Software development, procurement, & management fundamentals

Agile management

Part 1 of 5

Presented by 18F for:
Office of Childcare, HHS

August, 2022

Software development, procurement, & management fundamentals series

1. Agile management
2. Product ownership
3. User-centered design
4. Software development practices
5. Agile Contracting

What is 18F?

18F is a technology and design consultancy for the U.S. Government, inside the government.

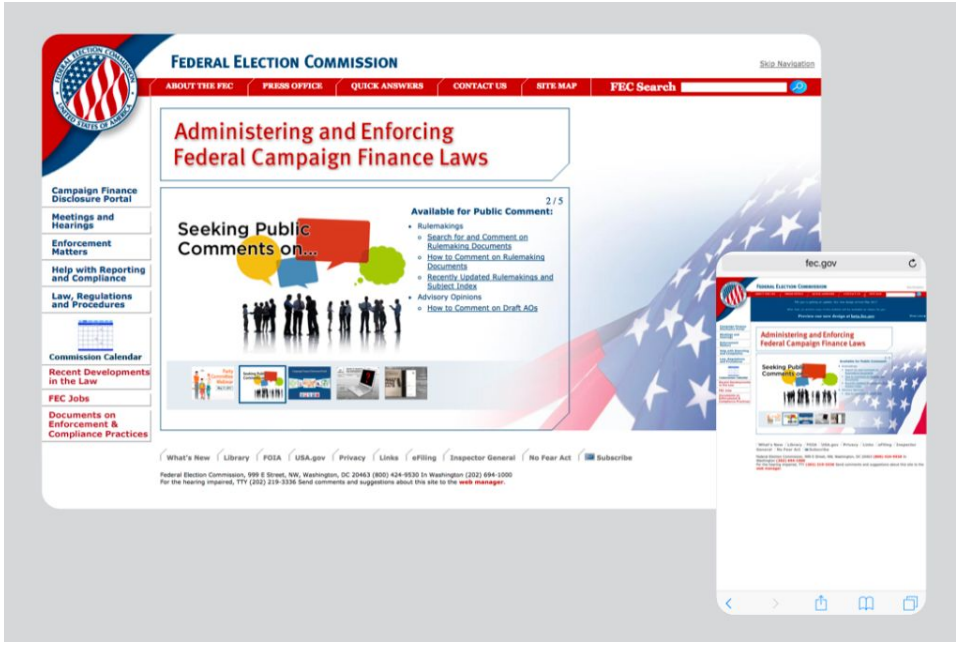
We share the same motivations as you: delivering great service to the public.

Lindsay Young

Experience

- Director of cloud.gov
- Civil Rights Division USDOJ, 18F
- Director of Cloud Adoption, USDA and HUD Centers of Excellence
- Director of Agency Partnerships, 18F
- Chief of Staff, GSA IT (CIO)
- Federal Election Commission Project, 18F

Federal Election Commission



FEDERAL ELECTION COMMISSION

Skip Navigation

ABOUT THE FEC | PRESS OFFICE | QUICK ANSWERS | CONTACT US | SITE MAP | FEC Search

Administering and Enforcing
Federal Campaign Finance Laws

Campaign Finance Disclosure Portal
Meetings and Hearings
Enforcement Matters
Help with Reporting and Compliance
Law, Regulations and Procedures
Commission Calendar
Recent Developments in the Law
FEC Jobs
Documents on Enforcement & Compliance Practices

Seeking Public Comments on...

2 / 5

Available for Public Comment:

