

# Implementing Global IT Systems: DIY Projects

by Ted Farrington

*With over 30 years of experience working in research and development (R&D) for a variety of companies and industries, I've seen my fair share of information technology (IT) systems implemented to support various global processes such as product lifecycle management (PLM), phase-gate, portfolio management, legal and finance.*

*Having observed these projects from all angles, I began to notice a pattern behind some of the [most common mistakes being made during the implementation of large global IT systems](#) in support of R&D or other business processes.*

*This series provides leading practices for avoiding the top ten most common mistakes.*

## Mistake #10: DIY Projects

### ?Find a trusted advisor to manage expectations and deal with vendors

Above a certain project size, a third-party integrator becomes a necessity, especially when neither the software vendor nor the company have the resources required to do the job. But in many cases, even for smaller projects where the company thinks they have resources, teams should engage a third-party integrator. Perhaps the integrator just manages resources largely supplied by the client. But, there are still two very good reasons to engage a third party for your project:

#### 1. Managing internal expectations and requests

[Scope creep is real.](#) Companies can be their own worst enemies and expectations can be hard to manage. Internal leaders are often too accommodating of requests from various parts of their organization, including:



“Can we include this in the project?”

“Can these people also have access?”

“Can we customize the system for our team?”

An internal project leader may feel the need to say “yes” to increase engagement and because he/she needs to work with these folks after the project is done. But accommodating these types of requests without thinking them through causes scope creep and therefore budget problems. System customization is a long-term issue because future system releases may not be compatible with customized modules.

Any good third-party integrator can say “no” as needed, and will take a tough “no customizations” position from the get-go.

## **2. Managing the vendor**

Someone will need to play hardball with the software vendor whenever expectations are not being met. An internal project leader, who doesn’t do this regularly, won’t know when and how far he/she should push the vendor. An experienced third-party integrator has probably worked with the vendor before and has a much better understanding of their capabilities, limitations and inner workings. This comes in handy when the project faces challenges, as all projects do!

## **Don’t even think about writing your own code**

The worst example of DIY projects is writing your own code when something meeting your critical needs is available off the shelf. I’ve seen companies try to maintain internally-developed code after the developer has left the company, and it’s not pretty. No one is dedicated to maintaining and upgrading the system and no commercial systems will want to interface with the home-grown code. The short-term cost saving is overwhelmed by the long-term risks involved. A good third-party integrator will advise against such DIY efforts and play a critical role in keeping the project in scope, on track, and successful in the long term for both large and small projects. The budget required for a third-party integrator will easily pay for itself in time and effort ensuring smooth sailing through the entire project.

---

Stay tuned to discover leading practices for avoiding these ten common mistakes. Being mindful of the challenges and solutions discussed in this series will greatly increase the chances of your next project becoming a sustainable success.

## The Entire Top Ten List:

<a href="#">1. The "Global" Roll Out</a>	<a href="#">6. Jumping the Gun</a>
<a href="#">2. Playing the Shell Game</a>	<a href="#">7. Skipping Stakeholders</a>
<a href="#">3. The "Fix All" Solution</a>	<a href="#">8. Skipping the Dress Rehearsal</a>
<a href="#">4. Ignoring Inconsistencies</a>	<a href="#">9. Self-Gathering Data</a>
<a href="#">5. Missing the Point</a>	10. DIY Projects

## Download the eBook:

[Top Ten Mistakes Made Implementing IT Systems and How to Avoid Them](#)

*Originally published on June 12th, 2017*

[What's your view? Add your question or comment](#)



### Ted Farrington

[ted.farrington@kalypso.com](mailto:ted.farrington@kalypso.com)

Ted's 35 years of research and development experience were built in the CPG industry, where he held leadership roles in advanced research and R&D. As a fellow at Kalypso, he uses his years of experience in breakthrough innovation, research foresight and R&D business processes & systems to support clients.