some directions to improve the quality and productivity of the software in Gaza Strip.

This paper presents the results of a survey to examine performing SQA through software development projects in organizations in Gaza, primarily centered using software engineering principles to support quality assurance process. The paper concludes that most of IT companies and organizations in Gaza Strip do not apply the quality assurance activities during the entire software development cycle, which leads to produce low quality software. It also concludes that most of Software organizations do not use common software engineering practices as required, the usage of software standards varied among of them, as most of them does not make documentation to their projects, and the testing process is mostly negligent.

Keywords: Software quality assurance, testing, software in Gaza Strip.

تقييم ممارسات تأكيد جودة البرمجيات في قطاع غزة

ملخص: ممن المهم فهم كيف يقوم مطور البرمجيات بممارسات تأكيد الجودة في قطاع غزة ، لأن ذلك يساعد المستفيدين من الأنظمة في فهم المخاطر التي يمكن حدوثها   أثناء إنتاج البرامج. وكذلك يساعد المطورين على إيجاد المناطق التي يمكن تطويرها لإنتاج أنظمة عالية الجودة.

الهدف من هذه الدراسة هو قياس تأثير استخدام ممارسات هندسة البرمجيات على   إنتاجية وجودة البرمجيات. الدراسة تظهر اختلاف الممارسات في عمليات إدارة المشاريع البرمجية ، طرق الحصول

على المتطلبات ، توثيق الأنظمة،استخدم المقاييس في كتابة الكود وفحص البرامج. وكذلك تقوم الدراسة بتقديم التوصيات اللازمة لتحسين الجودة والإنتاجية للبرمجيات في قطاع غزة.

هذه الورقة تقدم نتائج استبيان لفحص كيفية استخدام أنظمة الجودة في مشاريع البر مجيات في قطاع غزة. ومن استنتاجات هذه الدراسة أن معظم المؤسسات والشركات التي تنتج برمجيات لا تقوم بممارسة تأكيد الجودة في جميع مراحل إنتاج البرمجيات، وهذا أدى إلى إنتاج برمجيات ذات جودة منخفضة. وكذلك تم الاستنتاج أن معظم المؤسسات والشركات المنتجة للبرمجيات لا تقوم بممارسات هندسة البرمجيات مثل استخدام المعايير البرمجية ، توثيق البرامج وكذلك يتم الاستغناء عن الفحص في معظم الأحيان.

1. Introduction

Software Quality Assurance (SQA) is the combination of the entire software development process, which includes software design review, source code control, code review, change management, configuration management and testing [Schulmeyer, 2008]. The goals of SQA are to ensure that the project will be completed based on the approved specifications, standards and functionality and meet the customer needs.

Lately, the software development in Gaza has grown rapidly for several reasons such as: the graduation of large numbers of programmers and engineers with different skills in software development, the organizations directed to use IT in their work, availability of computers and internet in most homes, offices and organizations. Most of software used in Gaza are either developed locally by companies and IT units in the organizations or by individuals. Now we can see in every organization, ministry or institute that there is an IT unit responsible for developing applications to the organization, and monitoring the operation of these applications. Most of these units have common structure as manager, analyst and programmers. Because of this growth, we need to pay more attention to the quality of software produced.

But if we take a quick look to their work practices, it can be noticed that they neglect performing quality assurance practices. In other words, organization seldom applies all software engineering practices and methods that affect the quality of delivered software. For example, some of them make requirements specification documents and design analysis but do not do project planning, meanwhile, other do project planning but do not document analysis process and so on. So these missing practices cause a lot of modification in the software by the customer, software submission delay and other problems that make the software low in quality. In addition, there is no interest in the testing process by the local companies and organizations in Gaza. It is noted that many problems occurred when the products are released for use.

The purpose of this research is to measure the effects of using software engineering practices on the productivity and the quality of the software in Gaza Strip. To do that, a survey has been conducted for the SQA practices used by the developers in Gaza. The objectives of the survey are to know if the quality of practices and testing are used through development process by the developer; to determine whether some factors have an impact on the implementation of these practices (developer's experience, educational background, size of the software product, and number of users).

The rest of this paper is structured as follows: section two will be related work, section three will discuss our survey methodology, section four will give an overview of the people who participated in the survey, section five will give the survey results, and section six will concluded our work and will give some recommendations.

2. Related Work

There is no previous work on SQA in Palestine in general and Gaza Strip in particular. However, there are many in other parts of the world. For example, Iftikhar and Ali (2009) investigated the role of quality management practices in software industry of Pakistan. They presented a comparison between the more-experienced and the less-experienced firms with respect to the critical factors of quality management. The results of the research indicate that the "age of quality" in Pakistan software industry has a very limited influence over Software Quality Assurance. Also, in Pakistan, Javed et. al. (2012) addressed the problems for lacking interest in improving the software quality by higher authorities and software assurance team.

