
QUALITY ASSURANCE IN EXTREME PROGRAMMING

Plamen Balkanski

Abstract: Our previous research about possible quality improvements in Extreme Programming (XP) led us to a conclusion that XP supports many good engineering practices but there is still place for refinements. Our proposal was to add dedicated Quality Assurance (QA) measures, which should be sufficiently effective and at the same time simpler enough in the context of XP. This paper intends to analyze the possibilities for an effective way for applying approved quality assurance practices to XP. The last should not affect negatively to the process and in the meantime must lead to better quality assurance. We aim to make changes to XP that even if would slow down a bit the development process, will make it more suitable for widest range of projects including large and very large projects as well as life critical and highly reliable systems.

INTRODUCTION

By our means XP suggests some very good practices proposed almost by every proven software development method but at the same time skips most of the documentation, rejects code reusability and relies on some circumstances which can be considered as not always completely realizable. We think that XP might be extended in a way that it will become more suitable for large and very large projects and teams and at the same time we will be able to keep it extreme. The first step in this direction will be adding the QA. In this paper we will try to propose a convenient way to implement well-known and effective QA practices in the XP process. XP is considered as a defined and disciplined process. Having in mind this we will make comparisons and look for relations using ISO standard 9000-3 which defines Quality Management System relevant to software development and Capability Maturity Model for Software (CMM), which defines at its level 2 the QA practices. The goal of this research will be achieved by examining both sides' practices, comparing activities and analyzing possibilities for implementation of the QA proposed by the above mentioned documents in the XP process.

SOFTWARE QUALITY ASSURANCE PROPOSED BY CMM

As is well known the CMM describes an evolutionary improvement path from an immature process to a mature disciplined process. CMM defines key practices to improve the ability of the organization to meet goals for cost, functionality and quality.

The QA activities are defined at level 2. According to CMM the purpose of Software Quality Assurance (SQA) is to provide the management with appropriate visibility into the process being used by the software project and of the products being built. It is required that the project follows a written organizational policy for implementing the SQA. The project should have assigned a SQA group that is responsible for coordinating and implementing SQA for the project. This groups should be provided with adequate resources and funding and should include members that are trained to perform their SQA activities.

CMM defines eight activities to be performed as follows:

- A SQA plan is prepared for the software project according to a documented procedure. This plan is developed among with the overall project planning and is reviewed by the affected groups and individuals (managers at different levels, client representative and any other involved) The SQA plan should be managed and controlled.
- The SQA Group's activities are performed in accordance with the SQA plan which includes:
 - o Responsibilities and authority of the SQA group
 - o Resource requirements for the SQA group
 - o Schedule and funding of the project's SQA activities
 - o Participation in establishing the software development plan (SDP), standards and procedures for the project
 - o Evaluations to be performed
 - o Audits and reviews to be conducted
 - o Project standards and procedures to be used as the basis for the SQA group's reviews

- Procedures for documenting and tracking noncompliance issues
- Documentation to produce
- Method and frequency to providing feedback to other related groups
- The SQA group participates in the preparation and review of the project's software development plan, standards and procedures. The SQA group provides consultation and reviews on: compliance to organizational policy and to external standards and requirements, standards that are appropriate for use by the project, topics that should be addressed in the SDP and other areas assigned by the project. The SQA group verifies that the plans standards and procedures can be used to review and audit the software project
- The SQA group reviews the software engineering activities to verify compliance
- The SQA group audits designated software work products to verify compliance
 - Software products are evaluated before they are delivered to the customer
 - Software is evaluated against the designated software standards and procedures
 - Deviations are identified and tracked and Corrections are verified
- The SQA group periodically reports the result of its activities to the software engineering group
- Deviations identified in the software activities and software work products are documented and handled according to a documented procedure.
 - Deviations are documented and resolved if possible
 - Not resolvable items are periodically reviewed by senior management until they are resolved
- The SQA group conducts periodic reviews of its activities and findings with customer's SQA personnel as appropriate

At the same time when performing these activities, measurement and analysis are made to be used to find out the cost and schedule status of the SQA activities. These measures include:

- Completion of milestones for the SQA activities compared to the plan
- Work completed, effort expended and funds expended compared to the plan
- Numbers of products audits and activity reviews compared to the plan

CMM also proposes verification of the SQA activities made by 3 different instances.

- The SQA activities are reviewed by the senior management on a periodic basis to provide awareness of and insight into software process activities at an appropriate level of abstraction
- The SQA activities are reviewed with the project manager on both a periodic and event driven basis
- Independent experts periodically review SQA activities and software work products of the SQA group.