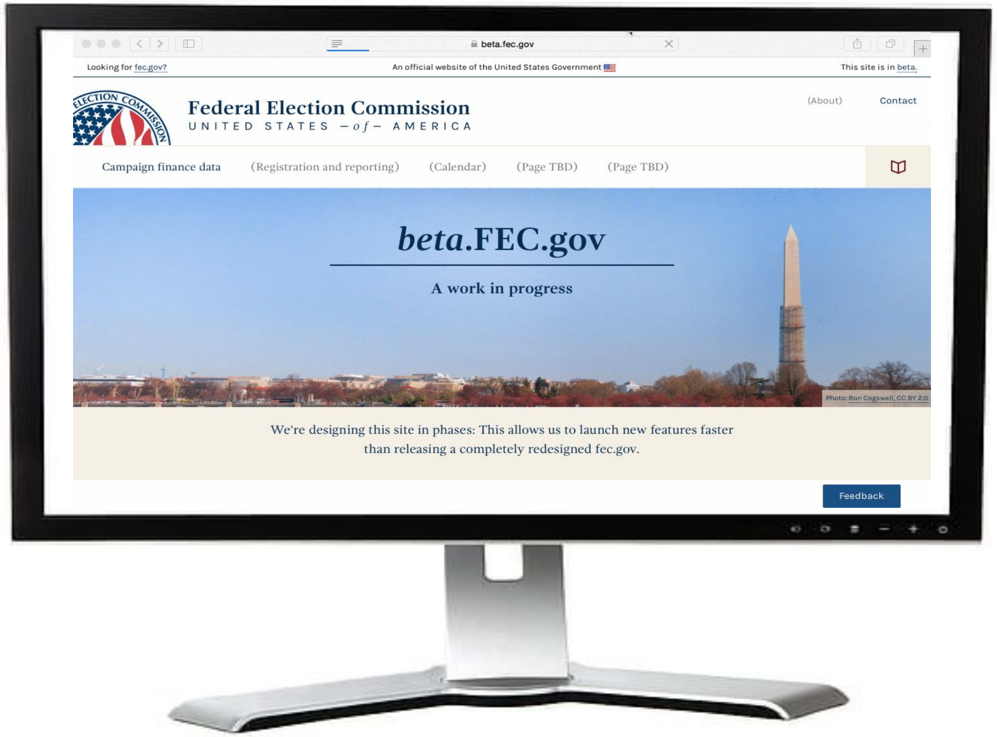
- Rulemakings
  - Search for and Comment on Rulemaking Documents
  - How to Comment on Rulemaking Documents
  - Recently Updated Rulemakings and Subject Index
- Advisory Opinions
  - How to Comment on Draft AOs

What's New | Library | FOIA | USA.gov | Privacy | Links | eFiling | Inspector General | No Fear Act | Subscribe

Federal Election Commission, 999 E Street, NW, Washington, DC 20463 (800) 424-9530 In Washington (202) 694-1000
For the hearing impaired, TTY (202) 219-3336 Send comments and suggestions about this site to the web manager.

fec.gov

Federal Election Commission



beta.fec.gov

Looking for fec.gov? An official website of the United States Government This site is in beta.

Federal Election Commission
UNITED STATES —of— AMERICA

(About) Contact

Campaign finance data (Registration and reporting) (Calendar) (Page TBD) (Page TBD)