Ow and Yaacob (1996) conducted a survey on software Quality Assurance in Malaysia. The survey investigated the trends in SQA investments with respect to development purposes. In their work, the QA problems encountered have been addressed from three main sources: people, management and technology. They found that there is a significant increase in awareness of the importance of software quality even by those companies which developed software systems for internal applications.

To develop high-quality software, Lee et. al. (2009) conducted a survey with a wide variety of companies and experts that were involved in the software testing in order to identify the current practices and opportunities for improvement of software testing methods and tools (STMTs). The survey results revealed five important findings regarding the current practices of STMTs and opportunities for improvement: low usage rate of STMTs, difficulties owing to a lack of STMTs, use of testing tools in a limited manner, demand for interoperability support between methods and tools of software development and testing, and the need for guidance to evaluate STMTs or to describe the capabilities of STMTs.

Yasuda and Yamada (2002) discuss the concept and practice of software quality assurance in Japan. They discussed which they called "ZD-type" products which seek high reliability software and CS-type products which seek customer satisfaction.

3. Survey methodology

The survey was carried out by filling an Electronic form. The form had a total of 50 questions about performing quality assurance and applying software practice that affect the quality. It is divided into six parts: general information about the Participant's characteristics, SQA, software engineering practices, project planning, peer review and testing.

Two experts from the Faculty of Information Technology at the Islamic University of Gaza were asked to validate the questionnaire. Experts' comments and suggestions were collected and evaluated carefully. Some modifications and additions were considered to the questions and the final questionnaire was constructed.

The questionnaire was available for 20 days. It was announced to software development groups by means of discussion lists and direct email messages sent to professionals and institutions. The parsed contents of answer were analyzed statistically using excel 2007.

4. Participants of the Survey

The Participants in the survey included UNRWA Organization, IT units in local universities, IT units in some ministries and local companies. The size of these Participants varied from low size organization to large size organizations. Sizes are: Very Small (2 – 10 Employee), Small (10 – 30 Employee), Medium (30-500 Employee), Large (>500 Employee). Figure (1) shows the percentage of participants in each size.

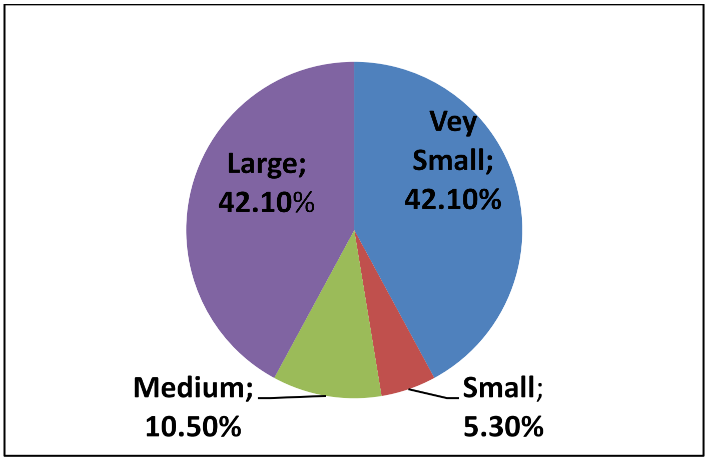

Figure 1: Participants Organizations Size

These organizations classified as 50% are profit organization and 50% are nonprofit organization. It was found that 72.2% of these organizations develop applications for its own use, and 27.8% produce software for others. Figure 2 shows the distribution of the job responsibilities in the sample. Since the participants may have more than one role, it is clear that most of them were programmers.

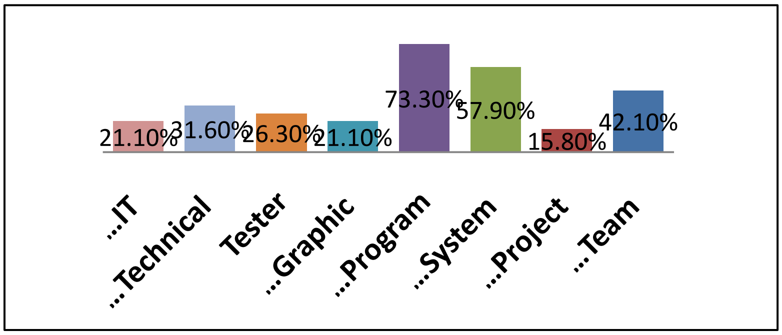

Figure 2: Participants roles Software Development process

Also, the sample shows that 63.2% of the participants are responsible for other team members in their institutes. 33.2 % of these participants are responsible for 2 employees, 33.3 % of them are responsible for 3 employees, 11.1 % of them are responsible for 7 employees, 11.1 % of them are responsible for 9-12 employees, and 11.1 % of them are responsible for 12-15 employee.