QUALITY ASSURANCE PROPOSED BY ISO 9000-3

ISO 9000-3 is the standard of the ISO 9000 series that is most relevant to software development and maintenance. Organizations typically use ISO 9000 standards to regulate their internal quality systems and assure the quality systems of their suppliers. ISO proposes a quality assurance manual that consists of management responsibilities, a set of measurements, analysis and improvement activities, and required documentation.

An ISO 9000 organization should have implemented a Quality Management System (QMS) that is continuously maintained for effectiveness and process improvement. The effectiveness of the quality management system should be improved by the use of Quality Policy, quality objectives, audit results, analysis of data, corrective and preventive actions and management reviews.

The organization defines and documents its Policy, which provides the overall objectives for an effective QMS. The Quality Policy should be relevant to the organization's goals and expectations of its customers.

ISO 9000 requires an organization to plan and perform audits. The results of audits are communicated to management and any deficiencies found are corrected.

ISO 9000 states that organizations must establish adequate statistical techniques and use them to verify the acceptability of process capability – this is also called measurement. According to ISO 9000-3 “there are currently no universally accepted measures of software quality”. The auditors can accept the use of statistical tools or any consistently collected and used data.

The organizations should implement and maintain documented procedure to initiate corrective and preventive actions. Corrective action procedures define the requirements for:

- Reviewing non conformities including customer complaints
- Determining causes of non-conformities
- Evaluating the need for action to ensure that non-conformities do not recur
- Determining and implementing the action needed
- Records of the results of action implemented
- Review of corrective action implemented

The QA manager is responsible for Corrective and Preventive actions and a feedback system should be used to provide early warnings of quality problems. Preventive action procedures define requirements for:

- Determining potential non-conformities and their causes
- Evaluating the need for action to prevent occurrence of non-conformities
- Determining and implementing the action needed
- Records of the results of action implemented
- Reviewing preventive action implemented

The QMS documentation structure can be described at five levels. “Level 1” is maintained in the form of Quality Policy. “Level 2” documentation is maintained in the form of the Quality Assurance Manual. “Level 3” consists of quality procedure; “Level 4” contains work instructions. “Level 5” documentation is maintained as records/reports.

QA IN XP AND WAYS OF IMPROVEMENT

Adding QA seems to be the easiest part on the way to develop XP version suitable for large teams and projects. If we study the XP practices carefully we will notice that there already is some form of QA. It just need to be structured and controlled as well as somehow documented in order to use its results for analyzes. Bellow we will try to find out which of the above practices, measures and recommendations are suitable for the XP process, what roles are needed to conduct the new actions and how the collected information can be used. All the changes must be made in a way that would not slow down the development but still reaching the desired QA process.

We will go through the QA practices proposed by CMM and ISO and then will review the situation with XP and will propose our solution where appropriate for actions to be taken, roles to be added or other changes.

a) QA

- CMM defines the SQA aim as: providing the management with appropriate visibility into the process.
- ISO proposes a quality assurance manual that consists of management responsibilities, measurements, analysis and improvement activities, and required documentation.

XP:

- In XP the QA is not even mentioned but a lot of QA practices are presented. These are:
 - o The **automated acceptance tests** to prove that a feature is implemented correctly. We can add here the *test-first development XP practice, which also relies on the acceptance tests.*
 - o **Pair programming** which assures that the entire source code is reviewed all the time.
 - o **Refactoring** practice, which means removing duplication, increasing code integration and lowering mixture of the code.
 - o **Collective ownership**, which proposes code reviewing a common coding standard.

We see that XP uses QA practices but the difference is that these practices are only oriented to achieve the direct goal. For example Automated Tests intent to assure that the current version is good enough for a release, but does not keep any data for the results and does not makes any decisions. The same is for Pair Programming – both programmers care for the same code and make fewer errors but the errors are not recorded.

b) QA group

- CMM: It is required that the project follows a written organizational policy for implementing SQA. The project should have assigned a SQA group that is responsible for coordinating and implementing SQA for the project
- ISO: The organization should have implemented a QMS that is continuously maintained for effectiveness and process improvement. The effectiveness of the quality management system should be improved by the use of Quality Policy, quality objectives, audit results, analysis of data, corrective and preventive actions and management reviews.

