★★★★★★★★

Why Most I/T Projects Get Off Track

★★★★★★★★

2010

PCA Business Notes Series

Apart from whether your project is managed by internal I/T staff or by an outside consultant, several factors are common causes for derailing software development initiatives.

In our experience, the reasons why we are called on most frequently to "takeover" a project that is failing include:

- The wrong people on the project (inexperience)
- Av overly rigid process (inflexibility)
- Unrealistic expectations established at the beginning of the project (optimism)
- An overly ambitious project scope (lack of discipline)

Take your pick: any one of these factors can derail the best of intentions and leave your project short on value and long on cost.

175 Highland Avenue
Needham, MA 02494
617.527.4722

The Wrong People

It is probably no real revelation that, who you engage to manage and develop your custom database application project — perhaps more than any other decision — has the biggest influence on the outcome, from both quality and financial perspectives.

This decision is especially risky when you are working with someone for the first time. It is difficult to know whether you have "picked the right horse" until you are well into the project — and everyone knows that it is hard to switch horses in the middle of a race! To the lay business person, most programmers appear to be very knowledgeable, and problems only become apparent (i.e. obvious to anyone) after considerable time and money is behind you.

Many under experienced software developers are "order takers," not consultants, and lack the business savvy or confidence necessary to challenge uninformed assumptions at the outset of a new project. Many are unable to readily discern between technical approaches that look good on paper from approaches that actually work in the real world. And few software developers bring the project management rigor and discipline necessary to drive critical business decisions, and keep projects focused and on track.

And because many developers tend to recommend only what they know (when you are a hammer, everything looks like a nail), it can be hit or miss on very important platform and coding design decisions that must be made at the beginning of a project. Getting early design decisions wrong is like flying to the moon: if the rocket is off just a few degrees at takeoff — be prepared for very lonely journey with no spot to land.

Over reliance on internal I/T personnel is also a common cause of project failure. While most I/T professionals are skilled with properly maintaining desktop applications, keeping networks secure and running smoothly, maintaining back-ups of critical data, etc. these common I/T skills and responsibilities are often irrelevant to the skills necessary to design and develop a custom business solution.

Business professionals often confuse I/T networking, security and DBA skill sets with the development and management skills and business acumen required to plan and execute custom solutions to complex business needs. Network I/T staff and professional database application design engineers are completely different skill sets. Our advice: don't confuse the two!

It is also not uncommon for centralized I/T departments to have control over I/T budget allocations — a situation that can delay or completely stifle line-of-business needs. Empowering I/T organizations with control over business application budgets is like letting Washington DC decide how you should spend your local Town's tax dollars. Our advice: allow line-of-business managers to compete for I/T dollars, and rely on the executive team to decide appropriate budget allocations based upon business priorities.

Overly Rigid Process

Static requirements often kill projects. For many custom development initiatives, getting to the "best answer" is developed thru an evolutionary process of discovery, where subject matter experts and technologists — starting with an initial set of business and functional requirements, and through continuous interaction and iteration — define, test and refine the solution. While this model may seem inefficient, the resulting value which emerges more than justifies the effort.

In our experience, the best informed decisions and optimal results evolve from a fluid process, and less so result from sticking to a thoroughly documented, rigid plan. While business requirements can (and usually are) constant and stable throughout the project (the "what," if you will), the technical solution to satisfying a specific set of business needs and objectives (the "how") must be fluid, and evolve through a common trial-and-error process. It is said that even the best of battle plans rarely survive the first bullet, and nowhere could this axiom hold truer than with software development.

This is "big idea" behind the Agile Development Methodology, which is rapidly displacing more traditional and static software development methodologies. In our experience, the best ideas can be (and often are) unforeseeable at the outset of a project, and thus the best results are often achieved thru an agile process, not a rigid plan. Our advice: don't let a rigid plan dictate people's decisions.

Practical Computer Applications, Inc.
175 Highland Avenue | Needham, MA
(617) 527-4722 |www.pcapps.com

Pictures, Not Words

Over-reliance on words is also a common recipe for failure. Request for Proposals (RFPs) and "detailed requirements specification" documents, project charters, etc. while sometimes helpful in defining high-level project success criteria, can oftentimes fail at defining what ends up mattering most once a project is complete:

- Is the application going to provide all the capabilities I need to get the job done?
- Is the application going to be intuitive and easy-to-use?
- How are the all the individual features/functions going to tie together within a cohesive workflow?

Real answers to these critical questions are most effectively and efficiently addressed by a visual prototype to drive discussion, refinement and consensus. To the contrary, absence of visual prototype *early in the project design stage* only serves to defer these critical issues to later in the development cycle – when fundamental changes to the application are always the most expensive!

An effective Prototype is highly leveraged: it should give everyone confidence that you're on the right track; it should confirm functionality, workflow and ease of use; it should provide the primary means to determine resources, schedule and cost; and finally, the Prototype should provide the source-code to drive the development phase.

Our advice: it's always to good idea to spend time up-front developing a project charter, but don't overcook the documentation, and don't let the documentation or rigid plans get in the way of making real progress.

Unrealistic Expectations

It is human nature to be optimistic. When it comes to custom application projects however, you can rarely go wrong with a healthy dose of skepticism!

Our advice: get 2 or 3 estimates from different developers, average them, then ask yourself: "If this project cost me twice as much, took twice as long to complete, and I ended up with half the functionality I needed, would I still start the project?" Never mind the numbers here — they vary depending upon the nature of the project and people involved — but if you're thinking this way, you are on the right track!

In our experience, the only reliable answers to "How much is this going to cost?" and "How long is this going to take?" require a Prototype Design phase that frames in the project, before development begins. Be wary of setting cost and schedule expectations (or believing them!) before a Prototype is vetted and approved by all involved. Once expectations have been established, it is very difficult to un-ring that bell.

PCA Recommendation

Take a close look at your needs and situation for the above risks. Is cost the primary factor for a short-term solution, where one or more of these risks are manageable? Or, is this a strategic project, where you (and others) will need to live with the results for quite some time?

Practical Computer Applications, Inc.
175 Highland Avenue | Needham, MA
(617) 527-4722 |www.pcapps.com