

# Procurement considerations for software development contracts

Many government agencies have a strong preference, sometimes even codified in regulation, for firm-fixed-price (FFP) contracts for IT projects. The current dogma is that FFP shifts the risk to the vendors for delivery. In reality, the risk is always with the government agency to deliver on behalf of its mission and users. Rather than trying to manage risk through contract type, we should always ask how the “thing” is priced and how the **government owners** can manage the contract scope to lead to a more successful delivery.

At General Services Administration (GSA), Technology Transformation Service (TTS) and 18F Office of Acquisition, we are committed to finding new ways to deliver working Information Technology (IT) systems along with our partner agencies. We believe that the [agile principles](#) of software development present a more modern, flexible approach that will lead to better outcomes. At a high level, these are the themes of agile:

- **Individuals and interactions** over **processes and tools**
- **Working software** over **comprehensive documentation**
- **Customer collaboration** over **contract negotiation**

- **Responding to change over following a plan**

Anyone with experience working with government IT contracts knows that these themes do not mesh well with the way that government typically interacts and contracts with outside vendors. However, many of us have participated in the status quo of “waterfall” contracts, as either buyers or technologists and have seen too many projects fail or overspend on software that ultimately doesn’t meet the needs of the mission.

So what does this mean when it comes to actual implementation of agile software development contracts?

### 1. Government buyers own the results, no matter what

There is a pervasive myth that when you contract out for services, especially with FFP contracts, you can shift performance and cost risk to the vendor. In reality, the government agency always assumes the risk if the vendor does not deliver. We’ve seen too many cases where the government buyer thinks they are protected against non-delivery by including tons of clauses and contract language about warranties and liabilities, only to be left holding the bag when things don’t work out. Even if the government is able to capture some costs through post-failure lawsuits, they ultimately won’t have a working product that delivers on behalf of their mission.

Our theory is that government should accept this situation and truly own the outcome of their contracts at all levels. Of course language should be included about what is expected of the vendor team, and oversight of the vendors’ performance should be the most critical activity of any cross-functional buying team. But the idea that a specific contract type, or a lengthy list of “shall” statements, will protect the government’s interest has anecdotally been proven over and over again as a faulty assumption.

This is why at TTS, we focus on “product owners” and “product management” and “product team” with our partner agencies, vs. spending week or months in “requirements-gathering sessions”. If the contract is formed in a way that establishes an overall vision and allows the product owner and vendor team to make adjustments to the implementation details as they learn, our partners put themselves in a much better position for success and a win-win for both our partner agencies and their vendor teams.

***Remember: “Customer Collaboration over Contract Negotiation”***

### 2. The contract type you choose must allow for flexibility

Responding to change is more important than following a plan. Don’t make the mistake of thinking that you or your vendors will know exactly what will be required prior to starting discovery and user research. Also - government product teams should assume that federal,

state, or local policies will change after a contract is awarded - often requiring the vendor to make tweaks to the product that is being built.

In software development contracts, FFP is counter to the understanding that the product owner and vendor team may need to adjust approaches and resource allocations after the contract is awarded. It's unreasonable to ask a government buyer to estimate the actual lifecycle costs before issuing a solicitation. It's just as unreasonable to ask a vendor to fix price something in a vacuum - your vendors will never know the true environment until they're able to "kick the tires" and learn new things in their bi-weekly sprints.

For this reason, we normally recommend Time-and-Materials (T&M) or Labor Hour (LH) contracts with partners that have a contract length any more than 2 months. These contract types can be established with a "maximum ceiling", and burn rates will need to be managed by the vendor's scrum team and monitored by the government product team. Having more flexible contract types allows for less "contract negotiation/change requests" than if it were issued as FFP. It also allows a contract to be cancelled/terminated without resorting to a claims process if (a) the vendor team is just not performing, or (b) the government team learns that the product will not provide the value they envisioned at the time of award, based on user feedback.

**Remember: *"Responding to change over following a plan"***

### 3. Shorter periods of performance on contracts

Far too frequently, the government issues multi-year contracts with a single vendor to develop a system or software - sometimes to avoid the "pain" involved with the procurement process. The unintended side effects of this include "vendor lock-in" and vendors that are not incentivized to deliver results early and often. Vendor lock-in means that the government buyer will be dependent on the single vendor to continue software development. In a worst case situation, the government will often lock-in with vendors that build closed-source, proprietary software to which only the vendor has intellectual property rights.

