Making Process and Product Quality Assurance (PPQA) Work on Small Projects

5TH ANNUAL CMMI TECHNOLOGY CONFERENCE AND USER GROUP

NOVEMBER 15, 2005

ELECTRONIC SYSTEMS LABORATORY
Georgia Tech Research Institute
Georgia Institute of Technology

Authors: Jean Swank, Jeanne Balsam, Lee Sheiner, and Mark Pellegrini

jeanne.balsam@gtri.gatech.edu

jean.swank@gtri.gatech.edu

lee.sheiner@gtri.gatech.edu

mark.pellegrini@gtri.gatech.edu

This page intentionally left blank.

Making Process and Product Quality Assurance (PPQA) Work on Small Projects

AUTHORS: JEAN SWANK, JEANNE BALSAM, LEE SHEINER, AND MARK PELLEGRINI

INTRODUCTION

Georgia Tech Research Institute (GTRI) is the nonprofit applied research arm of the Georgia Institute of Technology in Atlanta, Georgia. The Electronic Systems Laboratory (ELSYS) of the GTRI achieved a CMM Level 3 rating in June of 2003. ELSYS employs approximately 150 engineers and scientists working predominately on DoD related competitively bid contracts. Over the last 30 years, ELSYS researchers have established national reputations in areas such as: monopulse countermeasures, advanced radar warning receiver design, survivability, simulation models and analysis, and Electronic Counter Measures (ECM) technique development. GTRI/ELSYS core competencies include software and systems engineering for electronic warfare and avionics systems, reliability and maintainability upgrades, technology insertion, obsolescence programs, threat analysis, and mission critical software. Many of these projects are staffed by fewer than ten developers; some projects have only one or two.

ELSYS has transitioned the CMM Software Quality Assurance process to the CMMI Process and Product Quality Assurance (PPQA). This presentation will share lessons learned while transitioning to a fully compliant systems engineering PPQA function. The reasons for the following statements will be discussed:

1) Develop a generic QA plan and schedule that can be easily tailored for specific project/product needs.
2) Hire and/or recruit Quality Engineers that have enough experience that management respects their recommendations. These people can supplement technical and managerial expertise of the project team which visibly adds value to the development effort.
3) Have Quality Engineers act as mentors to the project team.
4) Analyze project and product risks to determine the most cost effective PPQA strategy.
5) Leverage project team process maturity for reduction of PPQA tasks and for process improvement. Praise process innovators and reward them to the extent that you are capable.
6) Concentrate PPQA on projects or product development efforts that have high risks; for example safety or technology.

This paper describes a method to implement effective PPQA in small organizations or small projects in order to produce best in class products with limited resources.

SMALL ORGANIZATIONS AND PROJECTS

In the context of this paper, “small organizations” refers to organizations of 150 or fewer people with projects ranging in size from one to twenty-five people. The resources a small organization is able to commit to PPQA are generally severely limited. In these organizations, a Quality Engineer typically cannot be assigned full-time to a single project.

A small organization may lack development phase specialization in its members. The same people who write the requirements may also be the testers. The designers might be the implementers. In very small organizations it may be the same people performing *all* of the development activities from start to finish.

This raises an interesting dilemma. What is more important, minimal quality assurance distributed equally on all projects or more comprehensive quality assurance on some, with others receiving very little? When different teams of people write requirements, design, implement, and test a product, there is a sort of "built-in" protection system. If the group writing the requirements does a poor job, the designers will, hopefully, complain that they have been given requirements that are too vague, or the testers will complain that they are un-testable. Likewise, if the coders are given a poor design, they may alert management that there is a problem. However, when the same people are doing all of the development from start to finish, they can walk down a primrose path, not realizing they have a disaster in the making. It is these projects that need PPQA the most, yet they can afford it the least. This is the challenge of making PPQA work on small projects.

Large organizations may have projects with multiple Quality Engineers assigned full-time to them, but they can still have some projects that are quite small, with limited resources and role-sharing. A large organization may experience minimal impact from the failure on one of its smaller projects, whereas a small organization may well experience severe consequences.

