An Agile Software Development Solicitation Guide

By Waldo Jaquith, Randy Hart, Mark Hopson, Vicki McFadden

Published on August 20, 2019

acquisition services | agile

This is Part II in a series about how to contract for agile software development services. See also: "Prerequisites for modular contracting".

Government solicitations to procure custom software are often long and complicated, include many pages of requirements, and can take months — even years — to write. But an agency can hire an agile software contractor with a solicitation that's only a dozen pages, written in an afternoon, using our agile contract format. It can be done under existing procurement regulations and within contracting officers' existing authority. The savings, in time and money, can be enormous.

Let's look at what goes into making a complete acquisition package using these methods. Beyond just the adapted format for the solicitation, it's our experience that there are a handful of additional elements required to make this work:

- **A Statement of Objectives — not a Statement of Work** — because you don't know exactly what needs to be done, so you can't possibly define it up front. The product owner, in collaboration with the developers, will determine on a sprint-by-sprint basis what work needs to be done. Anything else wouldn't be agile.
- **A labor-hour contract, not firm fixed price**. You're not buying a defined product, but instead buying developers' time. (Alternately, use a time-and-materials contract, as the contracting officer may prefer.) Historically, the default contract type has been firm fixed price, on the assumption that this reduces risk. However, if you are in a position to constantly measure ("inspect and accept") software quality, a time and materials contract — with a ceiling on total spending — allows for more flexibility for the software development team. A time and materials contract also allows for much easier escape clauses if the direction of the work changes or the contractor team is not producing quality software — if their work is inadequate, or their skills prove inappropriate, then no further work need be assigned to that contractor (effectively terminating the contract), and the contractor can be replaced. See our sample Determinations & Findings for a time and materials contract type.
- **A short base period of performance, usually between 6–12 months**. A couple of options to extend are fine, but keep them short, and never exceed a total contract length of three years. You're hiring a team to achieve a defined set of objectives and then leave.
- **A nominal appendix of the backlog of user stories.** This is how contractors can understand the specifics of the work that they're being asked to do, beyond whatever brief summary exists in the introduction. It's important that you make clear that the backlog has been included to illustrate the work that you presently believe will need to be done, but that it is *only* an illustration, and not a list of tasks that must be completed. It's a mistake to inventory user stories up front, rather than as part of performance, because there's no way to know up front what work will need to be done, as is inherent to agile. The language provided in the agile contract format allows for this flexibility, since the backlog allows for dynamic, evolving outcomes rather than a static, predefined list of needs.
- **Quality assurance and surveillance plan** (QASP) requiring that, at the end of each sprint, all code be complete, tested, accessible, deployed, documented, and secure. We publish a sample QASP that can be incorporated as-is.
- **Key personnel should include, at a minimum, the lead developer, and quite possibly the project manager** (a position that travels under various titles — whomever is responsible for unblocking things, following up on tasks, and serving as the contractor's interface to the client). You're buying people's time, and you want to make sure that you're getting the people who were advertised. Be wary of specifying too many key personnel, though — you're requiring contractors to put those people on the project if they get the contract, which can mean that they're functionally benched until the contract is awarded.
- **Allow for a distributed team**, whenever possible. We advocate for remote development, rather than requiring the contractor team to be onsite at your location. The best development resources are not in your city — they're spread out across your state and even the country. (There's a 150% difference in the salary of software developers between the most-expensive and least-expensive states in the United States.) Using a modern technology suite, collaboration is easy. The product owner should always know what the team is working on, and the team should be readily available to answer questions or huddle on any issue that arises. This is made possible with video calling for face-to-face collaboration (e.g. Google Meet, Zoom, etc.), instant messaging (e.g. Slack, Google Chat, etc.), task management (e.g. GitHub, Trello, etc.), and collaborative document editing (e.g. Google Docs, Office 365, etc.).
- **All work products must be published under an open source license** — or, better, committed to the public domain as a work of government. The contractor owns nothing.
- **All work will be committed to a public code repository** at least daily.
- **Technical proposals should be kept within page limitations**, preferably 2–3 pages, to minimize both contractor work and the time required for evaluation. Requests for more space may indicate that the contractor isn't taking an agile approach, or that they don't adequately understand what they're proposing.
- **The contractor must submit links to 2–3 source code repositories** where their illustrative past work can be seen. Allow this to include work done by key personnel (e.g., the technical lead) outside of their employment with the contractor, since many contractors will not have any public repositories to point to for lack of clients willing to work in the open. This is a far better indicator for how they are likely to perform under real-world conditions rather than the attempted simulation of coding challenges, hackathons, or other gimmicks.
- **Evaluation criteria should emphasize the contractor's proposed technical approach and their similar experience / past performance**. You're hiring a team based on their experience and their general approach to the work at hand, so there's little else to go on other than these two criteria. This also keeps the contractor's performance work statement brief and to the point.
- **Contractors' key personnel must participate in a verbal interview process.** This provides an opportunity to hear directly from the people who will be doing the work, rather than contractors' "capture managers," and usually proves a quick way of separating the wheat from the chaff.

All of the above can be incorporated into a solicitation in a dozen or so pages, not including appendices like standard administrative clauses, a description of the technical environment that the contractor will be working within, the product backlog, etc.

A solicitation like this can be put together in a few hours. In a standard, three-day 18F procurement workshop, we use only the latter half of the third day to lead our client through an exercise in which we put such a solicitation together. With the right training and coaching, it's not hard.

It's crucial to include the contracting officer in this process at every step of the way, starting as early as possible. When we host procurement workshops, we insist that the CO join the entire process, instead of just coming for the final day to hear about modular procurement and put together the solicitation. The CO needs to have the same base level of knowledge about the agile development process as everybody else in the room. Without knowing about agile software development, DevOps, etc., this prescribed contract format makes very little sense.

18F provides an agile contract template — it's a good starting point for anybody putting together a solicitation for custom software development.

It's a mistake to use hand-me-down government solicitations to procure agile software development services. By paring out everything unnecessary and focusing on procuring competent developers to achieve clear objectives, you can produce solicitations in hours, not months.

If you're interested in working with 18F, get in touch at inquiries18f@gsa.gov.

Follow 18F

18F on GitHub
18F on Twitter
18F on LinkedIn
RSS feed

‹ Previous post

Keeping your accounts secure

Next post ›

Building product management capacity in government part 1 – Our coaching philosophy

A federal guide to de-risk government technology projects

Announcing the federal field guide to de-risk government technology

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.