Agile Work Statements

Separate technical direction from contract work statements

Problem: Technical direction memorialized in a contract based on a rigid Statement of Work restrains flexibility when assumptions prove false

Recommendation: Separate technical direction from contract requirements and use a living roadmap adjusted to the product backlog and user feedback

Contract Includes:

- FAR, PoP, price
- Repeatable security, testing, and deployment processes
- Licenses and data rights
- Product vision, deliverables
- Meetings and reviews

Contract Excludes:

- Specified direction of technical implementation
- Prioritized list of features
- Long contract data requirements lists

Success: Contractors held accountable for delivery and integration through a disciplined process

Resources: GSA Agile Contract PWS Template; TechFAR Handbook; FAI Periodic Table; MIL-HDBK-245D

Context & Motivation

Government leaders have good reason to talk about the need for agility. The principles of agile have not only been adopted by the fastest growing startups in the commercial sector, but also by the largest incumbent firms including IBM, AT&T, Procter & Gamble, John Deere, and many others. Survival in commercial markets demands adaptation.

Though many definitions of agile exist, the Agile Manifesto provides four guiding principles:

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

The Agile Manifesto finds a companion concept in the idea of formal relational contracts. Nobel Prize winning economist Oliver Hart noted how strategic partnerships between organizations can be improved by focusing on desired outcomes and a process to manage the relationship.

For example, when Dell selected FedEx in 2005 for its hardware return-and-repair procedure, it drew up a 100+ page document filled with "supplier shall" statements. For nearly a decade, FedEx met the letter of the contract but neither side was happy. Just two years after switching to a formal relational contract, they were able to reduce costs by 42% and scrap by 67%. Both companies now consider the approach a best practice to be applied with all relationships.

In many ways, the heritage of defense contracting is colored with agile processes. Lockheed's Kelly Johnson liked to tell a story about the P-80, America's first jet. He got a letter contract to start work drafted, approved, and signed within 90 minutes. Similarly, in 1955 the entire specification for the F-4 Phantom II con-

tract fit within two pages. By contrast, in 1980 the C-17 specification consistent of 13,516 pages.

The long list of "supplier shall" statements that has pervaded government contracting works best for well-defined procurements. For innovative products such as software, the process gives only the illusion of control. When one thing is incentivized, another is disincentivized. This leads to the perplexing situation faced by Dell and FedEx where contract obligations are met but neither side is happy. This situation is also shared by many government programs.

When technical solutions are uncertain, it is wise to provide room for discretion in contractual relationships. As defense contracting scholar Frederic Scherer said in 1971, "given the kinds of technical problems characterizing modern-day weapons developments, inflexibility of contractual instruments is incompatible with economy."

In practice, this formal relational contract means removing much of the technical direction that presumes the product end-state from the contract work statement. In its place, there's a vision and a process. There is a lot more detail for crafting agile work statements including templates in the TechFAR Handbook, FAI Periodic Table, and GSA Agile Contract PWS Template. Essentially, the government buys a partner rather than a pre-specified product.

Collaboration. Once the contract provides for flexibility in technical direction, government product owners—often the contracting officer's representatives—must feel empowered to make decisions so long as it fits within the constraints set by the program. The agile contracting approach recognizes that priorities will change. Collaboration substitutes for extensive contract negotiations, and requires a mission command mindset throughout the organization. The product owner should consider the following collaborative tips:

- Face-to-face communication is best
- Trust the contractor for the first few sprints
- Do not reprioritize the product backlog in the middle of sprints
- Facilitate the contractor's access to end users
- Clearly identify how deliverables are validated

Accountability. Long work statements and thousands of scheduled tasks only give the appearance of accountability. Contractor schedules, however, often conceal rework and delays until after the sunk cost bias kicks in where government loses its leverage. By contrast, agile processes reduce customer risk by delivering functionality early. Delivered software is the primary measure of progress rather than percent complete or earned value. The product owner must thoroughly document artifacts from agile development and integrate them into the program requirements process.

> **"The backlog replaces the list of tons of requirements. And with iterative contracting, you throw out the IMS, give a capacity over a period of time, and prioritize based on product owner's vision and user feedback."**
>
> Tech Co-Founder

Deliverables. One fear of the agile approach is that funds will be sunk without a way of measuring progress to program requirements. Vaporware is the term for software development that doesn't leave behind user functionality. The contract should specify deliverables, such as sprint documentation and dates for capability releases. To ensure the right features get delivered, government product owner should help define acceptance criteria for items on the Little "r" requirements backlog. As the product reaches a stable baseline, performance or functional requirements can be added to the contract. This differs from an integrated master schedule because there is no critical path or baseline change requests that limits adaptation.

