

Rescuing and Revitalizing the Problem Project

Cinda Voegtli, President, ProjectConnections.com

Abstract

There's no worse "super-urgent demand" than the looming end-date of a critical project that's behind schedule—a schedule that was probably highly aggressive to start with. Time-to-market demands can't be ignored, so how do you course-correct, get back your momentum, and succeed in spite of the setbacks?

This paper, which I originally presented at the ProjectWorld conference, describes how to use the guidelines and techniques of Quality *Rapid* Product Development (QRPD®) to pinpoint and fully understand the project's past problems, revitalize your plan and your team, and get that product out the door. Typical scenarios from actual projects are used to illustrate how these approaches come into play in the real world. I also discuss how the same guidelines and techniques could have been used to avoid the problems in the first place.

Introduction

It's no fun when a project falls behind schedule. The demands on an already pressured team get worse, but the recovery plan may be unrealistic or non-existent. The natural tendency toward denial kicks in: "We'll catch up somehow" (method undefined). Often the sheer magnitude of the actual slip is not realized until the situation is grave.

Once the project's predicament is obvious, all too often the energy that's needed for finding a solution goes instead to recriminations, "spin control", and attempts to place blame elsewhere. The fact is that time-to-market still matters, so the company and the project team must go beyond those natural reactions to constructive action: a real plan for getting the product out the door. Unfortunately, since people may experience weighty feelings of failure and burnout from efforts to-date, the enthusiasm, energy, and momentum the team needs for a recovery can be extremely hard to come by.

So how do we get there from here? By accomplishing the following steps:

- **Step 1. Assess current status:**

Step 1a. What were the project's problems, at a nuts-and-bolts level? What other problems might still be lurking?

Step 1b. Based on those problems, where are we with respect to our technical design, our schedule, and the marketplace?

- **Step 2. Analyze alternatives:** What can we do now to best meet the company's goals?

- **Step 3. Decide new course:** Pick the best alternative and finalize the plan.

- **Step 4. Re-launch:** Get the team's energy up, everyone recommitted to a realistic plan, and get going!

These activities sound standard, but the way in which they are carried out has significant impact on whether the project can be recovered. The techniques of Quality *Rapid* Product Development (QRPD), a methodology I've used to manage projects for the last 10 years, can be put to good use in recovering the problem project. QRPD¹ is a set of management principles and practices geared to fast-paced development by high-performance teams in chaotic market environments. QRPD emphasizes the ultimate return on investment of the project, to put the "time-to-market" part of the equation in its proper perspective. The importance of individual team member contributions and responsibility is paramount. QRPD emphasizes honest, realistic, thorough assessment of a problem project: to have the best chance of revitalizing the plan and the team, you must look back at root causes of project problems, and look ahead for opportunities for synergy and ultimate overall success. This practical, personal, honest, realistic focus helps diffuse the negativity of the "impossible" project situation and get the project on the road to recovery.

Step 1. Assess the situation

The Importance of an Accurate Assessment

When a project is in trouble, the normal response is to simply start re-planning the rest of the project, possibly adding resources in an attempt to finish the same amount of work in less time. That method deals only with the symptoms (the lateness) and not necessarily the root causes of the lateness, and therefore may not actually be able to improve the outcome significantly. We must first accept the importance of understanding the real root causes of the project's problems.

Before deciding how to fix our project, we must also be honest about where we really are with respect to *meeting the corporate goals of this project*. We can't correctly decide our plan for recovering if we don't know *with accuracy* where we are, with respect to the end-date we originally planned, and with respect to being successful at that end-date with the right product, system, or service. The corporation's definition of success is "return on investment" from a project. For the dollars invested in product development (the development cost), what is the return in revenue? The other factors that contribute to a return are its "performance" (its feature set and attributes: does it provide what the customers want? is it manufacturable and serviceable?), its time to market (did we get to market fast enough?), and its cost (are customers willing to pay for it?). To achieve maximum return on investment on any project, the development team must choose the right balance of these factors: performance ("P"), time-to-market ("T"), product/service cost ("C"), and development cost ("C"). This balance is referred to in QRPD as the "PTCC" balance. We likewise need to pay attention to the PTCC balance in our decisions of how to recover from a floundering project.