XP:

- According to XP every project is different so it needs a different approach. However it is obvious that every team, which is working on more than one project gradually, builds its own software development process. So the next time everyone in the team understands better his responsibilities. The QA is in many ways independent from every single project system, which can be easily achieved in an XP team by adding another role – the QA Manager. As all other roles in the team individual that already has a role but not a programmer also can handle the QA role. The QA Manager will be involved in creating the plans and gathering any analytical information as well as taking decisions about any changes provoked by the QA results.

c) Activities

- CMM defines eight activities as listed above.
- ISO suggests that the organizations implement and maintain documented procedure to initiate corrective and preventive actions also described above.

XP:

Which of the proposed actions already exist in XP and which are a must? Both standards propose a documented procedure (ISO) or an SQA plan (CMM) that is missing in XP. We propose that the QA Manager must develop a documented procedure for the QA of an XP team. The QA process is very similar in most of the projects. Thus the QA manager can use a pattern, which can be tuned as appropriate for every project and of course developed through the process. Using such patterns or a generalized QA plan for the XP projects we think will save time or at least will not lead to serious delays.

In the context of XP we guess that it is better to use only one person for the QA instead of a group. This is needed for a number of reasons including keeping a small team and funding reasons. The QA Manager will take actions defined here and some automated registration programs (ARP) will help him with gathering the information. At the meantime as we cannot reject Code reviews then a lot of QA will be done by programmers but here come the ARPs which will present the QA manager with relevant information which can be used directly. The participation of the QA manager in establishing any plan or procedure or standard is automatically guaranteed by the fact that the team works together. Using common standards is related with the practice Collective Ownership that requires such common standards in every XP team.

The QA Manager should define a QA schedule. However we guess that this should be in relation to the versions issued by the XP team. This is because on every version there is a lot of testing and in this moment the integration can be also assessed.

Evaluations, tracking and reporting of the non-conformities are related to the used measures and will be discussed bellow. What we must note here is that the QA Manager should take part in defining the automated tests, as this is a good approach to evaluate clients as well as project internal quality measures.

d) Measures

- CMM proposes using measures which include:
 - o Completion of milestones for the SQA activities compared to the plan
 - o Work completed, effort expended and funds expended compared to the plan
 - o Numbers of products audits and activity reviews compared to the plan
- ISO states that organizations must establish adequate statistical techniques and use them to verify the acceptability of process capability. These can be the use of statistical tools or any consistently collected and used data.

XP:

- The first thing to say when writing about XP and measures is that in the initial process there are no measures. There are no analytical data as the only aim is to complete each project quickly and then to start over again with the next.

However we will need measures, as we would like to use the results for management decisions. As we have already mentioned above we propose using a number of ARPs which will record most of the measures like code compilation failures, syntax or functional changes, acceptance tests results or even programmers effective time. Such automatic registration can also be easily applied to some popular metrics as Lines of Code or Functional Points. All other tests as logical program behavior or program usability must be planned and assigned for failures registration by the QA Manager. Such failure registrations we propose must be added to the same database used by the ARPs.

Reporting of failures must be delivered to the QA Manager who can re-assign the problems to the team or send the lists to the Team Coach for re-assigning. Reporting of measures which do not directly affect the program functioning are delivered to the QA Manager, which can discuss them with other team members with management responsibilities and propose changes regarding the process.

CONCLUSION

By using the experience of ISO 9000-3 standard and CMM for Software regarding QA we made an analyze and conclude that adding QA to XP will be harmless for the process and will lead to the desired results – measuring which will provide the managers with relevant information and thus they will be able to make changes to improve their process. The main changes proposed by us are adding a new role – the QA manager that is responsible for the QA plan development and conducting and the use of ARP – small agents to record a number of measures which will be additionally proposed.

As was stated in the paper the QA manager on its own or by assigning tasks to the developers and using the ARPs fulfills nearly completely the requirements proposed by both ISO 9000-3 and CMM. At the same time it is clear that we will need a list of measures, which will be appropriate and can give the exact information that will be most useful and proper for an XP team. This is the first important step on the way developing an XP version, which can be used as a process for development of large and very large projects.

Literature

1. A.Eskenazi, P.Balkanski - Quality Improvement in XP Development
2. Beck K., Jeffries R., Anderson A., Hendrickson C., Jeffries Ronald E. Extreme Programming Installed
3. SW-CMM, Software Quality Assurance
4. ISO 9000-3, Quality management and quality assurance standards
5. Software metrics, SEI Curriculum Module SEI-CM-12-1.1

Author Information

Plamen Balkanski - I.T. Consultant in Global Industries UNISYS; Email - Plamen.Balkanski@unisys.com