

Written BY Phillip Brown // May 23, 2019

Complex Software Implementations Can Take A Village— Here's Why



Successful initiatives are very much about the partnerships that form coalesce between vendors, consultants, systems integrators and, of course, the ultimate end-user, the client.

So, you've selected the new "Wonder System." Congratulations! It is a big decision, and one you'll probably need to live with for at least the next 10 years.

We've all heard the war stories of failed implementations. Some well-documented examples include Target US, who decided to implement a new ERP system in Canada that contributed to the business closing 133 stores and firing 17,600 employees. A failed system consolidation cost HP \$160 million in order backlogs and lost revenue. Nike had a similar experience, losing \$100 million in revenue with stock prices dropping 20 percent. A \$98 million payroll system at Queensland Health in Australia came in at \$1.2 *billion* by the time it was finished. There are many more stories of failed implementations throughout the world—some public and some kept quiet. While I'm in insurance, and I hear these stories all the time, the same considerations are equally applicable to implementations across industries. So, why do we see so many failures, and what's the secret to a successful implementation? In my experience, it goes well beyond the selection of the technology itself. Although that's obviously critical, it's very much about the partnerships that form coalesce between vendors, consultants, systems integrators and of course, the ultimate end-user, the client.

Factors That Contribute to a Failed System Implementation

A major attribute we often see in failed implementations of all sizes is an organization trying to do too much with over-stretched internal resources—but more on that later.

Another major contributor is oversell of functionality by vendors. When selecting software, it really is a "buyer beware" situation. The key is to determine whether the functionality being demonstrated is available in a current or planned release, and to test it during selection. Product roadmaps carry disclosure messages for a reason: the functionality stated on the "roadmap" might just be an idea that is never really going to be delivered. Ideally, you will have end-to-end demonstration scenarios with inputs and expected outputs. If you need a system to generate results, ask for them to be demonstrated, throw in changes to the input data, and ask the vendor to recalculate to make sure results have not been hard-coded (yes, this actually happens a lot). If you're investing in a cloud-based system, insist on cloud-based demonstration rather than the pre-sales executive's high-powered laptop. During your demonstrations, throw in wild cards like having the demonstrator set up new items, add new rules, change structures, and add new reports. In other words, demonstrate all the items that the demonstrator told you were going to be "highly configurable by business users." If the experienced demo person can't do it, how will your business people? Don't believe the vendor hype—ask them to put up or shut up—no vaporware or "futures" should color your decision.

Okay, so you are confident the software you have selected is a fit for your purpose and you've made a solid choice. The next stage is critical: the selection and structure of the implementation team. Typically, this will include key people in your organization from the business and IT units, along with the vendor's subject matter experts and configuration team. This is normally a good combination in small implementations; however, a two-party implementation team can lead to a lot of issues in larger, more complex implementations where internal resources are already stretched, and staff still have their day-jobs to perform.

Often, a third party adds tremendous value. Ideally this will be a professional services organization that is heavily engaged in the overall planning, delivery, or support of core systems throughout the organization. There are two key aspects to this kind of implementation support: front end, such as running the project management office and helping define requirements for the new solution; and back end, the integration, data conversion, and testing. In many cases it might be two vendors—for example, a consulting firm such as PwC on the front end and a systems integrator such as Cognizant on the back end.

How Does This Type of Partnering Work?

Having a third-party partner run the front end of the project means the project will be structured, managed, and reported on in a way that is familiar to your organization, as the methodology is typically already being used by the partner in other projects throughout the organization. The partner can help the vendor team design a delivery methodology that is a blend of the vendor's approach and the customer's requirements. For example, if the vendor is traditionally "Waterfall" and the customer is "Agile," they will broker a model that ensures the capture of critical vendor waterfall elements with agile delivery components that supports a hybrid delivery model. This approach will include setting up the steering committee, ensuring project documentation, coordinating status reporting and risk management, and ensuring that all parties are on the same page in a pre-project phase before the implementation starts in earnest.

