InformationWeek CONNECTING THE BUSINESS TECHNOLOGY COMMUNITY

STRATEGIC CIO // ENTERPRISE AGILITY OCTOBER 8, 2014

How To Avoid 'Technical Success, Business Failure'

Here are six reasons such failures happen, along with fixes – and some real-world examples of technical successes gone wrong.

Russ Edelman
CEO, Corridor Company

Anyone involved in IT projects has probably felt the sting of a technical success that is also a business failure. Some may be in denial, but the reality is that this happens frequently, and in most cases it's due to a common set of underlying causes. The primary causes for a technical success and business failure (TSBF) can be blamed on not having an intelligent and grounded change management strategy.

What do I mean by technical success and business failure? Often the measure of failure – or its close cousin, mediocrity – is weak adoption by the intended users, who don't see the value in the technology. Often, the political consequences from an IT and business perspective are that no one will readily acknowledge the emperor's lack of clothing. Nevertheless, we still all walk away from the experience angry, disappointed, frustrated, and asking why.

Here are the top six causes and corresponding fixes, followed by some real-world examples of tech successes that ended in a business failure – or a failure followed by a business success.

The Cause	The Answer
Extended deployments: Analysis paralysis sets in as both the business and IT continually take half steps without ever finishing. Blame lack of time, lack of coordination, or everyone wanting input. The project continually gets delayed rather than rolling forward to some rendition of a solution.	To quote my COO, Jasmin Steely: At a certain point every project or stage must go to a "pencils-down state," even if the system isn't perfect. There comes a time when it is more important to get something out the door that people can use and that can solicit feedback for the next iteration.
Don't know what you don't know: Today's technologies, especially configurable software, offer so many choices they can quickly become overwhelming. And neither users nor IT sponsors necessarily know what the products can do until they see them in action. This translates into projects that take much longer than expected.	Consistent with the notion of pencils-down releases, prototyping and development demos are essential to the configuration/development process. This lets users start to learn what the system can and can't do. Plan on two or more iterations for any project.
Sprint to the finish but forget post-launch change management: Organizations often plan for a project to wrap up once it goes live. Even if that's technically true, business success requires changes after systems are exposed to real-world use.	Plan for focused training and a solid support program after go-live, and include a well-publicized post-launch communication campaign that illustrates the value of the system and how it is being adopted. Include stories of problems solved and quantitative before-and-after results if possible.
Disconnected IT and business users: Even with best intentions, there is frequently a void between IT and business-unit users. IT may pursue an initiative without consulting with the business units, or the business believes that IT doesn't really understand its requirements. The tension often drives SaaS or hosted options as business units try to get as far from IT as possible.	This is a relatively straightforward fix: Figure out a way to work together. IT should actively and intensely pursue business-user feedback through meetings, surveys, questionnaires, and similar techniques. This will allow the business users to actively participate in the dialogue.

The Cause	The Answer
Users don't see the value: If users don't understand how their lives can be made easier, the system won't get traction. The problem is that new systems require some level of change, which means a bit more work and learning for users. Show how there's added benefit from the added effort.	I strongly recommend user surveys before and after each business release: quantitative and qualitative; short, simple questions; with feedback (good and bad) shared with the constituency. Iterative releases let users see progress based on their feedback.
Mismatched user training: Why do companies skimp on training? Because we think people are wonderfully smart and don't need it, and who has the time, anyway? Plus, too often training takes a technical slant over the vital, real-world business orientation.	Make the format and materials for training fit the recipients. Executives might need only short video vignettes and a few minutes of training, while hands-on operators need more in-depth sessions.

Now let's explore how these technical successes can play out in some real-world examples.

No. 1: Global chip maker

Technical success: A global chip manufacturer deployed a system for enterprise use, with diverse language and localization support requirements.

Business failure: Pencils never went down, despite the executive sponsor pushing for it. Old and new users continually weighed in, preventing even the first release from finishing. It was launched for global use without substantive pilot deployment.

No. 2: City Year (education nonprofit)

Technical success: Welles Hatch, CIO of City Year, a national education nonprofit based in Boston, selected a new fundraising tool from Round Corner, which featured a collection of business configurations representing state-of-the-art best practices guidelines for campaign management and automation.

Business failure: Welles and his team leaned too heavily on established processes, which were significantly different from the Round Corner best practices. The project deteriorated, and Welles had to put it on hold until the team gained consensus on the right business implementation.

Business success: The project pause made Welles a tad unpopular at City Year, but it let the organization bring in consultants, who recommended many best practices that were well aligned with the Round Corner configurations. City Year improved its processes and got the project on track.

No. 3: Global financial institution

Technical success: Deborah Reilly, the division chief for information and knowledge management at a global financial institution, built a portal to show information on a country from multiple sources, recent documents, current staff assigned to work on the country, latest published data, last and next mission, and news feeds from outside. Technically, it worked perfectly.

Business failure: Adoption was very low for three reasons: There was no clear business owner (done by committee); information producers actually want some control over which documents appear (important versus recent); and there was no demand from consumers for this information.

Business success: The system did have one feature that was a success. It let users share documents for review and approval without using email. That sped up work – Department A could simply agree with Department B's comments, for example. Reilly cites an added benefit: "Contextual information on how we came to our advice is captured and reusable."

Meredith Henry, director of planning and strategic change at McDonald's, says there are four stages for people to adopt new information systems: generating awareness of the change; understanding (the "why now?"); committing to make the shift; and engaging to adopt the change. Too often, Henry says, technology projects try to go from awareness to engagement, "skipping the two most critical steps necessary for true adoption: understanding and commitment."

Russ Edelman is CEO of the Corridor Company and co-author of Nice Guys Can Get the Corner Office.