The questionnaire found that 89.5% of the participant's works on different projects at the same time and 10.5% work on one project only. The questionnaire also asked about the years of experience in the software development. Table (1) shows that most of them works in less than five years.

Value	Percent %
Less than one year	21.1%
Two - three years	26.3%
Three - five years	31.6%
More than five years	21.1%

Table 1: Participant years of experience

The survey asked about nature of the application that developer work on it. is the answers varied among respondents, some of them work on developing desktop application, others work on developing web applications, mobile applications or other applications. Some of them work on more than one type of these applications as shown in table 2.

Value	Percent %
Desktop applications	73.7%
Web applications	68.4%
Mobile applications	5.3%
Others	10.5%

Table 2: Types of applications

5.0 Survey Results

This section presents the results of the survey that tested Software Engineering application methods and practices in Gaza area to support software quality assurance process, and how it is used within IT units in the different institutes.

5.1 Quality Control

According to the survey, most organizations (67 %) do not have a unit responsible for quality assurance. Also, it is noticeable that some of these units are not fully dedicated to the quality assurance. About the nature of the project that developed locally in Gaza. The survey indicates that most of software developers (55.6%) work on developing simple and complex projects and the rest just develop simple projects.

In addition, most of the respondents (88.9 %) believe that the Software Quality Assurance is an important part of software development process. Regarding the question of whether they have defined metrics to measure the software quality, most developers (66.7%) are found to be concerned with developing long life products. The research tried to investigate the problem of late submission of the developed projects by the local organization in Gaza since it affects the quality of the produced software. As in figure (3), 55.6% of the respondents said that the average late is 10% of the delivered projects, 33.3% of them said the average late is 33.3% and 11.1% of the them say the late average is more than 50% of the projects .This means that the quality will be low according to time limitations and unplanned actions.

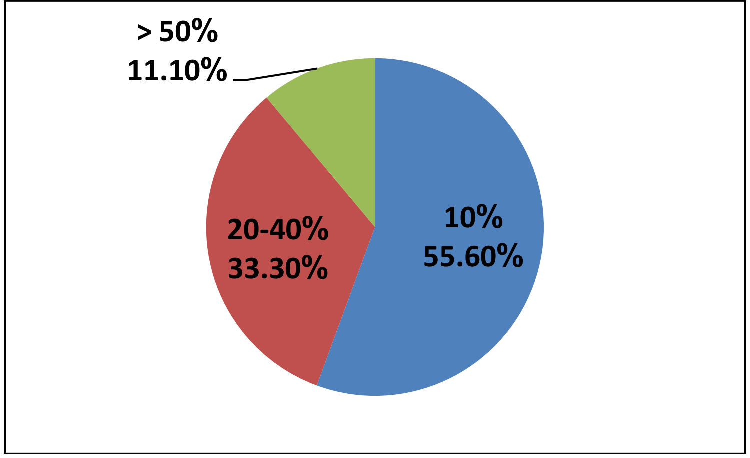

Figure 3: The average submission late

Also, we tried to know the reasons for the late submission, as shown in table (3); most of them (77.8%) said that the main reason is that “the customer does not know what he/she wants”. And the second reason was "repeated modification in the requirements". This means, as the participants believe, that the main reason for late submission is the customer understanding of the system.

Value	Percent %
Poor requirement definition	33.3%
The customer doesn't know what he/she wants	77.8%
Repeated modification in the requirements	55.6%
The development team don't have sufficient experience in the project area	11.1%
The customer did not offer the required resources	55.6%

Table 3.The Reasons for the late submission

About the productivity of our organizations in Gaza, the survey indicated that 44.4% of these organizations produce a project every four months, 22.2% of them produce a project six months, and 22.2% produce a project every year. About the average number of users for the developed projects, 44.4% of the respondents said the users for their projects are more than 60, as shown in figure (4). Accordingly, it can be concluded that systems built in Gaza Strip are mostly small.

