

Information Technology and Better Health: Overcoming the Risks

Author Commentary

CHAPTER ABSTRACT: Information technology (IT) holds the promise to transform academic health centers by making healthcare more effective in meeting the needs of individual patients; but, doing so demands careful planning to surmount the challenges of deploying complex systems. This chapter provides a system engineering approach for anticipating and reducing the risk of IT projects. Central to this approach is defining requirements from a user perspective at the time a project is conceived and using those requirements to ensure that the system operates as intended, meeting the needs of the enterprise and its patients.

Many risks are associated with IT projects, and CEOs of academic health centers need to pay attention to those risks. There is a methodology for managing those risks and if you implement the methodology, those risks can be greatly diminished.

For example, I have found that it is easy for various types of IT projects to go off the rails due to absence of planning and/or failure to communicate among the many parties that participate. This is particularly an issue for large-scale projects. Consider integration of the electronic health record—a large-scale project in which most health centers have engaged. In many ways, the EHR drives the organization and touches every piece of it. It needs to be designed so that it supports the organization's mission. That isn't just about implementing the software itself. Making the EHR work in purely practical terms means that it works from a functional standpoint in supporting the mission. Aligning those two goals while still keeping the project on time and on budget is a universal challenge. Managing the associated risk is critically important.

IT projects demand technical skills from your IT staff. But who are the right people? I believe there are two alternatives for project oversight. Either might work under the right circumstances. You always have to involve business and IT, and so who ends up leading and driving the project for completion is an important choice. One option is to put someone with strong business knowledge in charge of the project—versus someone who is stronger on the IT side. There are two perspectives on that issue, however my preference is more from the business side.

Managing large-scale IT projects in academic health centers should start with creating the requirements. It is very important to take a comprehensive look across the different IT platforms, prioritize them, and assess how they all need to interface with each other. It is extraordinarily difficult to get this right, in large part because software is likely acquired from multiple vendors. The software should communicate effectively, much the same as you want your staff to communicate. That needs to be conceptualized at the start.

Rather than deciding to buy software here and there, with the idea that you can figure out later how they can best work with each other, a broad systems vision is necessary from the onset. What are the pieces of technology that you would like to implement over time? What is the strategy for acquiring those pieces of software? How do you coordinate their development over time? How do you make them work with each other? I view that kind of planning as a crucial first step that needs to be undertaken prior to going out and executing a contract with a vendor. While it is often tempting to simply get a project started, planning in advance is vitally important.

Randolph W. Hall, PhD
Vice President of Research
University of Southern
California

Over the last five years, USC has invested significantly in software. In financial systems, research, HR, personnel management, and health records, almost every system in the university has been changed to something new. Implementation doesn't always work the way we hope it will work. All kinds of challenges come up along the way. When we have tried to implement too many IT projects at once, software fatigue comes into play. Employees need to learn pieces of software and change sometimes ingrained behaviors. Looking across all of the major systems to be implemented, it is critical to address them over time rather than try to tackle them all at once.