As we mentioned earlier, the government will "own" the results of anything that is delivered by a vendor. Building well-documented open source software should mitigate some of the switching costs that inevitably occur when transitioning from one vendor to another. Rather than a proprietary system, the government will be in control of the code repository and can share the existing code as part of future procurements. We believe there is more value in having shorter term contracts for specific modules of a system vs. outsourcing the entire system to a single vendor. This will also help reduce some of the "cost risk" associated with a T&M contract type. A six-month to year-long contract is much less risky than the typical 5-year contract where the exact specifications won't be known until the agile teams learn about the product and user needs.

This is especially true for our partner agencies that might be new to agile, modern software development contracts. We believe shorter contracts will reduce the risk to the government as its' product teams get used to agile methodologies. We also believe that shorter contracts for specific modules will lead to better relationships with vendors that are truly experts on building the type of software required for the specific module. The opposite is true for long-term contracts - these normally will go to more "professional services" or consulting type of firms that might have a broad array of experience within their company, but don't have the dedicated software talent that is required for small, agile vendor teams.

Finally, having a shorter term contract is a powerful forcing mechanism to ensure the vendor team is focused on delivering software that has value. There is more emphasis on delivery in a shorter contract with a well-defined vision than there would be in a long-term, open-ended "staff augmentation" type of contract.

**Remember: *"Responding to change over following a plan"***

4. Accept that a vendor won't provide a resource-loaded management plan at the time of proposal. Accept that you and your vendor won't be able to estimate the cost of every contingency up front.

Agile development teams assume there will be change as they learn more about the product needs and get feedback from the user community. Government buyers often have a difficult time estimating exactly how much a contract should cost in their "independent government cost estimates". There are just too many unknowns. Rather than pretend that a vendor will be able to provide a fully resource-loaded master schedule at the time of proposal, the government buyer should accept that their vendor may need to adjust their resources as they learn about the product needs.

To this end, we suggest procurement approaches where the vendor speaks to their development methodology and their technical capabilities versus submitting an exhaustive program management plan. The vendors should be able to provide an estimated level of effort and costs based on what is provided in the solicitation, but the government buyer needs to accept that the resources will shift as the vendor team interacts with the product owner and learns about user needs.

Note that this is much different than the waterfall approach most often found in government software contracts, and very different than a "Software Development Life Cycle" that have been pushed by many central governance shops. It's critical that the governance folks are aligned with the product owner and accept that everything won't be well-defined or estimated at the time of award.

It's also important to note that not having a detailed budget does not mean the contract is open-ended from a cost perspective. A skilled software development vendor will be

transparent about their actual costs and burn-down charts as they pick up steam and work with the government product owner. Owning the result means that the product owner and team are actively involved by monitoring and measuring at every step of the way to ensure that the vendor is delivering value. At the end of every two-week sprint the vendor should be able to report back on actual resources and the impact on the overall Not to Exceed budget. **Agile does not mean out of control.** In addition, having shorter periods of performance (see Point #3) will assist as a cost control function, even if the proposed costs can't be nailed down at the time of award.

**Remember: “Working software over comprehensive documentation”**

## Summing up:

We believe that the traditional ‘big bang’ approach to contracting sets us up failures that are hard to recover from, and make it extremely difficult to deliver the right product. We are recommending a different approach leads to better results in our experience, which requires thinking and acting in new ways. We believe that accepting ownership of technology rather than outsourcing it, working collaboratively rather than negotiating, thinking iteratively rather than in ‘big bangs’, and adapting to change rather than planning everything up front, will provide pieces of value much earlier in the process for everyone. When failures happen, they will be smaller, and because they are small, we can recover and learn. There will also be successes: incremental chunks of value delivered quickly and successively that can tell us we are moving towards the world we want to see. Agile contracting *practices* will help get us there, but the value of the approach can only be realized through active buy-in and engagement from both the buyer and the vendor in this new way of thinking and acting.