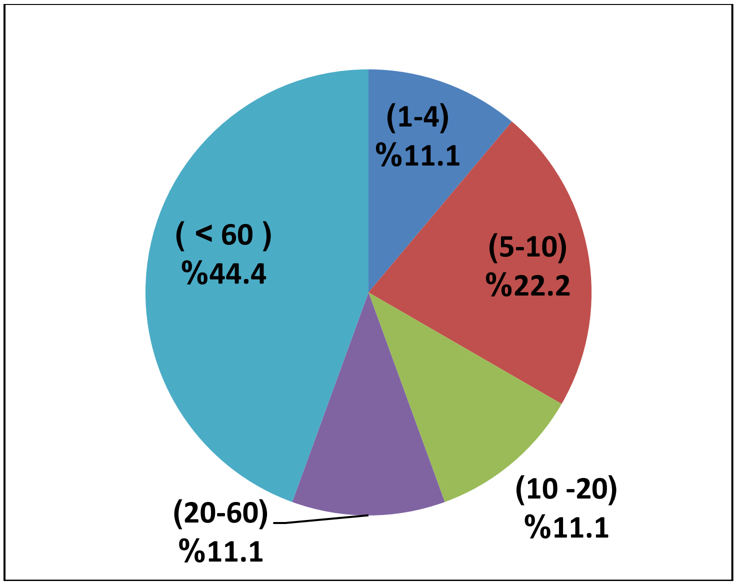

Figure 4: The average number of users

5.2 Applying software engineering practices to support quality assurance process

We asked the participants to comment on applying software engineering practices to support their quality assurance process and we received the following answers:

5.2.1 Requirements Analysis:

Requirements affect the quality of delivered software, since the poor requirements definition lead to low quality software that did not meet the customer needs. Therefore, we designed a set of questions to examine the usage of some requirements analysis activates in software development life cycle. The process began from validating the requirements elicitation and towards producing software requirements specification document. The questions measured the usage of these activities, and the results varied among the participants.

We asked the participants about doing analysis for the developed system before coding and we found that 55.6% of them do analysis before coding, and 44.4% rarely make analysis. Hence, it can be concluded that the one reason many systems failed is because of the lack of analysis.

About using IEEE standards for making software requirements specification report, it was found that 33.3% of them always use it, 44.4 % rarely use it, and 22.2 % never used it.

The most important question was about using requirements elicitation techniques; we found that most of developers (72.2 %) use interviews for this regard. Figure (5) summarizes these results. We can conclude that the participants use the easiest and the most familiar way of requirement elicitation which is interviews. Therefore, for more successful software systems, they need to pay more attention to the requirement elicitation step by using more sophisticated methods.

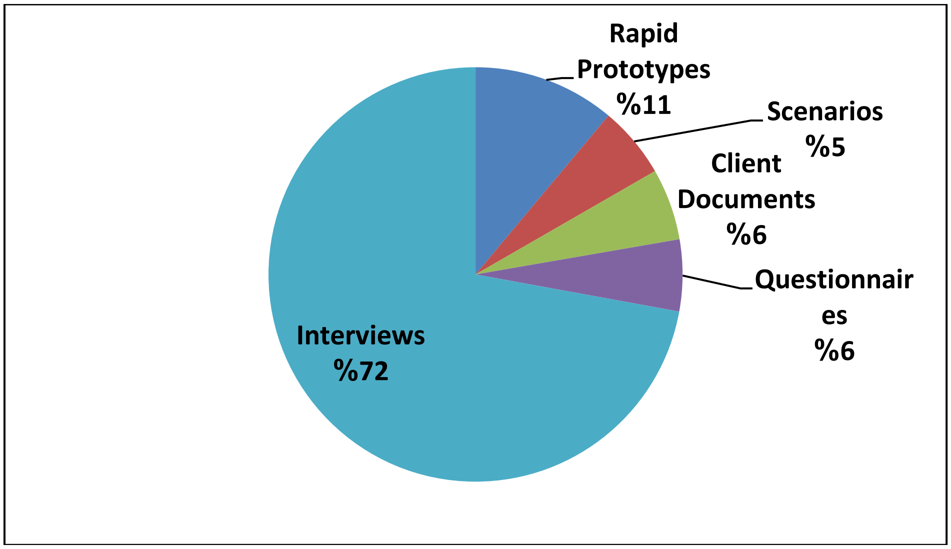

Figure 5: Requirements elicitation techniques

In case of ambiguous user needs, 89.5% of these users propose prototype of the developed system and give it to the customer, and then take the feedback to extract the requirements.

In addition, we investigated some issues related to requirement analysis process as user's involvement in requirement analysis process, we found that that 57.9% expressed that the lack of user's involvement in requirements process affects the speed of software production, and 55.6 % of them said that the poor communication between developers and customers may lead to poor understanding of requirements, poor user-interface design, 52.9 % of them said that the analyst experience is an essential factor in requirement analysis process to extract most of the user's requirements, 33.3% of them said that collecting all the requirements from the customer before starting the development process, 38.9 % of them use the standards in requirements documentation in their organization, 47.4% of them communicates with the client to get approval on the defined requirements. Table (4) shows the distribution of responses to this question in the sample.

