Lessons learned from previous engagements

18F

Our principles

(or what we try to do on every project)

18F

1. Break massive contracts into smaller modules

1. Break massive contracts into smaller modules
2. **Bring product strategy in-house**

1. Break massive contracts into smaller modules
2. Bring product strategy in-house
3. **Untangle the product from the legacy system**

1. Break massive contracts into smaller modules
2. Bring product strategy in-house
3. Untangle the product from the legacy system
4. **Organize around end-user value, not code**

1. Break massive contracts into smaller modules
2. Bring product strategy in-house
3. Untangle the product from the legacy system
4. Organize around end-user value, not code
5. **Work in an iterative way**

1. Break massive contracts into smaller modules
2. Bring product strategy in-house
3. Untangle the product from the legacy system
4. Organize around end-user value, not code
5. Work in an iterative way
6. **Take advantage of existing solutions**

What we’ve learned

We need you!
(and everyone else too)

18F

We know that this approach works. We also know that it’s hard. And it doesn’t work if part of the team is dragging their heels.

Ultimately, success depends on us everyone together. Half the group can’t do change at the other half.

Leadership needs different success metrics

18F

We’re trying a new way of doing things.

We're trying a new way of doing things. That means we need to try a new way of judging progress, too.

Here’s what we’re looking for:

+ backlogs get burned through
+ teams ID risks, and tackle them
+ new information affects strategy
+ we keep getting better at all of this

We won't be good at all of these things immediately — we're not used to doing them! The important thing is how quickly we **learn**.

+ new information affects strategy
+ we keep getting better at all of this

Drafting good procurement solicitations is an art

18F

We can’t just break big contracts into small pieces and send them out the door — our procurement solicitations have to reflect the approach we’re trying.

We can't just break big contracts into small pieces and send them out the door — our procurement solicitations have to reflect the approach we're trying. **Doing that helps vendors get excited about this approach, and staff appropriately.**

This approach doesn’t click for every vendor

Some vendors are really excited to work in this way.

Some vendors are really excited to work in this way. **Others… aren’t.**

Some vendors are really excited to work in this way. Others… aren’t. We need the whole team on board, and our vendors are part of that team.

It’s easier to look agile than **be** agile

18F

When you're judging vendor partners (and yourself), buzzwords like "sprint entry" and "velocity" don't mean much.

When you're judging vendor partners (and yourself), buzzwords like "sprint entry" and "velocity" don't mean much. **Instead, look for the team to learn quickly and change direction based on new information.**

A team that “runs waterfall” but finds ways to adapt continuously to new information is better than a team that “runs agile” but never changes their plan.

Agile execution is not a replacement for strategic planning

18F

Agile is a (really effective) way to get to a goal. You still have to do all of the work of setting a good goal.

We have to get serious about product management

18F

Product management is the glue that binds delivery to the overarching strategy.

Product management is the glue that binds delivery to the overarching strategy. **It's valuable everywhere, and absolutely critical to this way of working.**

Experience has taught us that there are great future product managers all over your org.

Experience has taught us that there are great future product managers all over your org. We have to give them space to learn a new role and empower them to be effective in that new role.

Data engineering is critical

Common software development patterns are aimed at systems where data loss is annoying, rather than devastating.

Common software development patterns are aimed at systems where data loss is annoying, rather than devastating. **We can’t afford that, so we need to take a data engineering approach to our systems.**

DevOps can make or break projects

18F

A stable DevOps platform is the basis of all meaningful iterative development.

A stable DevOps platform is the basis of all meaningful iterative development. We want parity between development, integration, and production environments, and deployment should be as automated as possible.

DevOps debt is particularly painful to pay down during the development process.

DevOps debt is particularly painful to pay down during the development process. **We need to build out (and test) our DevOps infrastructure before we start developing in earnest.**

Being transparent isn’t the same as being understood

Being transparent is great!

Being transparent is great! But making your process visible to others doesn’t mean that they can understand what you’re doing.

You need to create a clear, repeatable narrative about your project, and use it to contextualize all of your work.

You need to create a clear, repeatable narrative about your project, and use it to contextualize all of your work. Otherwise, people start to make assumptions about your work — that's annoying at best and can jeopardize the project at worst.