Pack Up Your Troubles

Fiona Charles

Originally published on stickyminds.com February 18, 2008.

I like troubled projects. You may think that's crazy, but troubled projects offer tremendous opportunities for creativity and accelerated learning. The more problems a project has, the more opportunities for problem solving!

I've worked as test manager on several troubled project "rescue teams", and discovered a few strategies for surviving—thriving, even—while playing my part in a turnaround.

The Testing Hot Seat

Troubled projects are difficult for everyone, but present particular challenges for testers and test managers. When a project management team has skirted issues, it is often the testing that exposes the cracks. If the project has a history of missing or poorly understood requirements, cumbersome architecture, design problems, or overly rushed coding, issues will cumulate, resulting in a system full of bugs. When the bug rate of arrival goes off the scale, testing slows down, and everyone's anxiety level goes up. Troubled projects are usually late—once they get into testing they get even later.

Just the Facts (Ma'am)

When the emotional temperature is rising, only a clear and calm presentation of facts will bring it down.

Testers tell you they can't test; everywhere they turn, there's a blocking bug. Everyone else says, "Why is testing taking so long? Test faster!"

Your first priority is to get clear information about the quality of the system (or current build) across as broad a range as possible. What's the smallest number of tests that will give you the most information quickly? Taste every significant component fast: a simple end-to-end test through an application, such as the most vanilla sale and return through a Point Of Sale system; or, for a web-based system, a series of quick dips into every page.

How you present the resulting information is critical. Ideally, use a format that gives a clear and comprehensive picture of system quality up front, and then keep using it to show progress and status. I use a colorful graphic table, designing it carefully to show simple categories of information across the entire application. Wherever possible, show the information in business terms, so it's meaningful to all stakeholders.

Play Nice With Others

To turn a project around, team members need to trust and be open to each other's ideas. Nothing destroys openness faster than blaming or dropping bombs on colleagues in meetings.

It can be very tempting for testers to be self-righteous about poor quality. Talk to your testers about the importance of avoiding blame and talk it through with each of them if it comes up.

If it looks as if others are blaming you for delays when the issue is really poor software quality, you may need to prove your point. Be very careful how you do this. Testing can't adequately compensate for quality problems, but it doesn't help a fraught situation to be too blunt. Stick to the facts. Deliver them as unemotionally as you can. And share your report it with the development lead before going public.

Take Care of Your Team

Troubled projects take a toll on everyone, but they're particularly hard on those with the least power. If you manage a team, make sure you support them.

Empower team members. Share information. Seek their input on important decisions. Encourage their participation. Give them responsibility for their testing decisions, and the obligation to explain them. Encourage them to manage their own stress.

Offer your team as much flexibility in work hours as you can. Ensure they know you support their taking time for important personal obligations. If ignored, even small things like getting a haircut can become flash points.

Build in treats. I'm not big on artificial "team-building" events, but work with your team to make the days fun. Sometimes small gestures are the most appreciated: bring in a box of chocolates, or a special team breakfast.

Most importantly, don't let your team be abused, particularly with mindless overtime. Long hours shouldn't be the automatic answer to every project delay. If they are essential to get over particular humps, manage them carefully and judiciously.

Secure Your Own Oxygen Mask First

Taking care of yourself can be the hardest thing to do, but it's critical. Your greatest assets are your heart and your head. Empathy, logic, creativity—all these will help you through a difficult time and offer value to the project. You can't access them effectively if you are short of sleep and cranky.

Get regular, preferably daily, exercise. Walk all or part of the way to work, or ride a bike. Swim or work out. Allow yourself the odd indulgence (from that box of chocolates, say), but make a point of eating healthy, nutritious food that boosts your energy and makes you feel good. Give yourself breaks. If you find you're losing perspective, get out and do something you enjoy, even if it's only for half an hour.

Keep Your Knees Bent

If you want to stay in problem-solving mode, stay flexible. A troubled project is not the time for rigid process over progress. Keep thinking of ways to solve problems, rather than constantly falling back on "the rules". For example, if a build is bad, pair testers with developers to work through the problems, rather than throwing it back. Use or encourage crisis management techniques where appropriate.

Make sure you know your bottom line, preferably before you need it. Mine is simple: I will not lie or provide misleading information. Flexibility doesn't extend that far! That means all important information about the system and project must be captured. Bugs must be logged. We can classify them as "found during development" if that makes people happier, but we can't just find and fix without keeping track.

Remember—It Can Happen To You

You may not seek out a project rescue, but even the best-run organizations can have an occasional lapse, and some companies make a habit of them. You can end up on a troubled project without trying.

Note that I said "troubled', not "doomed". You can learn lots on doomed projects, but I doubt you'll find the gain worth the misery. The best thing I ever learned on a fatally wounded project was to say "good-bye". It's not always easy to tell the difference, but the chief indicator for me is whether my interventions, and those of my colleagues, are making a positive difference. If a project begins to turn around when the team makes a concerted effort to get it under control, then there are good reasons to stay with it. If you keep your head clear, you will know when that is happening and when it is not.

The strategies I've described can help you maintain the right frame of mind to access your own creativity and devise tactical solutions to project problems. If you can do that, you too can enjoy troubled projects and grow your project skills.