Question	Rarely (<25%)	Sometimes (25-49%)	Often (50-74%)	Almost always (75%+)

Lack of user involvement	10.5%	15.8%	15.8%	57.9%
Poor communication between developers and customers	11.1%	11.1%	22.2%	55.6%
The analyst experience	0.0%	11.8%	35.3%	52.9%
Collect all the requirements before start development	5.6%	22.2%	38.9%	33.3%
Use standards in requirements documentation	16.7%	16.7%	27.8%	38.9%
customers change the requirements after development	16.7%	27.8%	16.7%	38.9%
The developer communicates with the client to get approval	10.5%	10.5%	31.6%	47.4%

Table 4: Requirements analysis process

We can conclude that the lack of relation between the user and the developer is the reason for requirement process failure.

5.2.2 Project Planning

Project planning is important to get high quality software that is delivered in the estimated schedule and budget and meet the customer's requirements. The survey designed a set of questions to validate this. It also includes planning activities in software development life cycle, beginning from project planning for the developed projects toward producing software project plan document.

The questions measured the usage of these activities, and the results varied among the participants. We investigate some issues related to project planning, 33.3% of them prepare project plan report for the developed projects, and 15.8% prepare monitoring reports, 28.9 % of them use time planning that allows tracking projects development, personal tasks, and other activities, 16.7% of them make enough risk management for their projects. Table (5) shows the distribution of responses to this question in the sample.

	Rarely (<25%)	Sometimes (25-49%)	Often (50-74%)	Almost always (75%+)	Don't know / N/A

Preparing Project plan reports	**22.2%**	**16.7%**	**27.8%**	**33.3%**	**0.0%**
Preparing monitoring reports to evaluate the progress.	**47.4%**	**10.5%**	**21.1%**	**15.8%**	**5.3%**
Using time planning and to-do list software.	**22.2%**	**16.7%**	**22.2%**	**38.9%**	**0.0%**
Needs to change Time plane according to late in project development.	**33.3%**	**22.2%**	**27.8%**	**16.7%**	**0.0%**
Making enough risk management.	**33.3%**	**22.2%**	**22.2%**	**16.7%**	**5.6%**
The challenges faced by someone building a three-month application are quite different than the challenges faced by someone building a one-year application.	**21.1%**	**26.3%**	**15.8%**	**36.8%**	**0.0%**
Omitting necessary tasks from estimates can add 20 to 30 percent to a development schedule.	**26.3%**	**21.1%**	**26.3%**	**21.1%**	**5.31%**

Table 5: Project Planning Activities

About using version control systems to share the resources of projects among the team member and manage the changes of the system, it found that most of them (46.7%) use Visual SourceSafe (SVN), while 33.3 % of them do not use any versioning system as shown in table (6).

Value	Percent %
Visual SourceSafe (SVN)	46.7%
CVs	13.3%
Subversion	13.3%
Other Open Source revision control system	6.7%
None	33.3%

Table 6: Types of used versioning system

It is obvious that Visual Source Safe is the most famous tool for version control among organizations who build software systems in Gaza Strip.

About using coding standards, it was found that 80% of developers have coding standards in their organizations and 20 % do not. Hence, we can conclude that coding makes that most important issue for people who work in the software development in Gaza Strip.

5.3 Peer Review

When we asked about using peer review to review and inspect the code by the developers, 66.7% of the respondents use peer review. That is another indication that participants concentrate on coding.

For those who did not apply peer reviews about the reason why they did not do so. We found that (30.3%) did not perform peer review because they are too busy in their work.

Value	Percent %
It is considered unnecessary as the code is usually of high quality	11.1%
Work is too busy to review code	33.3%
Reviewing code brings no benefit	11.1%
Unsure how to review to best effect	22.2%
Other	22.2%

Table7. Reasons to not perform peer review

When asking developers about when they perform peer review, as in figure 6, it can be seen that 55.6% of the respondents perform code review before product release and 55.6% of them use peer review continuously.