Step 1a. Assessing the Project

The most thorough and revealing way to assess a project is to look for violations of the 10 QRPD management "commandments." QRPD recommends numerous day-to-day techniques for carrying out each of the commandments; project problems can often be traced to violations of those techniques. If we methodically examine our execution against the recommendations of QRPD, we can get a realistic assessment of what has gone wrong,

the magnitude of our underlying problems and their potential impact to the project, and opportunities for getting back on track.

The following sections provides questions for recognizing the cause of your project's problems based on the QRPD commandments, and the recommended actions for correcting those problems to help recover the project. QRPD provides numerous detailed techniques for carrying out the 10 commandments; this section highlights one or more typical technique violations for each commandment.

1. Focus on a clear and limited vision.

The number one reason for not getting done on time is that we tend to take on too much, of too risky a nature, at once, and lose sight of the most important set of *limited* objectives. To hold our time-to-market we have to carefully choose the features most important for maximizing return on investment, and maintain our focus on those goals. Which of these characterized your project?

"The requirements are still changing, changing..." "So what is this thing we're developing?" "Here's the feature list of the week." These all mean that the definition of the product is not stable and changes are not controlled.

If your requirements are unstable, it is imperative that you stop and create a Project Vision. The Project Vision is a short, high-level document, created by the team, that captures non-negotiable project requirements. It focuses on the benefits provided to the customer and an understanding of what the customer will use to judge quality, key technologies and mandatory features, crucial product factors, and financial goals and constraints. No matter where you are in the project schedule, you may not have the right definition, and changes may still happen to lengthen your schedule further. The fastest and most profitable way to finish the project may be to redefine the product! The Vision will bring team understanding and agreement upon what the project must create to meet the company's goals.

"This product isn't what we want." This suggests that the vision was incorrect, because customers weren't adequately involved in defining the product. Or the vision may have been defined correctly, but the team strayed from the vision during

development. When you defined this project, did you talk to customers? Did you ensure that everyone on the team participated early in defining the vision? Did you control changes to the Vision and other product specifications?

In writing the Project Vision, make sure you have real data from the market or your internal customers. If that hasn't happened to-date on your project, do it now. Interview both leading edge and target customers. Leading edge customers will help you understand where visionary users think the market is going. Target customers will give you the perspective of how to make your product or service succeed for the masses, for a sustained period of time. If the market soul-searching and high level of cross-functional inputs necessary to write this document has not been performed, you may have missed a critical aspect of the definition, and it is still worth doing, no matter how far along you are in the project. Then the team **MUST** keep the Vision accessible and review it periodically.

"This is a complex do-it-all product; we can't predict when we'll be done." This is symptomatic of a team that didn't pick its battles or identify which features and technologies were absolutely necessary in the light of balancing PTCC.

Were you trying to do too much technologically in one project? Analyze the technological risks. Have you undertaken more than three innovations - technological or other fundamental aspects which no one currently in-house has ever tried to implement? A *QRPD* rule of thumb is to limit these "major" innovations to 3. Because of the unknowns involved, the probability of disruptions to the schedule is high.

After the full assessment, this understanding of technical risks will be used to help decide whether the first release should contain everything you are attempting, to get the project done with the best balance of PTCC.

2. Assemble the right team and leader

The first *QRPD* commandment dealt with the appropriateness of the project definition and the team's focus on it; the second commandment deals with the appropriateness of the team itself. Look for the following problems:

Is the leader viewed as powerless or not worthy of respect? The team leader is of paramount importance in rapid product development. This factor is often underestimated, especially by companies new to the discipline of project management. Project management for intense development efforts is far more than developing and tracking a schedule. The leader also has responsibility for ensuring the PTCC balance is defined during project planning; maintaining a thriving team in the face of all obstacles; and constantly judging real results and working actively to keep the project homed in on its PTCC goals.

QRPD describes the right leader as "an obsessed leader who is confident, honest, technical, a model worker, a people person, and has high standards and management ability."

