



Patterns of Project Management Success

Software vendors can improve their project ROI by following proven paths to project success.

By Rudolf Melik, Tenrox

(Published in February 2008 by [SandHill.com](#))

in this issue

- » [Special Account Services: One Size Does Not Fit All](#)
- » [Patterns of Project Management Success](#)
- » [Running Technical Support with SaaS Solutions](#)
- » [Feedback Gathering: A Process, Not an Event](#)
- » [Selling Service Contracts through Channel Partners](#)
- » [Beyond Satisfaction to Customer Loyalty](#)

DOWNLOAD PDF



Looking back on 2007, I found myself asking, which projects were successful? Which ones disappointed or failed, and why?

The answers were surprising. We identified a set of characteristics which helped predict a successful outcome for our business projects – and another set which forecasted failure. Software vendors can learn from these patterns of project management success and can deliver improved productivity and return from their project teams.

Consistency is Key

If you think about which departments are good at project management, IT generally stands out. The IT group takes service more seriously than ever before. But the software developers who build software are not nearly as focused on the best practices of project management. Neither are the other departments in software companies – or companies in any industry, for that matter.

There are plenty of benefits to good project management but the most important is consistency. Customers expect consistency. Think about when you go to get a haircut. What if the experience was different every time? One time the style is great, the next terrible. One time the salon is trendy, the next time, it's playing classical music. The inconsistency would turn you away.

The same thing happens with customers. Too much inconsistency keeps customers from approaching a vendor for another project. They know that the result might be fantastic, horrible or something in between.

The fact is that successful companies are probably doing a very good job at project management. They deliver a consistent experience for customers and receive repeat business because of it. And if you look at

the companies who are failing, I would bet that poor project management is one of the causes – even if it isn't cited up front. Executives will speak of management failures, marketing mistakes, cash flow mismanagement and hiring errors but the inability to run a project correctly lies at the base of many of these problems.

The Common Denominators of Failure

Looking back at the projects that we considered failures, we found several common characteristics.

- **Poorly Communicated Initial Requirements and/or Scope**

Scope change happens far too often and we all seem defenseless against it. Project stakeholders add new requirements, change their minds or discover new exceptions in the middle of project execution. Functionality that was a “nice to have” suddenly becomes a “must have” or a “we can't live without it” requirement.

The key to avoiding this problem is maturity. Experienced project managers are a precious commodity. The more veteran managers involved, the better the chance that expectations and scope are managed properly.

These experts have been through the failures of past projects and know what to watch for. I believe having veterans on the team is one of the most important aspects of project management success.

- **Poorly Managed Expectations**

On quite a few occasions, we have run into people who expect project management software to perform magic. Once the software is implemented, they are discouraged by how much work there is to do. They underestimate the continuous investments and the scope change management it takes to develop the reporting, dashboards and integration their executive team expects.

On the Tenrox side, our project managers must to do a better job of describing the process, emphasizing (if not insisting on) a phased (instead of a big-bang) approach, and explaining the challenges of enterprise software implementations. In my book *The Rise of the Project Workforce: Managing People and Projects in a Flat World*, published by Wiley & Sons 2007, (read more at www.projectworkforcebook.com) entire chapters are dedicated to business case development, implementation challenges and user adoption. These chapters are a highly recommended read for anyone who is about to start an enterprise software selection and implementation project.

- **The Blame Game**

Project team members (whether inside or outside the company) blame one another when things go wrong. However, in 95 percent of the cases we examined, the failures were a collective effort.

There were mistakes, misunderstandings, lack of expertise, poor scope control, bad judgment, miscommunication, poor project management or a combination of the above by just about every internal and external project contributor. In addition, there's also a lot of complaining going on, everyone interpreting or referencing contracts and written agreements just to try and shift the blame elsewhere.

- **False Starts**

There are a lot of activities preceding a project's initiation. However, failed projects often experience extended periods of lull and inactivity after they start. A desire to optimize resources means that critical resources and attention are pulled away from the initial project to resolve a different crisis or another, more urgent matter. When the time comes to restart the project, the original team is no longer available and not focused on the task at hand.