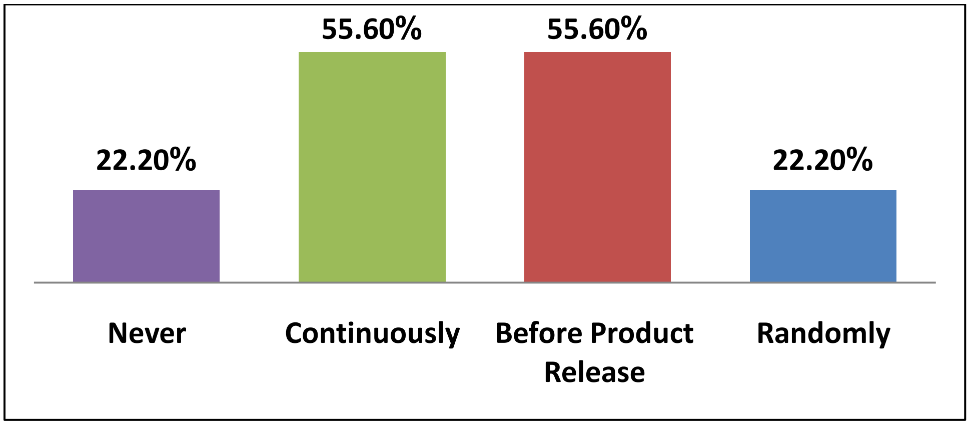

Figure 6: When perform peer review

When we asked about the average number of persons who perform peer review, almost 22.2% of all respondents said that they never reviewed someone else's code, 33.3% of the respondents ask a second party to review their code frequently, and 22.2% rarely ask for assistance from a second party to review the code.

About the average number of people who review the other developers' code, 44.4% of them indicated that (two–four) individuals do the review, 33.3% said that one person do the review, and 11.1% of them said more than five persons do the review. See figure 7.

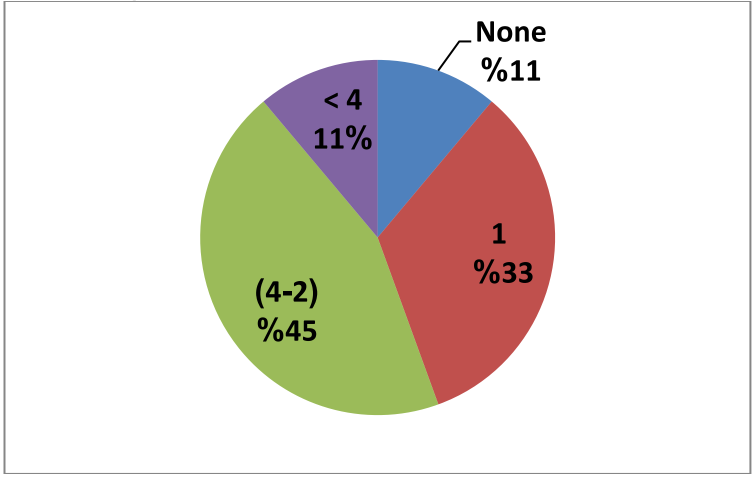

Figure 7: Average number of persons do peer review

5.4 Testing

Testing is an important part of the development process for all projects. It is important to ensure that the quality of software is high and satisfies the customer's needs. However, according to the survey results, we found that testing activities is neglected in our organizations in Gaza.

One of these questions was about having testers in IT units and companies responsible for testing process in Gaza. It was found that 73.3% of them do not have testers in their institutes. From this, it is clear that institutes do not pay much attention to testing phase. And this is the reason for the high number of software failure.

However, it was found by 100% that all the developers test their code by themselves. This is obvious because any developer needs to test his code before the releasing phase.

About making test cases plans in testing process, the result was that 53.3 % of developers make test cases plans. When asked about kinds of tests that they perform, developers indicated that most of them are involved in unit testing and database testing (figure 8). This supports the last conclusion about doing tests by developer rather than testers since unit test and database test are normally done by developers.

