

# How to Avoid the 3 Top IT Project Risks

For a number of years, the federal government has been on a mission to reduce waste and enhance efficiencies across departments, including IT. A recent example of this is President Trump's [innovation team](#), which is powered by some of the tech industry's biggest names and echoes a similar program that [President Barack Obama initiated in 2008](#). The goal for both programs: streamline government, raise accountability and improve performance -- without sacrificing innovation.

Unfortunately, these efforts have fallen short of the mark, at least in terms of IT. In fact, according to the CIO Council's 2017 [State of Federal Information Technology report](#), 43 percent of the federal government's \$80 billion in IT projects cataloged in September 2016 were listed as over budget or behind schedule.

Inadequate project scoping and estimation are key factors behind this alarmingly high percentage. As outlined in the Government Accountability Office's [Cost Estimating and Assessment Guide](#), "Software is a key component in almost all major systems the federal government acquires. Estimating software development, however, can be difficult and complex. Most often, creating an estimate based on an unachievable schedule causes software cost estimates to be far off target."

The GAO guide further highlights how bad estimates stem from "a lack of understanding of how staffing, schedule, software complexity, and technology all interrelate."

GAO advocates for a scope-based parametric estimation approach, which has proved to be much more accurate and effective in supporting software project planning compared to role-based and task-based estimation processes. Role-based approaches rely on the knowledge of subject matter experts, but that knowledge can vary wildly in terms of its accuracy. Task-based approaches require an enormous amount of detail that's often very difficult to incorporate into the critical early planning stages of a software project and are not conducive to change.

Conversely, the scope-based parametric approach provides stakeholders with a more accurate and flexible scoping and estimating solution that is grounded in historical data. Estimators can create a number of "what if" scenarios early in the project's life cycle so they can better plan for the inevitable challenges and roadblocks that will occur during development. Knowing these ahead of time allows them to more accurately scope costs, timeframes and staffing required to finish the project on time and within budget, thereby avoiding some common mistakes that lead to overruns.

Let's take a look at some of these pitfalls and how parametric estimation can help government CIOs and their teams avoid these traps.

## **Inaccurate Estimates**

One of the biggest mistakes CIOs make in IT portfolio planning is not using a robust, scientific method to estimate software project duration and effort demand at the outset of a project. Too many rely on informal role-based or task-based estimation approaches to answer key questions: What's a *truly realistic* timeframe for completion of this project? How many people will we *really* need to make this work? How much will it *actually* cost?

The danger with these approaches is that they are often overly optimistic and fail to consider important factors that will impact the project's completion, such as complexity, team efficiency or requirements growth. The

result could be a [HealthCare.gov](https://www.healthcare.gov) or [Defense Integrated Military Human Resources System](#)-level debacle, a project marked by poor quality and cybersecurity deficiencies or, worse, the need to completely scuttle a project.

Parametric estimation involves a top-down process that allows agencies to leverage historical information and forecasting models to more accurately assess projects from the beginning. It provides a flexible framework that can be re-forecasted as requirements change. This helps CIOs address potential risks before the start of a project and mitigate those risks throughout development.

### **Overstaffing of Projects**

In 1975, software engineer and author [Fred Brooks](#) wrote that “adding manpower to a late software project makes it later.” Yet in 2017, the practice of throwing more bodies at a problem still persists. In fact, [research](#) shows that overstaffed projects result in more defects, increased costs and the need for rework. They also often take longer to complete.

The reason is simple math. Fewer people means a more streamlined and efficient process. Adding more people to a project increases the necessary communication paths and adds to the complexity of the process, leading to miscommunications, software defects and increased costs.

CIOs should model appropriate staffing levels in relation to their IT portfolios during the planning stages of every project. They must analyze and assess strategies to ensure that the staff they are committing to a project accurately matches their demand vs. capacity planning models. While some teams may need to be larger than others, smaller teams will be more efficient, nimble, productive.

The “smaller team size is better” rule should apply even with projects that may be running behind schedule. Adding more people to a late running project will only compound the problem and slow things down even more. Aside from adding to the cost of the project and introducing new lines of communication, developers already working on the project will need to take time to train the new people -- slowing things down even more.

### **Denial of a Project’s True Status**

No one wants to admit when a project is in trouble, but a system that allows for quantitatively measuring of progress enables a CIO’s team of project managers to promptly take corrective action to keep projects on track. Conversely, waiting until the end of a project can have an adverse impact on any follow-on activities. Look no further than the [problems plaguing the F-35 Joint Strike Fighter](#) program to see how waiting too long to course correct can cause major issues.

If there’s even a hint that things are falling behind or going wrong, managers should stop, re-estimate and take an honest look at what it will take to correct any issues. If what they discover is unacceptable, they may need to reduce their scope, examine different options and come up with a more realistic plan.

Fortunately, agencies that have implemented agile development methodologies are already ahead of the game in their ability to leverage parametric estimation and tracking. Agile embraces a scope-centric approach to estimation and planning that recognizes that the longer the duration of a project, the more likely it is to be impacted by changes. Agile projects are built in short iterations, which means that quantitative data on product construction is readily available on a routine basis for tracking and forecasting in a parametric tool.

Parametric estimation provides agencies with a solid foundation for project success that doesn’t tie them into a rigid framework. This combination of flexibility, reliable historical data and proven mathematical algorithms is the key to helping agencies gain efficiency, stay on track and still deliver innovative software programs.