PROCESS AND PRODUCT QUALITY ASSURANCE

According to the Software Engineering Institute, "The purpose of Process and Product Quality Assurance is to provide staff and management with objective insight into processes and associated work products.[1]" For small projects there is one key word in this phrase unique to those projects: objective. Generally, everyone on a very small project has fairly good insight to what is happening on the project; what's missing is an objective set of eyes. On large projects there is inherently some objective oversight from other team members. Regardless of whether the project is large or small, management external to the project should be kept objectively informed of the technical and process status of the project.

PLANNING

Through years of experience, GTRI has determined that having a generic Quality Assurance (QA) Plan for the organization is the most effective both in cost and functionality. Additionally, a generic schedule that includes all tasks required by the organization's standard process is used as the starting point for each project. A database is used to track these schedules and any other supplemental material that is project or product specific. Over time the organization should develop a library of generic schedules for each product development type. The generic QA Plan serves approximately seventy-percent of the projects without revision. The QA Plan may be supplemented to address specific tailored processes, risks, and mitigation strategies.

[1] Mary Beth Chrissis, Mike Konrad, and Sandy Shrum, *CMMI* (Addison Wesley, 2003), 429.

The guidelines shown in Table 1 are provided as an example for developing a generic QA Plan for the organization. In general the QA plan needs to be consistent with plans developed for other purposes. It should include the introductory sections including: Identification, Scope, Document Overview, Referenced Documents, and Organizational Structure. Additionally, it should have standard tasks that parallel those of the organization's standard process.

Table 1

Section Name	Description
➢ Tasks	This section provides an outline with detailed expectations for each type of audit and review that QA will provide oversight and support.
o Perform Start-Up Tasks	These tasks are normally executed a single time, but may be repeated for major contractual changes or for incremental developments where this level of coordination/re-planning is necessary.
▪ Attend Project Initiation Meeting	
▪ Generate QA Plans and Schedules	
▪ Prepare for and Attend the Plan Review Meeting	
▪ Prepare Orientation for and Attend Project Kick-off Meeting	
▪ Attend Customer Meetings	
o Conduct Periodic Reviews of QA Activities	Explicitly define what reviews of QA activities are required. Define what type of data is shared at each level and the minimum frequency of communication.
▪ QA Manager	
▪ Lab Director	
▪ Senior Management	
▪ Project Director	
▪ Project Team	
o Mentor Project Team in Organizational Process Activities	Define general mentoring activities that Quality Engineers will conduct during the life of the project, including the value of those activities.
o Support Customer Quality Management System	Define minimum types of support that will be provided by the Quality Engineer to the customer.
o Resolving Disputes	Define methods for resolving disputes between the Quality Engineer and the project team.
➢ Documentation	Define where additional documentation associated with QA activities will be stored (may be by reference).
➢ Standards, Practices, and Conventions	Define where the official organizational standards, practices, policies, guidelines, and conventions are located.
o Tailoring of Standard Process	Reference tailoring practices for the organization's standard process.
o Monitoring Compliance	Define how compliance will be monitored and how deviations will be processed.
➢ Reviews and Audits	Define who shall conduct reviews and audits and how the respective processes and standards will be used in these reviews.
o Technical Reviews	Reference applicable procedures and standards for conduct of technical reviews. Include the level of detail necessary for each type of review and audit to be conducted. Define how data collected during the reviews will be analyzed and reported.
▪ Conduct Periodic Reviews of Project Activities	
• Configuration Management	
• Software Product Engineering Process	
• Hardware Product Engineering Process	
▪ Peer Reviews	
▪ Technical Audits	
• Collect Measurement Data and Document Deviations	
• Analyze Data and Report Results	
▪ Managerial Reviews	
o Configuration Management	Detail specific processes for configuration management audits.
o Problem reporting and corrective actions	Define how problems are reported and corrective actions are tracked to closure.
o QA Document Identification Conventions	Define document identification conventions for QA artifacts.

QUALITY ENGINEERS

To add the greatest value it is essential that quality engineers be well-qualified in both managerial and technical areas. In GTRI, Quality Engineers are required to have technical degrees in either computer science or engineering, with practical experience in product development including project management. In additional to being familiar with the organization's defined engineering processes, they also need to understand project planning and risk management. They are capable of performing technical and managerial work, although they are assigned to the project as an organizationally independent monitor.

Having the ability to do the actual work makes a Quality Engineer far more valuable than a "box-checker." Certainly, checking-the-boxes along the development path is better than having no independent verification at all, but some problems do not lend themselves to discovery simply by seeing if the proper box has been checked. Technically trained Quality Engineers are more likely to detect trouble on a project (and more quickly) than ones who do not understand the technical aspects of the product whose development processes they are monitoring.

Highly qualified Quality Engineers bolster the capabilities of the project team because they add an independent set of eyes to the project team. They attend meetings, review project documentation, and are aware of what the team is doing. Not only can they help spot problems, they can provide suggestions and advice that are valued, as they are recognized by the team as being qualified to do so. Quality Engineers who participate in multiple projects provide another valuable service – they can share technical information between these projects in a way that someone who is simply checking-the-boxes never could. This can help avoid conflicts between products that share requirements or resources. Often the Quality Engineer can share solutions developed on one project with a different project team that is having similar problems.

MENTORING

Classroom training alone is seldom enough to provide a practice capability. In small organizations typically it is expected that employees will learn through self-study and informal mentoring from other project team members. The Quality Engineer is in a good position to identify developers who are in need of mentoring in order to develop a practice capability with the organization's standards and processes.

ASSESSING QUALITY RISKS

In order to properly apply scarce quality assurance resources to projects, it is first necessary to identify the highest areas of risk. A number of factors need to be considered.

Personnel – Knowledge of the capabilities and work habits of the people on a project team can be valuable in deciding where to allocate resources. If the team members are known to generally conform to the organization's defined processes – with everything else being equal – it would be more effective to allocate quality assurance resources to other project teams that are known to be less process compliant or technically challenged. Quite simply, it makes sense to spend time looking for process violations among people who have a history of violating the organization's processes. Another personnel factor is the level of technical experience of the team. Inexperienced developers would ordinarily warrant closer inspection than those who are veterans. If the Quality Engineers are well-trained and capable of doing technical work, as the authors contend they should be, they can periodically sample the work and sound a warning if there appears to be a problem. In any case, sufficient and appropriate peer reviews of an inexperienced developers' work should be conducted; proper quality assurance verifies that these reviews are being scheduled and completed.

Development Phases – Ideally, a defined process would be followed throughout the entire lifecycle of every product, including a continuous verification by quality assurance that the product is being built correctly. However, in the absence of enough resources to verify continuously, there are certain key phases of development where the quality of the output needs to be verified; for example, the requirements, design, development, and testing phases. Unquestionably it would be better to have requirements written correctly from the start, but if there is a problem with them, it is better to detect and correct the problem before they are used as the foundation for design rather than afterwards. Likewise, a poor design should be identified and corrected before it is implemented. If the quality assurance budget only permits limited involvement of a Quality Engineer in the product's development, it is better to schedule the time at the critical phases, rather than concentrating in a single phase. A set of rock-solid requirements is a good start, but if the entire quality assurance budget was spent on their development and the project goes astray during design, this is not a good trade-off.

Cost of Failure – Sometimes if a product fails, the cost of failure can exceed the money spent on developing it. For example, it could be a key component of some other much larger product or system whose success is dependent upon the smaller one. Loss of reputation or team morale is also an important consideration. But sometimes a small product is just a small product, and if it fails it doesn't have dire consequences for the organization. If two projects have an equal chance of having problems, but one has far greater consequences to the organization if it fails, it makes sense to put more resources on the one that is more important.

Familiarity with the Subject Area – If the product being developed uses new technologies, is planned for deployment in unfamiliar environments or has problems that the organization has never faced before, it is probably a good candidate for more quality assurance resources than one without these challenges. If it is a new product that is very similar in function and scope to an earlier product, it will pose less risk than an unfamiliar one. However, the experience of the project team needs to be considered. Even if the product is very similar to other ones that the organization has created, if the project team has no direct experience with the similar products the risk may still be high.

ADDRESSING PROJECT RISKS

The job of the quality engineer is to make sure that project risks are assessed and documented. Additionally, quality risks should be factored in when assigning resources. In ELSYS, the project is responsible for funding quality assurance. If the project lacks sufficient funding, then corporate resources may be used to insure sufficient oversight is given to the project. After quality and technical risks have been assessed, the risk mitigation plan should be implemented and monitored.

INSTITUTIONALIZING PROCESSES

In a perfect world there would be no need for quality assurance activities; everyone would be qualified to do their job, they would do it perfectly every time, and everyone would follow the organization's procedures for developing products. The world, unfortunately, isn't perfect. Neither is it completely imperfect, where every developer needs a full-time Quality Engineer sitting next to them watching everything they do. The reality is somewhere in between.

The most effective way to utilize good processes to create outstanding products is to create an environment where the project teams *want* to follow these processes, rather than do it because they are forced to do so. Thus, process improvement for product development is more effective when it comes from the bottom up, rather than from the top down. The people doing the work suffer the consequences of their own mistakes, and they can identify the ones that could have been avoided through better processes. The most motivated of these people will take it upon themselves to tailor or to extend the organization's processes to meet their needs. The Quality Engineer is management's representative "in the trenches," and can identify process improvements that should be more generally distributed. Some of these improvements will be generally applicable within the organization and should be incorporated as changes to the defined processes.

The organization needs to identify "star players" who utilize existing processes and work to improve those processes, and those individuals who may not necessarily improve processes, but comply with them. These people should be praised, rewarded, and encouraged to continue doing so. They become role models for the other developers, encouraging them to be compliant and innovative as well. When the project team is voluntarily and enthusiastically following the organization's processes institutionalization occurs and the need for quality assurance is diminished, reducing the need for those scare resources.

PRACTICAL CONSIDERATIONS

Responsibility for project outcome rests on the project manager's shoulders and those of senior management. As such, quality engineers are responsible for bringing concerns to the attention of the project managers and if necessary, senior management. The quality engineer acts as the conscience of the organization, not the police. Senior management must support the quality engineer with a concrete stance on processes and policies. In order for senior management to give that support they must respect the decisions of the quality engineers. Tailoring of processes should be allowed when it makes sense. Variance should be approved when necessary. Quality of the product should always be the guiding value, not who's in charge. Thus, quality engineers should mentor project team members and listen to their concerns to ensure that the best quality processes are utilized and best quality products are built. Make sure that there is a two-way street for communications.

SUMMARY

Process and Product Quality Assurance is the means by which project team members, as well as management, get insight into the processes used and work products produced during the duration of a product's development. Small projects by their nature have proportionally more need for PPQA, yet they are often the ones that can least afford it. Clearly, it is important for the resources that are dedicated towards small projects to be used as effectively as possible because the margin for error is lower than for larger projects.

If the personnel who are performing PPQA are technically qualified to do the type of work they are monitoring, they can be far more effective in performing their duties than those who are not. They are more likely to detect problems in the product, and identify them earlier, than someone who does not understand the work they are monitoring.

Planning PPQA activities can take proportionally more time for a smaller project than a larger one, so it is desirable to streamline the planning process as much as possible. The use of generic plans and schedule templates can help reduce the time needed to plan PPQA activities.

When deciding how to allocate scarce PPQA resources to projects, the risks must be evaluated to decide which projects need more of those resources. The technical experience level of the project team members, their history of process compliance, and their familiarity with the specific type of product being developed must all be considered. Each project's potential cost to the organization if it fails must also be considered when determining the level of effort that should be allocated for PPQA.

Personnel who perform PPQA on multiple small projects are in a unique position to facilitate communication if most developers in the organization normally work on a single project at a time. They can help to rapidly spread important technical information between the project teams. They can also help to identify experts on one project team whose expertise might be extremely valuable in solving a problem another team is confronting.

Even in small doses, the presence of PPQA on a project reminds the team that there is a process that they need to follow as they develop their products, and there are certain standards that those products must meet. They become the "little voice" in the minds of the developers and act as the conscience of the team regarding process compliance.