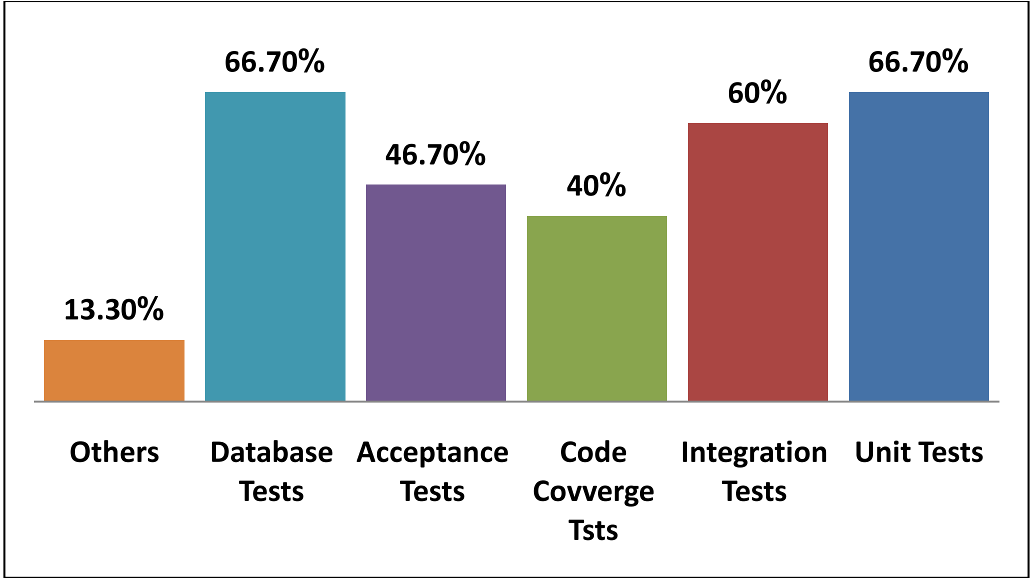

Figure 8: Used testing types

It was found also that most of developers (55.6 %) in Gaza do not receive any formal education/training in software testing. So, we recommend that developers must have training in testing to reduce number of software failures.

As shown in table (8), most participants (77.8%) performance continuous testing for their projects and 44.4% of them test the products before releasing them in the development site, and 22.2 % of them make testing at the client side.

Value	Percent %
Continuously	77.8%
Before product release (alpha testing)	44.4%
After product is released to specific users	22.2%
Randomly	0%

Table 8. When do testing

When asking about the used strategies in choosing the test cases, it was found that most of developers use inputs that simulate the user's behavior 77.8%, and 55.6% of them choose input that may cause failure, as indicated in table (9).

Test Strategy	Percentage
Provide inputs trying to simulate valid user behaviour	77.8 %
Choose those inputs most likely to cause failure	55.6 %
Choose inputs according to experience	55.6 %
Use script to provide random values as inputs	11.1 %
Provide Extreme values as inputs	33.3 %
Provide Boundary conditions as inputs	22.2 %
Try extreme loads on the system	44.4 %
Others	11.1 %

Table (9): Strategies used in choosing test cases

About how much time they spend in testing as in figure (9), it is noticed that 55.6% of the respondents spent less than 20% of their time on testing.

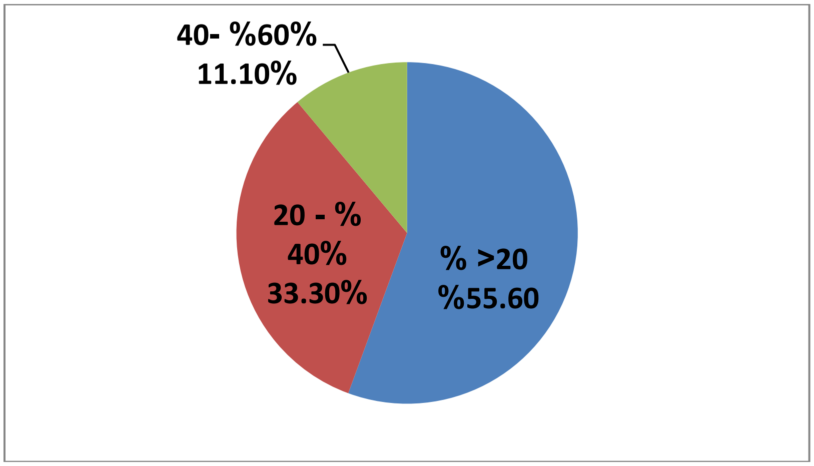

Figure 9: Spent time in testing

In addition, as shown in table 10, it is found that the percentage of covered source code by the testing activity was 12.5%, of who cover more that 80 % of the code, 37.5 % cover 40% – 60% of the code.

Value	Percent %
40% - 60%	37.5%
60% - 80%	37.5%
>80%	12.5%
Don't know	12.5%

Table 10: Covered code by testing

When asking about the percentage of discovered defects during testing, we found that most of developers (55.6%) discover 20 - 40% of the defects during testing as shown in table (11). This is normal because developers use testing only during development.

Value	Percent %
<20%	22.2%
20% - 40%	55.6%
40% - 60%	22.2%
60% - 80%	0%
>80%	0%

Table 11: Discovered defects during testing

But when we asked about defects discovered by the user, 44.4% of developers indicated that less than 20% of the defects discovered by the user, since it affects the quality of the delivered software and affects the success of the development organization.