Course Correct. Rather than switch vendors at the first sign of trouble, product owners should talk with company managers to address any instances of personnel being unqualified or uncooperative before re-competing the effort. Similarly, product owners should be monitored by the product lead and the contracting officer to verify that they are qualified to carry the responsibility entrusted to them.

Scaling. Though many will admit that agile processes seem to work for small applications, they have doubts about whether it applies to major programs that will involve hundreds or thousands of people. However, Gall's Law states that all complex systems that work evolved from simpler systems that worked. Evidence for this statement is as pervasive in nature as it is in systems engineering. Project Hindsight in the 1960s, for example, found how each major weapon system required dozens if not hundreds of significant technologies be developed before the system could be made possible. If all these technologies were scheduled as part of a single development program, then efficiency is lost. The chance that all components will advance and integrate as planned is vanishingly small.

> **"A complex system that works is invariably found to have evolved from a simple system that worked. The inverse proposition also appears to be true: A complex system designed from scratch never works and cannot be made to work."**

Adoption. Take an agile approach to replacing legacy systems. The alternative to replacing one major system with another is to "boil the frog." According to the fable, if you put a frog in boiling water it will jump out, but if you slowly turn thestove heat up it will slowly cook to death. Similar- ly, rather than requiring that a new system meet all the requirements found in a legacy system, it can introduce the core functionality, solve niche problems that legacy systems cannot address, and use work-arounds to fill gaps. User adoption can then drive system transition and help steer the direction of continuous development cycles.

Legacy Program Example. The Navy's F/A-18 program office has been using an agile process for many years. The program office had nearly 700 organic technical folks who could take ownership of short one-page little "r" requirements. Technical direction was separated from the contract and the government leads worked closely with the contractor. This allowed the program office to issue task orders within one week. More than 200 separate requirements were being worked at any time, and they could be deployed as part of regular capability releases. While this sped up fielding and lowered costs, it required a critical mass of organic technical capability and a culture to support it.

Wrap Up. There is empirical evidence that agile development processes work. Yet these pro- cesses often fail in organizations when the business functions remain fixed to industrial era methods. The failed situation is sometimes called "water-agile-fall." For contracting to support agile software development, work statements should invite a collaborative environment that ensures the Government has sufficient insight and participation in design choices. The con-tract remains legally enforceable, and account-ability is improved through early releases of functioning product.

Major Elements of a Performance Work Statement for Agile Contracts

Vision statement. Provide the high-level scope and business motivation.

Agile processes. Identify the core construct of the desired process:

- Sprint duration. Allow the contractor to define a sprint duration, but should be short (two to four weeks).
- Product Backlog. A list of features or user stories to be developed as well as defects to be fixed. User stories often come in the short format "As a ___, I want ___ so that ___." Each item will be assigned an effort estimated by the development team and a business value estimate by the product owner. A solicitation may ask for an initial product backlog, but this isn't to be contractually binding.

Definition of Done. During each sprint planning meeting, the parties will agree to the conditions of acceptance testing for each item on the backlog. Tests should be conducted and passed, code has been reviewed, standards have been met, and documentation has been completed. It defines whether the product is shippable.

Business. Regular business items like type of contract, price, and period of performance. Additional items may include: cybersecurity requirements, intellectual property, terminations, and technology standards.

Roles and responsibilities. Besides the contracting officer and procurement manager's duties, identify at least three additional roles and their responsibilities in agile "ceremonies":

- Product Owner. This is generally the Contracting Officer's Representative, who communicates the customer vision, takes the lead of prioritizing the backlog, and participates in sprint planning/review activities.
- Development Team. A cross-functional team which performs on each sprint. Contractor should provide key personnel, skill types, and hourly rate.
- Scrum Master. A contractor who ensures cooperation between the Product Owner and Development Team, but is not the project manager. Attends sprint planning/review activities.*

Deliverables. Agile contract deliverables should not require specific features sets, but instead:

- Product backlog at the beginning of each sprint
- Reports at the end of each sprint on design files, product demos, performance metrics, etc.
- Development prototypes when required, at the end of a sprint or task order
- Code repository of the source code that corresponds to government rights

For more resources, see the Agile Contracts primer, Contracts for Agile Software Development, and the book Agile Contracts.