These delays often kill the project's momentum, scope, and shifts priorities. Soon no one has the same initial drive and passion they once had to execute the project to its completion. Companies need to let their team members focus on a single project rather than asking them to multitask, impacting their ability to focus.

Patterns of Success

In the same way that it was easy to find predictors of failure, the patterns of success stood out as well.

- **Small Teams**

The project team should be very small - no more than 5 people. These small teams develop an incredible bond, become good friends and comrades throughout the duration of the project. They end up working very closely together, often above and beyond the call of duty. Such teams often seem happier and livelier than larger teams and obtain a higher level of satisfaction when the project goals are achieved. As I look back at my own career, there was a small team behind every spectacularly successful project I was involved with – and those were the projects I am the most proud of and spawn the memories that I cherish most.

Small teams can always outperform large teams. When teams get too big, they begin to play the blame game, they become disloyal and politics begin to creep in.

- **A Zero-Blame Policy**

Successful teams have an almost implicit understanding among themselves that no one is to be blamed when something goes wrong. Our research shows that from execution to completion, successful projects face as many (if not more!) roadblocks as failed projects; it is how the team members come together as true partners, with a passion to overcome these challenges which

foretells success or failure. The project contributors must recognize that they are all part of one team. The relationship is not perceived to be “customer/supplier” but professionals working together as peers to achieve one common goal.

- **An Experienced Leader**

Enterprise software implementations can be very difficult if the customer lacks an understanding of the many challenges such a project will face. Battle-tested project champions are able to skillfully navigate the internal politics, user adoption, scope change, deployment and integration challenges.

Too many companies are trying to save money today by hiring less experienced staff – and they pay for it by having poor project results. There is a saying, “A group of sheep led by a lion can beat a group of lions led by a sheep.” As I stated earlier, an experienced team leader has a huge impact on the potential for project success.

- **A Roll-Up-Your-Sleeves Work Ethic**

Somewhat similar to the “small teams” pattern, successful project managers and teams do a lot and delegate very little. You never hear statements like “I assigned the work and it did not get done,” “I asked her to do it, and I don’t know what happened,” “I assumed it was done,” “It’s not my responsibility” ... things just get done. Management hierarchy and delegation are not used as excuses in getting the job done.

- **Use of the POTS (Plain Old Telephone System)**

Poorly run projects are usually characterized by a lot of people sending emails. Some conversation threads go on for days with a stream of never-ending replies and forwards. Successful project teams just meet once or pick up the telephone and talk. Disagreements end quickly and the team converges on a solution that is almost instantly implemented.

Very often, we rely too much on technology. When we look at a project that is not going well, we see too many emails with people trying to cover themselves, “I sent my status. Here is the proof that I did what I was supposed to do.” There is a ton of documentation for failures but success requires no “proof”.

- **Management by Milestones**

Successful teams understand that enterprise software projects never really end. There is always another report, dashboard, enhancement or integration that is absolutely required. The only way to close an enterprise software project is to define milestones and celebrate the achievements, all the while defining new objectives to strive towards. Without recognizing and celebrating incremental successes, there will be no sense of achievement, and stakeholders will feel that the project is in a perpetual state of

execution.

Many projects are managed through metrics but they are not as easy to monitor because they keep changing. Milestones are a lot easier. Define what success means for next month. When you hit that target, celebrate.

There is not enough celebration associated with project completions. At Tenrox, the software development process never ends. We need to define success on a short term basis and congratulate the team when a milestone is reached. That way, everyone feels like they've contributed and can feel good about their accomplishments.

2007 was a magnificent year for many software companies. We completed a record number of successful customer projects. At Tenrox, our customers and project teams have learned how to avoid these patterns of failure and discovered how to emphasize, as well as, encourage these patterns of success. We'll all continue to hone our knowledge of project management success in 2008.

About Rudolf Melik.....

*Rudolf Melik is CEO of **Tenrox**.*

Comments? Suggestions? We would like to hear from you. Please email the editor at sspanews@thesspa.com.

[next article>>>](#)

Distributed by SSPA - 11031 Via Frontera - Suite A - San Diego CA - 92127
©2008 SSPA