



Prerequisites for modular contracting

By [Waldo Jaquith](#)

Published on February 28, 2019

[acquisition services](#) [procurement](#)

At 18F, we often advocate for [modular contracting](#) — the practice of breaking up large, custom software procurements into a small constellation of smaller contracts. In doing so, we've learned that we're more effective when promoting that message to people who are familiar with the fundamentals of modern software development practices.

We've run dozens of workshops to teach our state and federal clients about modular contracting. Like any good agile organization, we constantly re-examine what we're teaching and why. In doing so, we've learned that modular procurement requires agile, product thinking, user-centered design, DevSecOps, and loosely-coupled architecture.

Follow 18F

- [18F on GitHub](#)
- [18F on Twitter](#)
- [18F on LinkedIn](#)
- [RSS feed](#)

Agile software development

It's not difficult to manage the contract for a software development team that is small, self-contained, self-motivated, self-improving, self-organizing, customer-focused, and building toward a shared set of objectives. [That's agile](#). For Contracting Officer Representatives or Contracting Officers who may be accustomed to adversarial relationships with software vendors under a traditional procurement, working with an agile software team is a very different reality.

By having a development team ship a functioning, documented, tested product every couple of weeks, it's obvious whether they're working effectively, and it's easy for a new development team to pick up where a prior one left off. By dividing a big project into smaller, stand-alone projects, each of those can be completed by a single agile team, and those teams can even be at different vendors under different contracts. By having somebody within the agency serve as the [product owner](#), and thus as the fulcrum for all work, key knowledge and decisions remain within the government, making it easier to end contracts that aren't performing, knowing that a replacement team can pick up where the prior one left off. It is crucial that agencies always be able to cancel a contract if the agile team isn't performing, or if the agency discovers what they envisioned at the time of the RFP doesn't serve user needs. Agile helps to ensure that the agency is always in a position to do that.

Product thinking

Users of software are trying to solve a problem, and [product thinking](#) is about identifying that problem, connecting it to a vision for the overall work, and the strategy for accomplishing that. That work, [product management](#), must be overseen by a single person. As with the product owner, this person needs to be an agency employee, rather than a contractor, so that this key knowledge stays with the agency, while vendors come and go. This is a different person than the product owner — the work being done here is higher-level.

User-centered design

Nobody knows the end-users of the software better than the agency, and [user-centered design](#) centers the experiences and needs of those people throughout the design process. The agency can remain in control by having the vendor build software that is centered on the user needs by employing a user research process that's coordinated with product management, so that the product owner can ensure that the vendor is producing high-value work.

DevSecOps

One vendor can use the infrastructure established by another vendor with absolutely minimal setup time, if they use standardized continuous integration and [continuous deployment practices](#), ideally with infrastructure-as-a-service (often referred to "IaaS") providers. After the initial vendor sets up a government-owned source code repository, continuous integration service, static and dynamic analysis service, and gated deployment to a cloud host, subsequent vendors can easily use that same pipeline with a few minutes of configuration work.

Components loosely coupled via APIs

The old model was to design monolithic systems (often referred to as legacy systems) containing complex interactions, and attempt to coordinate work between vendors collaborating on it. Now we have vendors build a series of smaller, standalone systems, with only one vendor working on any one of those systems. Each of those components communicates with the others via standardized, documented API methods, for example, [REST](#). (This approach is also known as building "[microservices](#).") This simplicity facilitates "emergent architecture," in which the structure of an overall application can be designed gradually, as it is needed, instead of being designed up front, which facilitates agile.

Each prerequisite matters

Collectively, these practices make modular procurement viable. If agile was omitted, it would be difficult for a new vendor to pick up the work of an old vendor. If DevSecOps were omitted, it wouldn't be plausible to deliver working product increments at the end of each sprint. If user-centered design were omitted, there'd be a real risk of delivering a system that wouldn't address user needs. And so on.

Modular procurement of custom products is often wise, and we'll continue to advocate for it and support our partner agencies in that goal, but it is part of a larger process that includes all of these prerequisites that are necessary for modular procurement to be viable.

Thanks to Alan Atlas, Mark Hopson, and Vicki McFadden for their work in identifying these prerequisites for modular procurement

[Previous post](#)

[Product management at 18F, part 4 - 10x](#)

[Next post](#)

[The U.S. Data Federation wants to make it easier to collect, combine, and exchange data across government](#)

An Agile Software Development Solicitation Guide

This is part 2 in a series about how to contract for agile software development services. In this post we'll show you how to create a complete solicitation in hours, not months, using our agile contract format

The value of cross-functional teams

A core concept of agile is that teams are cross-functional: the team collectively possesses all of the skills necessary to get the job done. We embrace that at 18F, and take it a little farther, and not just on agile teams.

Am I doing this right?: Antipatterns in agile contracting

As agencies try to adopt agile development practices and modular contracting methods, there are several anti-patterns that we have noticed through the course of our work. We address how these can hinder success and alternatives to consider.