It is important to note that often a "wrong" leader is not replaced because of the negative "demotion" connotations. Overcome that barrier! In tough situations, the individuals usually realize they're wrong for the job and welcome a graceful reassignment; and your project will not be able to recover without a strong, obsessed project leader.

Are people moving on and off the team? Since urgency tends to permeate everything we undertake, we often unrealistically expect our team members to "do it all", not just on one project but on several. However, if time to market is really important, we **MUST** stabilize our project team. It only looks like it costs less to have people split among projects. In reality, the project end-dates will suffer and the company will lose revenue because the projects will not ship on time. In *QRPD* we calculate a "late cost per week" based on this lost revenue, to clarify the dollars associated with time and ensure proper resource decisions. The proper PTCC balance may involve more development cost to achieve shorter development times.

Have there been major technical snafus? Determine whether experts have been used to lower the risk of your technological innovations. To finish your project, find the right experts, even if they're out of house, and make sure the path is clear for the rest of project. Combat any "not invented here" syndrome that may exist. Everyone's lives will be easier.

3. Initiate early cross-functional cooperation

Common causes of project delays: usability, testability, manufacturability, and maintainability issues that are not discovered until late in the project and are serious enough to necessitate re-design. Did cross-functional groups either not participate in earlier definition and design reviews, or did they participate but not understand what they were supposed to contribute? Signs:

You're supposed to be shipping in volume and the technicians can't test the boards.

It's almost time for release and the beta customers are complaining about the user interface.

No one thought to plan for deployment so you have to stop at the end and write a database conversion tool.

The only remedy for people not being involved is to get them involved. No matter how far down the road you are in the project, STOP and get every functional group involved. Review the designs or prototypes in their current state. Have your cross-functional team members look for areas where they will have to spend "extra" money if they don't get what they want. Have them think about past problems they've encountered once products or services have been released. DON'T assume that you're close to the end of your project and the product is good enough as is. Although it might not seem like the fastest way to finish a super-urgent project, a redesign now may be quicker than trying to limp through volume manufacture, and will definitely be less expensive than rejection of your product by the customer.

4. Create a synergistic, mission-oriented environment

It isn't enough to have a good team in place; the team has to operate within an environment that promotes their sense of mission, supports that mission, and allows them to achieve phenomenal results. Assess the following:

Is there a lack of urgency on your super-urgent project? It will carry over to the super-urgent completion schedule! Are individual team members in the dark about the overall goals of the project? Involve them in creating a Project Vision - even if you're "late" in the project! Do they

understand and buy into the PTCC balance? Do they know the cost of delay? Calculate the late cost per week number and make sure everyone understands it.

Is the team mired in unproductive work, slowing them down and probably affecting morale as well? Endless management meetings (especially prevalent for "problem" projects); excessive status reporting; lack of proper communication tools? Have the team members analyze individually where their time is going and rate the usefulness of that time to the project's goals. Then make sure the project leader follows through and removes the obstacles to productivity.

5. Reward participants commensurably

The lack of proper rewards is usually not directly related to huge schedule hit until things get so bad that members of the team leave—and then it's too late to correct the reward issues!. Lack of rewards can also have insidious effects on the team's productivity. Assess the following:

Are some people not dedicated? Not paying attention to milestones? Are their bad attitudes rubbing off on the team? Look for large and small ways to appreciate people. Include milestone celebrations, "going the extra mile" awards, plans for a big party at the end, both team and individual recognition, financial rewards commensurate with the culture of your company, industry, and geographical area.

NOTE: Attention to rewards can be very important to getting your team's momentum back. When you are late and must depend on your team to work hard to recover, an out-of-whack reward system will keep people from buying in and fully supporting the recovery effort

6. Use innovative parallel design strategies

Here the emphasis is on what the team can do technically to ensure success. System design problems unfortunately usually show up late in the project. By the time you figure out that integration is a nightmare because the code is "spaghetti," or not modularly designed, the code is already a mess. Unfortunately, the concept of "system engineering" is often relegated to the aerospace or defense industry; fast-paced commercial projects may throw that thoughtful high-return time out the window. Commandment 6 is meant to get those ideas firmly rooted (and

**Some pages intentionally left out of this preview.
Please register to see the full document.**