Introduction
Modern software product development
Agile is a thing already
Agile is something you are, not something you do
18F Agile based project approach
Agile fundamentals
Agile lexicon

18F Agile based project approach

The Agile Manifesto is realized at 18F in the combined practices of iterative software development, product management, user-centered design, and DevOps.

We start with a product vision and strategy, informed by users and the overall mission of 18F or one of our partner agencies. We do this so that the work always stays connected to an overarching goal that everyone understands and is excited about.

We also work to ensure that the infrastructure and process is there to enable continuous delivery of software to real users (DevOps), and that a clear agile delivery process is set up. Teams are free to tailor their agile process to suit their own situations. Many of them employ the SAFe methodology, while other teams use more traditional Scrum or Kanban methods.

We conduct discovery research before we build anything. Depending on the complexity of your problem space, this can take up to 2 to 3 months. As opposed to ‘requirements gathering’, this process involves actually visiting with users and prototyping to test out multiple concepts quickly before investing a lot of money in building something.

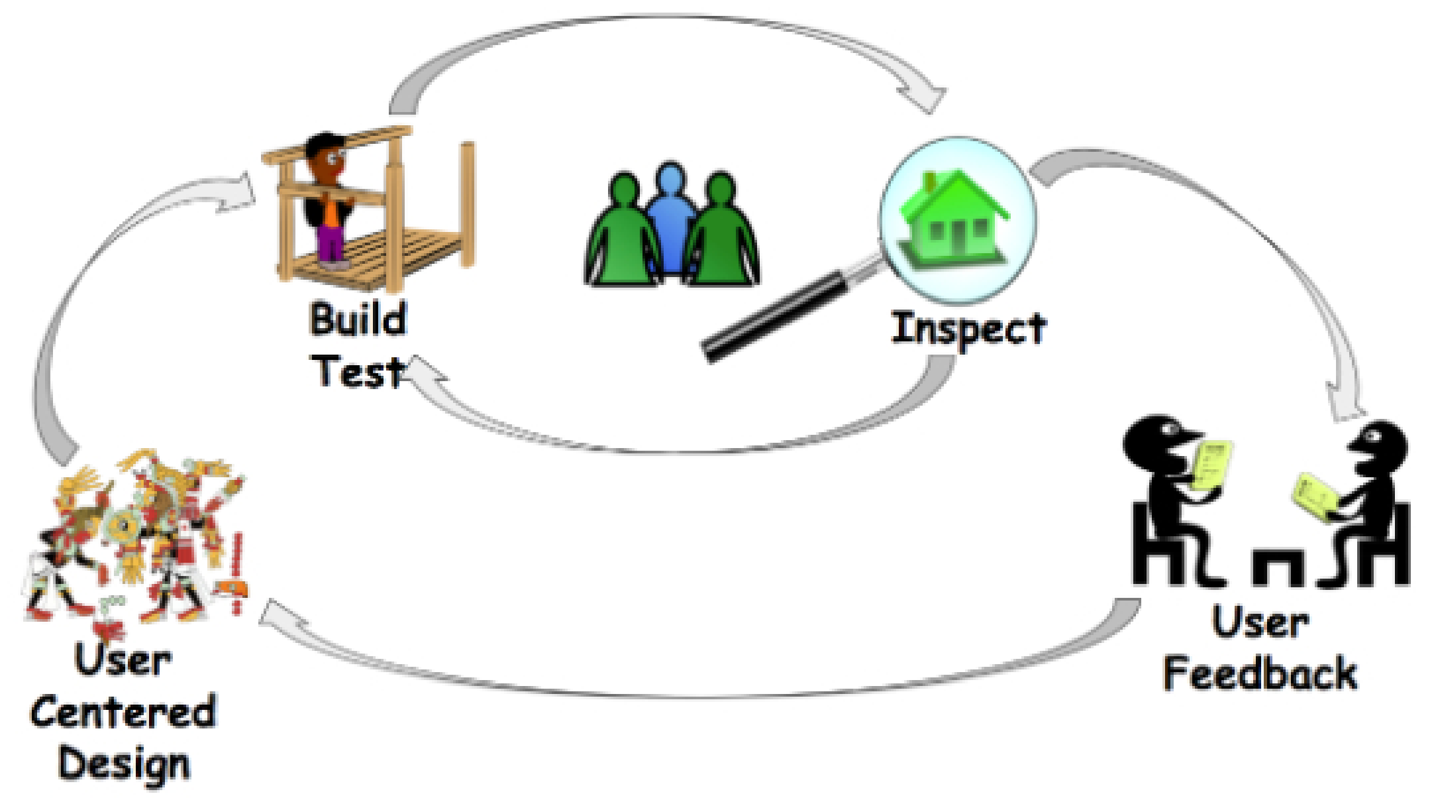
When we build, we aim to release early and often to real users in real life situations. Ultimately, the government’s investment should be measured in working software, not phase documents or milestones. Only working systems are of value to real constituents. Based on that premise, teams are only able to measure the success of a waterfall project after most costs have been incurred, because that is the first time working software is delivered. This is a huge risk. Agile projects allow the government to measure success at more regular intervals and if necessary make course corrections. The ability to measure and adapt reduces the overall risk to the project.

Not all of our projects are greenfield; sometimes we work on legacy systems. Incremental development is important in these situtations as well. Incremental development can allow end users to start seeing benefits even before the legacy system can be fully replaced. If a workflow is worth preserving, then portions of the old systems must be kept up until the new system is fully operational, no matter what approach is used. In a waterfall approach, the end user must wait many months or years to use the new system and see any benefit — and that’s assuming there are no problems with the rollout. With an agile approach, the end user could start seeing benefits in a shorter timeframe. Depending on the architecture of the legacy system, parts of it may even be able to be decommissioned and turned off before the full new system is in place to save money.

Having well-researched hypotheses beforehand allows us to be deliberate about what we build and why. Research continues throughout the agile process that we can test our hypotheses and pivot when needed. Since our work is centered around user research and user feedback, it’s important to develop participant recruiting strategies, and a research and feedback cadence that will work within the constraints of your agile process, and to ensure that the team is staffed appropriately.

A common pitfall is expecting agile to be a silver bullet to all that ails software development, and to expect agile to eliminate all project risks. We've seen teams try agile and, when the practices fail to eliminate all the challenges, write off agile entirely, claiming "agile doesn't work." In reality, agile does not eliminate risk completely; it provides techniques to manage risk more effectively than traditional waterfall processes. Unlike waterfall, agile accepts that the unknowns will lead to change. Agile treats change as an integral part of the process, as opposed to exceptions that need to be resolved via change control mechanisms. This enables agile teams to manage risk by allowing change to drive course corrections. However, some agile adoptions focus on the ceremonies as opposed to the intent. Incomplete, incorrect, or uninformed adoption of agile techniques can lead to an ineffective process we call “agilefall.” 18F has written a blog post about how to avoid this pitfall. Effective agile adoption will enable an organization to be nimble and respond effectively to the inevitable change that arises during software development.

Agile Development Integrates with User-Centered Design



This site is maintained by 18F.
Read our linking policy.
Have questions or want to contact us? Email us at 18F@gsa.gov.

GSA
18F Agile
An official website of the GSA's Technology Transformation Services

About GSA
FOIA requests
Office of the Inspector General
Privacy policy
Accessibility support
No FEAR Act data
Performance reports

Looking for U.S. government information and services? Visit USA.gov

Was this page helpful?