JOURNAL OF CLINICAL RESEARCH BEST PRACTICES

Vol. 15, No. 9, September 2019 "Happy Trials to You"

So You Want a New Computer System?

By Erika Stevens and Brian Chappell

Introduction

The increasing complexity, regulatory requirements, data collection and document volumes, and pressure to decrease the time and cost of clinical research have combined to force clinical research sites, sponsors and CROs to modernize and expand their use of computer systems. The increasing complexity and interaction of these systems increases the challenge. The net effect is that modernizing computer systems is becoming both more important and more difficult, while the lifetime of new systems becomes shorter.

Implementing a new computer system or upgrading an existing system — protected behind a "paper wall" from other systems — is no longer viable. You must now consider its real-time, automated interactions with other systems that might, themselves, be in the midst of change.

While the technology and standards to collect, store, manipulate and communicate data and documents between systems is advancing, there is still no substitute for the time-honored process of planning, designing, developing, testing, integrating and deploying computer systems. What *has* changed is that the business exigencies are becoming ever less forgiving of shortcomings in this process.

In this article, we will discuss the basics of this process, understanding that every step in the process is exponentially more complex than can be described here.

Phase 1. Planning

In the planning phase, you will reconcile the high-level objectives for the *future* of your organization with the on-the-ground realities of *today* to create high-level specifications, timelines, costs, etc., for modernizing your system.

The first step is to create a cross-functional steering committee empowered to make decisions under the supervision of an executive sponsor, whose primary roles are to guide the steering committee and protect the project from external challenges.

The second step is to describe the current state, the desired future state, and the gaps between the two.

The third step is to develop a plan to move from the current to the future state as quickly and inexpensively as possible, with a minimum of disruption and risk. You will likely be torn between two unappealing options: incrementally improving an existing cumbersome and inflexible legacy system or betting on a brand-new system with unpredictable consequences.

While you will not be able to develop a user training and support program until the system is close to deployment, your plan should incorporate these essential elements.

Your plan should address not only the internal requirements of the project but also how you will engage and communicate with the stakeholders during the project. If they feel excluded, deployment might be unpleasant for all concerned.

Phase 2. Design & Validation

In the design phase, you will refine and detail the specifications for the new system and its interactions with other systems, essentially creating a blue print for the new system and "city map" showing how it interacts with other systems. A modular design will increase flexibility for swapping in new components down the road. Computer systems must always be sensitive to current and intended workflows.

You will then validate the design with a broad cross section of intended users, "customers," and other stakeholders who will be affected by it.

Don't be surprised if several iterations of design and validation are required, noting, however, that system requirements will be evolving during the entire process, so, at some point, you will have to mark the design "final" and hope it can be realized before it becomes obsolete.

Phase 3. Development

Now that you know with some specificity what, exactly, needs to be built, the first step in the development phase is to create a detailed *development* plan. This plan should include a timeline with major milestones, function and feature priorities, personnel requirements and assignments, coding standards and code libraries, a user-interface style guide, test methods and cases, risk identification and mitigations, and a many other details. A high percentage of software development projects fail, so consider acquiring an existing system from a software vendor (after thorough due diligence), even if it does not meet 100% of your requirements.

The second step in the development process is to build the system, iteratively testing each piece to make sure it works as intended and does not break something else in the system.

Phase 4. System Testing

Congratulations! You have built a computer system. It is now time to test the system as a whole to make sure it works properly. Even if you have included actual users in the design and development phases, user testing in this phase will almost certainly reveal new requirements and various dissatisfactions, accompanied by cries of anguish and distress. Someone, perhaps the executive sponsor, will have to decide which of these issues must be addressed in version 1.0 of the system and which must be deferred to a future release. Remember that perfect software systems are as rare as unicorns, and today's unicorn is tomorrow's donkey.

Phase 5. Integration

Now that you have designed, built and tested your system, it is time to see "how well it plays with others." Unfortunately, no matter how interoperable your system design, it will have to work with other systems that will, shall we say, have their own quirks — and the only solutions might be workarounds too ugly to discuss in public. Someone will probably ask you "Why didn't you think of this before?" even if that other problem system has unexpectedly changed.

The thought has probably crossed your mind by now that developing your own software might a fool's errand. That may very well be the case, but it might be the only way to get the functionality you really, really need. Fortunately, there are plenty of consultants and companies that can advise your internal team or do the entire project for you. There are also plenty of software companies that can provide commercial software systems that might be good enough, affordable and a lot lower risk. Keep in mind, however, that you will still need to integrate that system with your other systems, with essentially no control over the data design and communication protocols. And, you can say goodbye to a consistent user interface across all your systems.

This is where the debates over commercial vs. homegrown software and best-of-breed vs. all-in-one software become interesting.

Phase 6. Deployment & Maintenance

Now that you have integrated your new computer system with the others it interacts with and everything appears to be copacetic, you can deploy it, i.e., "put it into production," preferably in phases, if possible. The number of users and use cases will increase exponentially, so expect a deluge of problems and new requirements. Collecting, understanding and addressing the ongoing stream of new requirements and "bugs" is called "maintenance."

Conclusion

Deploying a new computer system can enhance the flexibility and adaptability of your organization. It can also be a stern test of your organization's ability to manage change, an essential capability in the modern clinical research industry. Organizations that can effectively implement new computer systems have a huge advantage, possibly decisive, over those that cannot.

Relying on the arcane expertise of software professionals for the survival and prosperity of your organization can be unnerving, so the more you can learn about the process, the more likely you will be able to guide and manage it.

Authors

Erika Stevens, MA, is Principal at Recherche Transformation Rapide. Contact her at 1.646.660.0000 or Erika@recherchetransformationrapid.com.

Brian Chappell, MPH is Strategy Business Operations, Manager at Ascension Healthcare. Contact him at 1.646.323.6165 or Brian.chappell@ascension.org.