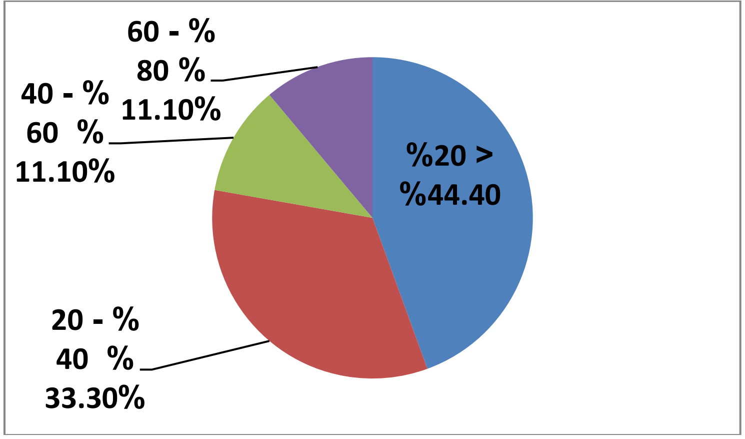

Figure 10: Discovered defects by the user

About using automated testing tools, most of developers (77.3 %) do not use automated testing tools. The reason for this is that most of them do not have training for software testing.

5.5 Maintenance:

One of the important factors that affect the quality of software was the repeated modifications after the software release, it is found that 44.4% of developers face this problem sometimes, and 11.1% of them indicated it happens always as in figure 11.

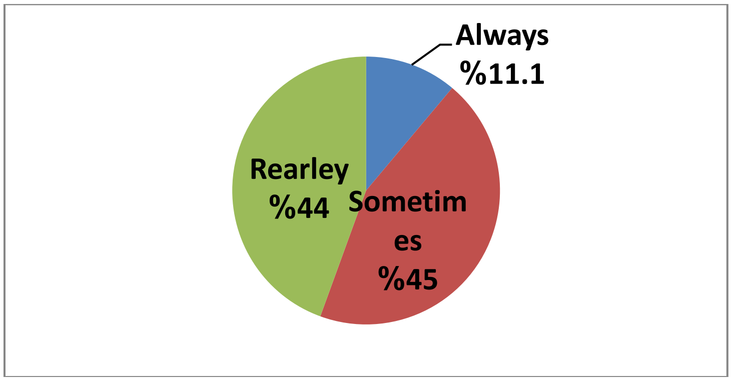

Figure 11: Repeated modifications after software releasing

This is considered as main deficiency in software development, since maintenance is sometimes more important than development. So, we recommend that software developers to take care more in maintenance to produce more useful software.

6.0 Conclusion and Recommendations

From this survey, some conclusions can be made about performing software quality assurance activities and applying software engineering practices our institutes in Gaza Strip. We can sum up that SQA practices are neglected by most of the local organizations in Gaza, in other words it is seldom to find an organization that applies all software engineering practices and methods; which leads to the production of low quality software. Further, there is no interest in testing process by many local companies and organizations in Gaza. Consequently, many problems occur when these products are released to the customer. These factors affect the quality, strengths and maintenance of the software. It can be see that running applications need time to reach a consistent state according to weakens in its structure or insufficient usage of software engineering practices.

To improve the quality of locally produced software in Gaza, it is recommended to invest more time in testing process that minimize the cost of maintenance after releasing the software; to improve the quality of the software; to provide training for the staff in testing methodologies; to have defined and confirmed requirements by the user, since most problems that face the user in Gaza results from the poor requirements that lead to low quality software; and to monitor the quality assurance activities through the software development life cycle.

References

[1] Javed,A ; Maqsood, M. ; Qazi,K. ; Shah K. ,"How To Improve Software Quality Assurance In Developing Countries", Advanced Computing: An International Journal (ACIJ), Vol.3, No.2, March (2012).

[2] Kandt, R. ”Software Engineering Quality Practices”, Auerbach Publications 2005.

[3] Alsultanny,Y. and Wohaishi, A. "Requirements of Software Quality Assurance Model" . Second International Conference on Environmental and Computer Science , Dubai, UAE December 28-December 30, 2009.

[4] Parnas ,D. "The Role of Inspection in Software Quality Assurance", IEEE Trans. Software Eng. 29(8): 674-676 (2003).

[5] Schulmeyer, G. Handbook of Software Quality Assurance Fourth Edition. Artech House 2008.

[6] Iqbal, n. and Qureshi, R. "Improvement of Key Problems of Software Testing in Quality Assurance" SCI.INT.(LAHORE), 21(1),25-28,2009.

[7] Iftikhar A. and Ali M. "Software Quality Assurance A Study Based on Pakistan’s Software Industry" Pak. j. eng. technol. sci. Volume 1, No 2, 2011.

[8] Lee, J. and Lee, K. "Survey on software testing practices" IET Software,2011.

[9] Yasuda .K. and T Yamada ,S. The Concept and Practice of Software Quality Assurance in Japan in the Global Era". Proceedings of the 7th IEEE International Symposium on High Assurance Systems Engineering 2002.