Successful software strategies

Highlights from the 18F Derisking Guide

Traditional approaches in government have not served us well: 29% of large government IT projects fail. But only 14% of small government IT projects fail. How can you use this knowledge to your advantage?

Sizing your IT project

When taking on a large IT project or modernization, you can cut the risk in half by changing the way you conceive and manage it. Make your project smaller and break it up into smaller, independent efforts. We call this modular contracting. When we use modular contracting, we see improved quality, lower risk, and projects delivered on time.

The fundamentals of modular contracting:

- Don't enter into contracts longer than 3 calendar years in duration.
- Cap each team's budget for labor at no more than $10 million total; that typically means between $1.5 and $4 million a year for a team of 6–9 people.
- Instead of listing all requirements, describe your problems and goals in a Statement of Objectives. Articulate user stories that deliver real value to a specific user.
- Make your deliverables working software. Reports can be useful, but what you're paying for is software code and configuration.
- Make sure that the data and code is in the government's possession and not something you'll have to pay to access later.
- Pay for needed work incrementally using a time and materials contract.
- Create real accountability by setting objective measures and tests. This Quality Assessment Surveillance Plan shows how you can monitor and review delivered code so that it's well written enough to function properly for the public that will be relying on it.

Here is an example of an RFP from the U.S. Tax Court that embodies this approach. This project successfully replaced a legacy system with a new one, Dawson, that has been lauded as a huge improvement.

What's available in the market

Commercial off the shelf software (COTS) helps you to avoid reinventing the wheel when you're implementing a software system. For products where there are large markets or clear

well-defined needs, like email, a provider can set up a service that you don't have to directly manage — you just administer your account. It's also usually cheaper and works well.

For many specialized government needs, there isn't a large enough market to create a plug-and-play solution. Software advertised as COTS often requires large investments in customization and maintenance, giving you the downsides of large, obscure systems that you don't control *and* the downsides of custom development. You want to avoid finding yourself in this situation, but it's not always easy to know before you buy it.

Based on our experience:

- Software demonstrations from sales teams are not a good indicator of success. It's much easier to build a software demo than real software.
- You should ask how well the product integrates with other services and ask for real evidence. We suggest building with loosely coupled parts. Look for no-cost access to data, clear data models, and already available APIs.
- You should reach out independently to other people who use a given tool or vendor (not just people who have been picked for recommendations from the vendor).
- An Enterprise Service Bus (ESB) is a common way to enable data integration with other systems. While we recommend relying on available APIs, if the COTS software includes one, make sure you plan for the long-term licensing costs.
- The labor costs to configure a commercial product should be less than the licensing cost. If you're paying more, you're developing custom software. Use the practices we describe to do it well.
- By treating your customizations and configurations as code, and owning the work done by your contractor, you can minimize the cost to change for your future. Open source your customizations, whether configuration or code.

Hiring

Projects with agency positions for Product Owner and Technical Lead are more likely to succeed and meet agency objectives. These roles help make technical decisions and manage vendors to achieve agency objectives.

A Product Owner is not a project manager, but instead defines the vision for the project and ensures that the vendor is continually delivering value for the agency and end users. The Technical Lead helps the vendor make technical decisions, guides the vendor's technical approach and ensures that the vendor is delivering high quality software.

You can fill these positions by teaching current staff new skills or by hiring.

- Example Position Description for Product Owner
- Example Position Description for Technical Lead