

Implementing Global IT Systems: Don't Skip the Dress Rehearsal

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 #8: Skipping the Dress Rehearsal

Perform real-world testing to ensure smooth sailing

Throughout my experiences implementing global IT systems, I was not the IT expert. I usually represented the customer—typically a global R&D organization. Despite that, I learned very quickly just how important system testing is to the success of the project. It's always easier to fix a mistake caught in testing than one caught after go live. Believe me – I know firsthand. While any testing is better than no testing, I recommend gauging where your organization is to understand what needs to be done to work towards the highest level of testing, and ultimately, success.

Here are the four levels of testing I've seen over my career:

Level Zero: "Turn it on, it will work"

I actually saw one project go live with essentially zero real-world testing! The system was designed to automate the global specification approval process, which had always been done manually. When the system went live, a basic raw material substitution that would require two or three manual approvals, triggered over a thousand required approvals in the automated system! Oops! User feedback was fast and direct, to say the least. No project should go live this way. Every project needs to provide a means for post-launch feedback, as even the best testing protocols can miss something. In the true tradition of Dilbert, the team that launched this mistake was recognized for the great work they had done (obviously not determined by popular vote).

Level One: Positive testing

This is a small scale, but thorough, testing of the system's functionality performed by trained lead users. A good test requires a lot of planning, especially for a global system. Every campus should participate and every system functionality should be tested. The question being addressed here is, "If the users do what they're supposed to do, does the system perform its functions properly?"

I've seen positive testing done poorly, where folks who design and install the system just "kick the tires" of a few key functions, go live, and then resort to Level Zero to deal with complaints. I've also seen it done very well, with all functions and locations tested. Guess which system had a smoother implementation?

Level Two: Stress testing

As an R&D professional, I had never heard of this type of testing until I had the opportunity to watch a global PLM system as a member of the PMO. Since then, I've always pushed for it. My IT partner leveraged his relationship with a large software company to get this done. If we expected 100 users to be on the system simultaneously, a software package was used to simulate 1000 simultaneous users. After it was done, my IT partner came to me and said, "Ted, you have a memory leak!" To which I responded, "What do you mean? I'm only 50 years old, my memory is fine!" That's when I learned that a "memory leak" is when a user (real or simulated) logs off the system, but the memory they've been using is not entirely freed up for the next user. The 10X stress test saw this accumulating memory loss. The system would have crashed due to insufficient memory a few days or weeks after going live. Without stress testing, we would not have known to fix the problem before going live.

Level Three: Negative testing

This is an important level of testing that rarely gets done at all, much less done well. What seems unfortunate to me is that with system training becoming more online and less thorough (have you ever raised your hand to ask the online training module a question?), this testing is even more important. While Level One asked how the system would perform given correct inputs, Level Three asks "How will the system respond if the user makes mistakes?" Examples include data entry errors, trying to advance without all required information and logic conflicts. Any necessary tweaks can be proactive rather than reactive.

Through experience, I learned how important testing is to the success of the project. As the testing process gets more stringent and more levels of testing are completed, the project implementation goes more smoothly. Skipping this step is like skipping the dress rehearsal before a big performance. A few test runs before launch can save both you and all end users a lot of pain and frustration in the long run and prevent an onslaught of tomatoes.

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:

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

Download the eBook:

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

Originally published on May 30th, 2017

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



Ted Farrington

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.