Government Innovators Network

A forum for innovation in the public sector

Blog
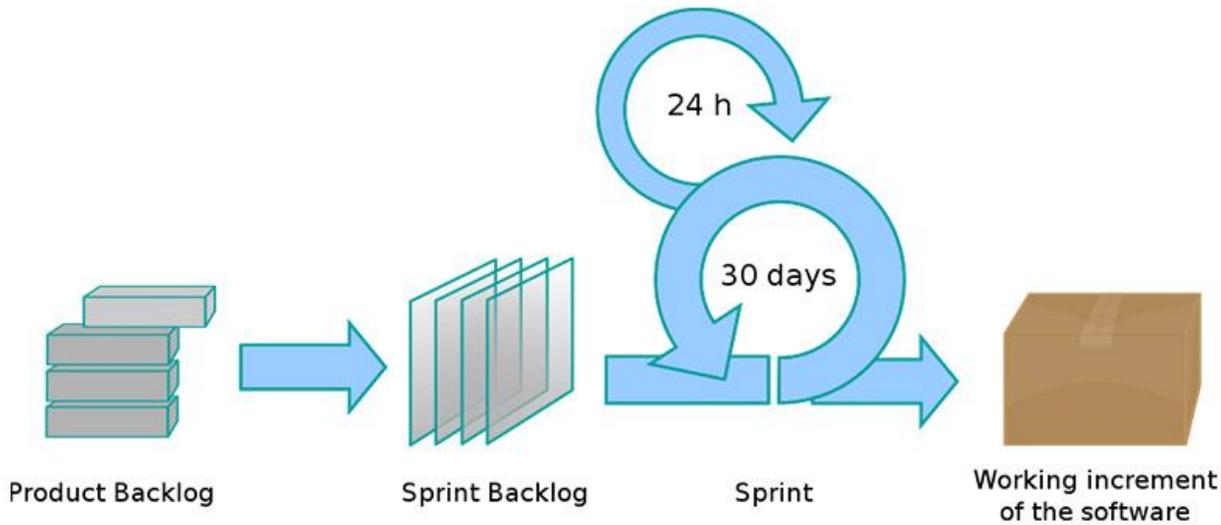
OCTOBER 5, 2018

The Path to "Agile" Policymaking

By Arjun Bisen (/blog/author/111606)

💬 0 Comments   👍 58 Likes (/blog/path-agile-policymaking?rate=Ya7H2lCsi6wvASDl2tf9Ba4j4LZlNFeqpagtYnoYpYw)

To meet the expectations of citizens, allocate resources, and encourage accountability, governments are structured to plan projects in advance in detailed policy cycles that respect hierarchies, annual budgets, and approval and review processes. But changeable and fast-moving issues, which are increasingly common in the digital age, demand more flexibility and responsiveness from government as it crafts and implements policy (https://www.theglobeandmail.com/report-on-business/government-businesses-not-keeping-up-with-speed-of-technology-report/article36664704/). By borrowing and adapting the "Agile (https://www.cio.com/article/3156998/agile-development/agile-project-management-a-beginners-guide.html)" project management methodology from the dynamic software and start-up worlds, policymakers can thrive in an uncertain environment, increase civic engagement, and improve lives.

What Is Agile?

Agile methodology involves user-centered design, cross-functional teams, prototyping, rapid iteration, and continuous feedback loops.

Agile was created to maximize a team's chances of delivering a reliable product that meets the user's needs, while minimizing the risk of failure due to environmental changes and limited testing. In an Agile process, multidisciplinary teams break down large products or projects into smaller, discrete features and deliver these features one-by-one while testing for functionality and usability throughout the project.

In a government context, instead of working on each individual stage of a program or policy in isolation — for example, planning, design, implementation, or testing/review, teams work to create successively improved versions of a "Minimum Viable Product" (MVP) that can be tested directly with users. In a traditional linear development cycle, testing is done only with a comprehensive final product — heightening the potential risk of failure if user demands are poorly understood along the way.

Agile project management emerged from the 2001 Manifesto for Agile Software Development (http://agilemanifesto.org/), which listed 12 principles for software development. I have summarized the principles and how they can be applied to public policy below:

AGILE PRINCIPLES	APPLICATION TO POLICYMAKING
Focus on customers over contract negotiations	Policies should focus on the end users/stakeholders. To do this, policymakers need to understand user behavior in detail and design policies to address their needs. Policymakers should not delegate this step to implementing agencies.
Prioritize working software before documentation	This refers to the art of maximizing the work *not* done. Focus on delivering viable policy products to test and iterate upon over documentation and communication. While important, excessive documentation can limit the ability of teams to adapt to change.
Encourage individual interactions, not just process	Rapid and regular informal communication is more important than formal bureaucratic communication. Verbal communication that is early and frequent will help address barriers and challenges as soon as possible without the need for lengthy memos and documentation.
Plan for change instead of following a plan	Expect, welcome, and adapt to change. Design plans and policies expecting your assumptions to prove false at some point. Create rapid feedback mechanisms to better understand the change underway. Do not wait for end-of-program evaluations to seek feedback on policy performance.

How Do You Do It?

Step 1: Form multidisciplinary teams that include not only policymakers but also policy implementers.

Step 2: Define the various 'users' of the policy, and brainstorm the behaviors and characteristics that divide them into useful personas. Conduct user interviews and/or observe user behaviors when they interact with the policy or program to understand user experiences and points of friction. Users are not only the agencies who implement the policy but also end users who are affected by the policy (e.g., businesses and citizens). Focus on *behaviors* and *needs* — not only on what users they say they want. Journey mapping (https://www.atlassian.com/team-playbook/plays/journey-mapping) is a useful exercise for understanding experiences with the policy. Tech companies often hire a Product Manager (https://www.atlassian.com/agile/product-management) whose primary job is to advocate for the user's needs in the organization.

Step 3: Analyze the problem as in any policy process (http://www.atlas101.ca/pm/concepts/bardachs-eightfold-path-to-more-effective-problem-solving/). Break the issue into policy features the users would be interested in, and that address specific "user stories (https://www.atlassian.com/agile/project-management/user-stories)" or needs.

Step 4: Break down the policy features into parts to form a "sprint (https://www.sinnaps.com/en/project-management-blog/agile-project-management-sprint-methodology)," a burst of work that produces an MVP. Meet frequently as a team (often referred to as daily "scrums (http://scrummethodology.com/)") to quickly discuss progress, next steps, and any internal blockers or challenges. Assign a scrum master (meeting lead) to ensure those challenges are addressed.

Step 5: Prototype and test the policy MVP with key users and stakeholders. You can test multiple policy ideas at once to understand the strengths and weaknesses of each. Testing does not need to be excessively scientific and rigorous; the main goal is to test your assumptions in the real world. Get the policy in front of the users who matter most and understand its impact on their operations and behaviors.

Step 6: Consolidate learning into a final product, while making time to reflect on and refine your process through a sprint "retrospective (https://www.atlassian.com/team-playbook/plays/retrospective)." One simple format is to ask each team member what the team should start, stop, and continue doing in the next sprint.

Step 7: Compile successful policy features into the initial product for release. Include features in the product that can elicit regular feedback from users. