

RCG INFORMATION TECHNOLOGY



delivery excellence[®]

Quality Gates in Solution Projects *White Paper*



RCG
INFORMATION
TECHNOLOGY

Q-Gates in Solutions Projects: All They're Made Out to Be

Ron Younack
RCG Information Technology

Anyone who has ever managed a project has a war story or two to tell. A project that went awry because the staff tried to do too much, too fast, or too cheap happens much too often. Unfortunately, it's the professional project manager who often tries to juggle the time/cost/functionality/quality paradigm on his own. Having been there, I know it gets very lonely.

For example, I recall one particular project on which everything seemed to take twice as long as was originally estimated. The variations were justifiable – the estimates assumed experienced resources but the team consisted of primarily “junior” level resources. Equipment ordered did not arrive on time. In addition, the business users were not available to meet with the business analysts as required, thus, causing delays in the project. Of course, while the workload was increasing, the one item that never changed was the end date - it was fixed, and the team was told that they had to find a way to get the system done within the original timeframes. Sure, more resources could be added, but remember the adage that “nine women cannot make a baby in one month”?

In what was a misdirected but valiant effort, the project manager took a chance and decided to conduct multiple phases of the project concurrently (parallel tasking) in an attempt to stay within the time constraints of the project. He instructed the team to start coding before detailed design was completed, assuming that the amount of retrofitting required to synchronize the two at a later date would be minimal. As you might imagine, the manager's worst nightmare resulted. The quality of the code was disastrous because of the lack of detailed design information available to the programmers, and retrofitting the code to the finalized design was extensive. In the end, the team missed the deadline. Everyone conveniently forgot any of the manager's earlier warnings, so it came to be the manager's fault. After all, the Project Manager is accountable for the success of the project. Sound familiar?

So what went wrong? Very simply, one person, the project manager, was allowed to unilaterally circumvent the software development methodology that was being followed. There was no “gate,” or formal completion criteria, that prevented the team from moving forward into coding without having completed detailed design. At the very least, a risk assessment should have been conducted for approval by the project stakeholders. If the risk was warranted and the decision to move forward was jointly made, accountability would have been shared by all involved. Instead, the Project Manager was allowed to make that decision on his own accord. Yes, he was well meaning, but *doing right* and *doing the right thing* are two very different things.

Most software development organizations have quality control processes in place. Many go as far as having an independent test group to ensure that the final product released is of sound quality and meets requirements, standards, etc. However, what is often missing are adequate *quality assurance* processes – processes to ensure predictable quality is delivered. By definition, successful quality assurance will result in reduced quality control failures. A very important distinction that is often understated is that quality control without quality assurance may find errors after the fact, rather than preventing errors before they occur.

One of the techniques employed within quality assurance is a quality gate (Q-gate) concept. A Q-gate process, when used in conjunction with project management, helps manage the balance between software cost, functionality, and quality. A Q-gate marks the formal end to a particular process within the software development lifecycle, a “gate” through which the project proceeds when moving from one

phase to another. Each Q-gate process results in the certification that all appropriate work products required to move forward to subsequent project activities have been completed and reviewed, and meet quality expectations. Using a set of pre-determined exit criteria established for each phase or milestone being certified, a Q-gate results in a pass/fail decision for moving forward. Time and cost issues, as well as risks, are identified during the process, and based upon the Q-gate results, key stakeholders associated with the project are provided with the necessary information to make business decisions to take action on any deviations. The benefits of the Q-gate concept are:

1. Project stakeholders share responsibility with the project manager for the project's quality outcome.
2. Q-gates provide a mechanism for the project team to readily assess the quality of their work products throughout the project life cycle and improve quality at the source.
3. Q-gates ensure that intermediate work products meet the needs of downstream activities through a formal, disciplined process. Problems, their resolution, and opportunities for improvement are identified; and risk assessment or escalation procedures are invoked, if appropriate.
4. Q-gates enhance project communications, thereby resulting in improved management of the expectations of key stakeholders through their participation in Q-gate certifications.

In practice, the Q-gate is a fairly simple process – it formalizes what already does or should have been happening all along and ensures communication to all stakeholders. It can be as simple as a checklist which results in a pass/fail score. For each major milestone or phase within an organization's methodology, a Q-gate checklist would be constructed containing a list of "exit criteria", tasks and/or deliverables which must be completed prior to moving forward to the next activity. Deliverables might be mandatory or optional, and individual Q-gate exit criteria might be optional based upon project attributes such as size, duration, or type of project being executed. As is the case with all processes and procedures implemented for software development, Q-gates must be flexible enough to allow customization on a project by project basis.

Recalling the project manager's decision to proceed with coding, even though design had not been completed that was described earlier, a formal Q-Gate process would not have allowed that to happen without proper communication, discussion and agreement. However, if the business risks in doing so were outweighed by the potential benefits, at least that decision would have been jointly made based upon a solid risk assessment instead of an emotional, unilateral decision. And the Project Manager described would have continued his employment.

About the Author

Ronald Younack is a Director in RCG IT's Implementation Quality Assurance (IQA) group. He has over 30 years of experience in the delivery of large-scale Application Development projects in the banking and financial services areas. He holds a BS degree in Information Systems Technology from Trinity College. He is also certified by the Project Management Institute as a Project Management Professional (PMP) and holds a Certified Computing Professional (CCP) designation issued by the Institute for the Certification of Computing Professions (ICCP).