Having a partner on the back end ensures that the vendor settles into the technical environment of the customer, understanding the internal system architecture, and various methodologies for data conversion, integration, testing, and quality assurance protocols that vary considerably from customer to customer.

As mentioned earlier, it is great if front-end and back-end are provided by a single third party. If not, you might end up with a team of four or more parties. One significant factor in making this work is getting a clear understanding of the roles and responsibilities of all parties—typically through a mechanism like a RACI model—and ensuring that this same document is embedded in the contractual documents of all parties, so everyone is on the same page.

Based on my experience, it is easier to contract with the individual parties for delivery of their own resources rather than trying to broker a master agreement with sub-contractors reporting to one "Master Partner." Often, a small vendor can't accept the terms and conditions offered by a much larger partner that has massive scale, resulting in significant discounts, benefits, and performance penalties. This mismatch leads to months of extra negotiation, and the parties are rarely happy from a contractual perspective. Individual agreements are far easier to manage, and with them, the individual parties are more willing to be flexible on a day-to-day basis throughout the project.

The next major hurdle will be the assignment of your own resources to the project. Ideally, you want full-time dedication of human resources to the project, but this is not always possible. For example, the work of generating commissions and handling issues in a new commission system is constant, so you need to support the resourcing in some way. This will mean part-time allocation of resources to the project, or looking to your partners or other external resources, such as temporary agencies, for assistance. Something that comes up a lot



is building out test cases for user acceptance testing. This is a tedious job for current staff in addition to their daily jobs, and the partner resources who helped to gather business requirements might be able to help with this task.

You will also need significant business and technical resources allocated to user acceptance testing. You will want to ensure that adequate time is allocated to testing—that issues are tightly managed—or you are guaranteed significant project delays. Again, your partners might be able to help with this. The same goes for data conversion, integration, testing and quality assurance tasks; these are all technical items that a back-end partner should be in the perfect position to support.

In the end, someone must perform these tasks, and there is no magic bullet. Rather than risk over-burdening your own people, the additional cost of third-party support is well worth the investment, as it reduces project risk significantly and greatly increases the ability of the project team to deliver on time.

Face to Face

From a working perspective, it is important to bring your parties face-to-face on a regular basis, as this helps to build a cohesive team. It is also important to have the vendor and your partners get together to educate the respective teams and discuss approach and methodology prior to the start of your engagement. This way, there can be preliminary alignment, and it's not coming out of your project budget. Once the engagement starts, consider holding team events at least quarterly. If you set this expectation during negotiation, the parties can factor the expense into their estimates so that traveling to attend team events is not an issue. If not, you may not have the full participation you'd really like.

Finally, it all comes down to having a solid communication plan between the parties to keep everyone synchronized throughout the project. The relationship between the vendor and one or two partners is more complex than the average marriage, so there will be issues. Regular communication will help avoid issue escalation and should keep everyone on track. There are other factors that impact project success—we will discuss them in a subsequent article—but I hope that this will set you up for success in even the most complex of implementations.

Phillip Brown is Vice President of Global & Strategic Sales at [VUE Software](https://vuesoftware.com). He can be reached at Philip.Brown@vuesoftware.com.

End of Article

Original Link: <http://iireporter.com/complex-software-implementations-can-take-a-village-heres-why/>



VUE Software.

About VUE Software

VUE Software is an innovative provider of performance-driven solutions built exclusively for the insurance industry. With over twenty-three years of experience in Life & Annuities, Health and P&C Insurance, VUE Software is one of the most experienced and established Insurance Distribution Technology providers in the business today. VUE Software is leading the Distribution Modernization movement, bringing clients a solid path to revenue growth and competitive advantage.



VUE Software is the product division of Computer Solutions and Software International, LLC. (CSSI). This Featured Article is for informational purposes only. CSSI MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. VUE Software and the VUE Software logo are either registered trademarks or trademarks of CSSI in the United States and/or other countries. All other trademarks are property of their respective owners. © 2014 Computer Solutions and Software International, LLC (CSSI).