beta.FEC.gov

A work in progress

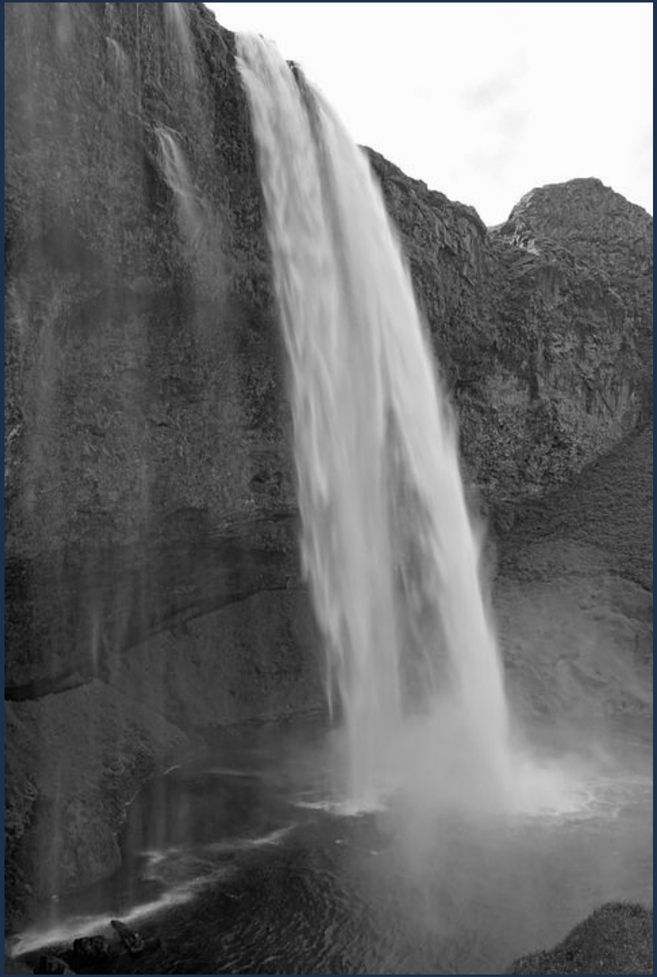
Photo: Ron Cogswell, CC BY 2.0

We're designing this site in phases: This allows us to launch new features faster than releasing a completely redesigned fec.gov.

Feedback

Agenda

1. Intro
2. Bad news/ Good news
3. Opportunities
4. What to prioritize
5. Am I doing it "right?"
6. Putting the pieces together



2/ Bad news/ Good news

Drawbacks of traditional practices

- Long planning processes
- Impossible to anticipate all needs
- Documentation becomes a stand in for progress
- Stakeholder driven decisions not informed by user research
- Get what you asked for — not what you need *now*.
- Adjustments are hard to make

The bad news: This is hard work

- Thank you for taking on this important work to support child care
- We understand that many of you are may be under-resourced
- You may be in a situation where you are bound by previous decisions
- There are no silver bullets

The good news

- You can make things better
- There are best practices that can help
- Small, frequent changes add up over time
- You are not alone on this journey

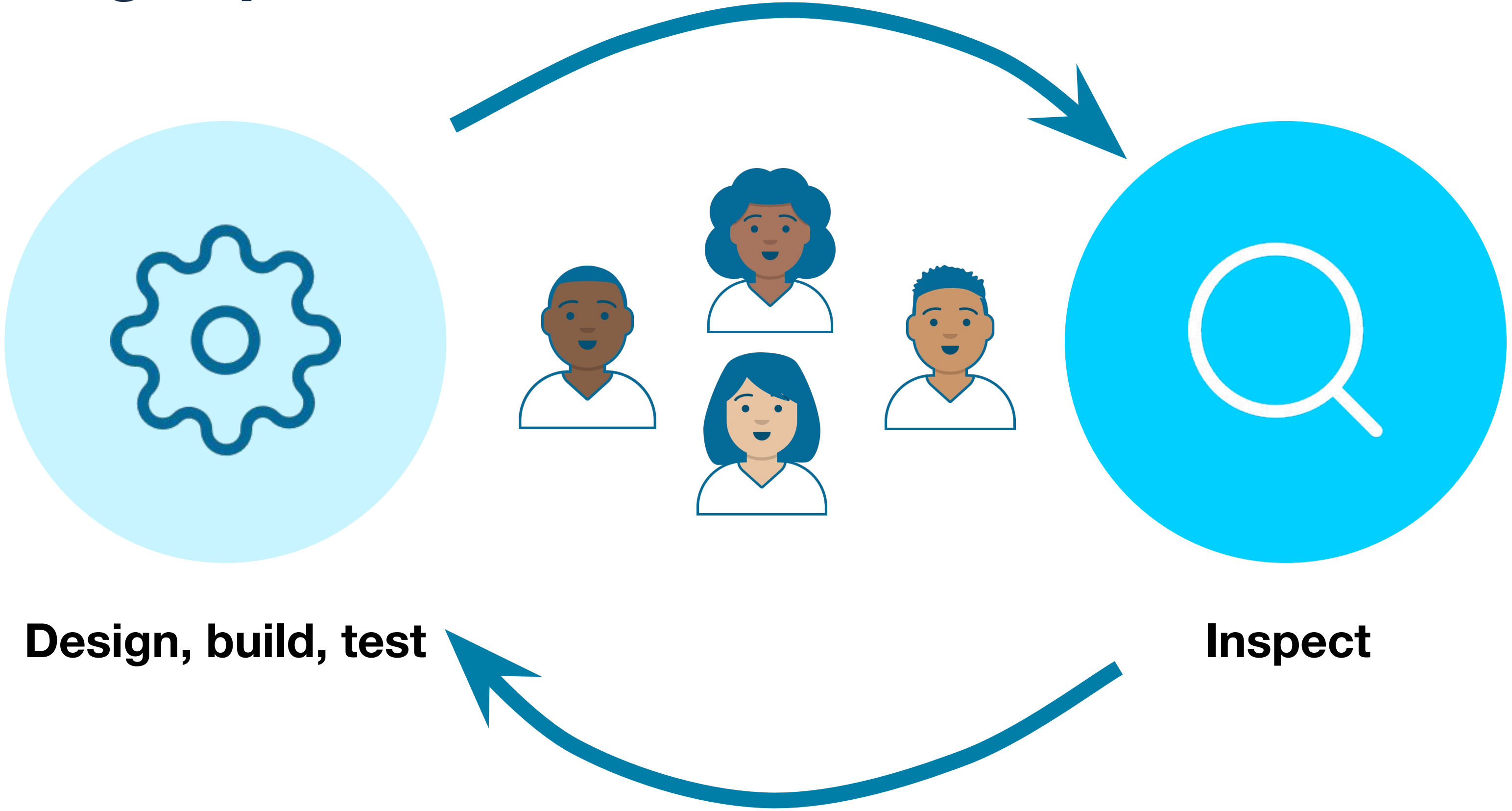
Best practices research

Only 13% of large government IT projects succeed

Agile projects are 1 ½ times more likely to to succeed than waterfall projects

Agile projects are 25% more productive

The agile process



Example

No planning

- Should we rent a car or a boat?
- Don't know where you are going
- You don't go anywhere

Waterfall

- Fully plan before you start
- Decide and commit to every detail
- You can't change
- When something goes wrong, the whole trip fails

Agile

- Planning in increments
- Accepting inability to perfectly predict the future.
- End result in mind
- Get into the details as needed
- You make changes as needed
- You can even choose to upgrade your goal

Agile activities

Activities	Examples
Research	Usability research & design research
Plan	Road mapping & sprint planning
Make something	Prototype, build, & create service or policy
Learn from mistakes	Retrospective & post mortems
Keep trying	Keep feedback loops small & keep repeating these processes

Use observable, working outcomes to measure success

Common agile methods

Scrum

- Breaking up into two-week increments called sprints

Kanban

- Caps the number of tasks in flight
- More flexible

When beginning agile

- Start with structure
- Start small, with pilots
- Transitions take time
- Support a collaborative culture

3/ Opportunities

Do what you can when you can

- Starting a new project, feature or initiative
- Building agile teams (contracting/ staffing)
- Managing agile projects

Starting out:
Solve the right problems

- Small pieces
- Is there an easier way to solve this problem?
- Do we need to build anything at all?
- What can we do now that adds the most value to our users?

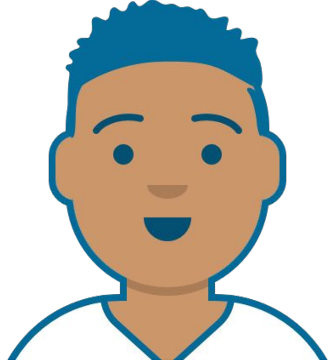
Building agile teams: Roles

Building agile teams: Product owner

- Empowered subject matter expert to make decisions and guide the product
- This is really a full time commitment. This should be an employee.
- The product owner doesn't need to be technical, they need to understand the user needs and create a vision for the product
- A product vision isn't a feature popularity contest, it describes the overall flow and purpose of the product

(We will go in depth with this in the Product Owner talk)

Building agile teams: Project manager/ SCRUM master



- This person will be your master of ceremonies and track work from user need to production
- If this is your first agile project, try to insist on someone with experience and not just a certification
- Once you have the rhythms of agile established, the whole team can help a beginner come up to speed

Building agile teams: Technical lead

- This is not a traditional Architect, because we want more involvement from the entire team to solve problems
- Communication is key to the success of the role
- This person helps advise on solutions and makes sure quality standards are being met
- Hire a tech lead if you can, better for accountability and oversight aspects of the role
- It is hard to hire for technical talent if that is not your area of expertise, but it is worth it

(We will go in depth in the software development practices talk)

Building agile teams: Designer

- There are many types of design; if you only get to choose one speciality, user experience is the place to start
- If you have the opportunity, products benefit from visual designers and content designers
- If you don't have designers, you will still have to make design choices that impact the ease of use and success of your product; whoever is making those choices will benefit from additional design knowledge

(We will go in depth with this topic in the user-centered design talk)

Building agile teams: Additional contributors

- Roles vary based on the project needs
- Engineers, writers, lawyers and other contributors to your product
- Will probably be a blend of staff and contractors

Building agile teams: Other considerations

- Try to keep total team size in the 3-9 range. If you have more than that, split up into sub teams
- Meetings take up a lot of time so you will only get ¼ of the productivity for ½ staff. Make sure you plan accordingly

Managing agile projects

The most important things about agile are:

- You frequently test what you build with people that will eventually use the product or service
- You change your plans based on what you learn from your testing
- You build things as early and as small as possible so you don't get too far ahead of your user feedback

Managing agile projects

The value of using agile & modern software practices:

- Stakeholders, collaborators, and implementers agree on measurable outcomes
- More information and context for decisions when they are getting made
- We can make course corrections quickly, with fewer bureaucratic bottlenecks

Managing agile projects

Cross-functional teams

- Collaboration across discipline
- Use your team’s whole brain to solve
- Include your security team from the beginning

4/ What to prioritize

If you don't know where to start

- Improve processes
- Clarify language
- Focus on customer experience
- Remove heavy weight control processes
- Define outcomes and hold the project accountable

Process

- You can't make a system that is better than its underlying workflow
- Don't reenforce bad processes with a shiny new system
- Try to use any technical project as an excuse to improve your processes

Improving processes

- Look for bottlenecks, redundant approvals and places where it is not clear to users what the next step should be
- Work together with all your stakeholders to streamline and simplify your processes

Language

- Use plain language that is more comfortable for your end user
- Remove or archive content that is old or not needed
- Work with content designers if you have the opportunity
- Can you support additional languages to reach a wider audience?

Prioritize the experience of the people using the software

- Work on small improvements with a big impact first
- Teach stakeholders to make decisions based on testing, not executive input
- If you are the executive, keep the conversation focused on user needs and let your product owner make decisions that get you to an agreed outcome

Reinvent or remove change control boards

- Replace change control boards with regular communication with decision makers
- Use a weekly ship model
- Organizations with heavyweight change processes are **2.6 times** more likely to be low performers*

Accountability to outcomes

- Define the outcome you hope to achieve
- Agree on metrics that measure that outcome
- Have an enforceable quality assurance plan with clear standards
- When things go wrong, use it as an opportunity to learn

5/ Am I doing it right?

Good signs

- The whole team collaborates on solutions and implementation ideas
- You are adding to or improving your product in production at least every two weeks
- You catch mistakes early and fix them quickly
- You changed your plans based on usability testing

Bad signs

- One person making all the decisions
- Many inflexible deadlines
- Tasks take more than two weeks
- Difficulty making changes
- Large backlog of bug reports
- A flood of with change requests and upcharges
- It feels the same

One person making all the decisions

- Need more work on creating a collaborative culture
- People need to know that it's safe to contribute
- People need to know their opinion is valued

Many inflexible deadlines

- You need to transition away from waterfall and replace these plans with a project roadmap

Tasks take more than two weeks

- Spend more time breaking tasks into smaller pieces
- Make sure your tasks are well defined
- Don't spend months of research in the beginning of the project

It is hard to make a change (or roll one back)

- You need more automation and more DevOps practices

Large backlog of bug reports

- Make sure you are prioritizing fundamentals before feature requests
- You may be understaffed
- Worst case scenario—you need to replace your system

Flooded with change requests and upcharges

- You need to change your contract structure to allow for agile methods

It feels the same

- Replace plans with a project roadmap
- Focus on how to make meetings more
- Consider how you feel

Putting the pieces together

Putting the pieces together

To get the full benefits of agile management:

- Solid product ownership & vision
- User-centered design practices
- Modern software development practices & technical oversight
- Agile contracting

Software development, procurement, & management fundamentals series

1 Agile management

2 Product ownership

3 User-centered design

4 Software development practices

5 Agile Contracting

Agile Management

Lindsay Young
18F