

# Upgrades

## Achieving Long-Term Transformational Success with PLM in Life Sciences, Part 15

by Dave Hadfield

In this series, we have considered why life sciences companies aren't achieving the full value of PLM. So far, we've covered building the [business and PLM strategies](#), and we've considered the [hurdles companies are likely to face](#) while building the strategy. In the last entry we looked at the [common issues of time and cost](#). Are we past those yet? Okay cool! Next we'll consider a particularly tricky strategy question; how to plan for upgrades.

Upgrades update the software with the latest offering. Companies upgrade for many reasons, but the most common reason is to maintain support from the vendor.

Don't underestimate the risk and cost of upgrades; they are often quite painful and can degrade internal support and potential value over the long term. Sadly, I have seen this happen firsthand after companies have invested tens of millions into PLM, only to be stumped by a painful upgrade and subsequently turned off. The program dies under its own weight. For our [pioneers](#) heading out west, they reached the Mississippi but didn't get across.

There are a few techniques I can recommend.

First, consider how to reduce upgrade frequency over the long term. One of the big drivers for upgrades is when PLM systems are coupled with authoring tools, especially CAD. To get the new CAD program many still believe we have to bring along all of PLM. This is both the tail wagging the dog and a throwback to old school product data management (PDM) thinking. PLM outgrew its CAD/PDM roots many years ago, yet may still think of it this way. My recommendation is to consider carefully separating the PDM tool from the PLM tool, so that we can be more agile with CAD upgrades and let CAD/PDM tools do its job. This may sound counter to an integrated strategy, but in this case I think it's an acceptable compromise. Instead, integrate the PDM tool with the PLM tool (potentially very loosely). If your vendor doesn't have a separate PLM tool, there probably is little stopping you from having two instances.

My colleague [John Hubert](#), a PLM veteran of nearly 30 years, takes this point further. His recommendation is to consider keeping upgrade cycles to five years or longer. Some people might disagree, but John and I have seen companies be successful running the same instance for as many as ten years. Yes, the eventual upgrade will be painful and companies may face support problems, but the alternative - upgrading every two to three years - can be detrimental to progression of the overall roadmap. Base PLM packages don't evolve as quickly as you might expect; it's taken years to get to the kind of advancements I spoke of previously. Pick a point in time, where the version has stability, and consider the pros and cons of staying with it for a longer period of time. Even if direct vendor support lapses, you may be able to get by with other methods, so it's a matter of evaluating the risk.

Select PLM vendors address the upgrade issue head-on, and will upgrade your solution (including all your customizations) as part of your subscription. In fact, they may even guarantee your solution will upgrade so you never have to worry about configuring and customizing the system to work exactly as your business needs. Look for this kind of guarantee before you buy, and [contact me](#) if you'd like to know which vendors offer this. If your vendor doesn't - and most don't - they still may be the best fit for many other reasons, but you will pay a much bigger price for upgrades at some point.

Finally, consider how to build out or implement end-user automation capabilities within the product lifecycle. We have or will select our core platform, but that doesn't force us to buy every add-on module that automates individual processes. This is the topic for my next entry (to build or to buy automation capabilities). Answering this question will have a profound impact on the eventual cost of upgrades, usability, organizational change and other variables.

---

### More In This Series

The Missed Opportunity and How We Can Overcome It

- [PLM, the Great Missed Opportunity in Life Sciences](#)
- [PLM Pioneers](#)
- [Adoption Obstacles](#)
- [It's Time to Get Excited About PLM in Life Sciences](#)

## The Business Benefits

- [Risk Reduction](#)
- [Cost Reduction](#)
- [Innovation Enablement](#)

## The Basics of Technology and Strategy

- [Technology Choices](#)
- [The "PLM Program" is Where We Went Wrong](#)
- [Getting to a Business Transformation Strategy \(with an Emphasis on Product Innovation\)](#)

## PLM Strategy

- [Setting the Stage and First Release](#)
- [Beyond the First Release](#)
- [Automation Initiative Prioritization and Grouping](#)

## Solving Coming PLM Strategy Problems

- [Time and Cost](#)
- [Upgrades](#)
- [To Build or To Buy?](#)
- [Vendor First or Strategy First?](#)

## Making it Real – People, Governance and Methodology

- [Transformational PLM is Hard – It's Time to Rally the Troops](#)
- [Amazing PLM Governance](#)
- [Ten Traits any PLM Team Must Have](#)
- [Three Characteristics of a Successful Implementation Methodology](#)

*Originally published on August 11th, 2015*

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

## About the Author



Dave Hadfield

[dave.hadfield@kalypso.com](mailto:dave.hadfield@kalypso.com)

Dave brings over 17 years of experience in product lifecycle management (PLM) to Kalypso's clients, with deep expertise in the medical device industry.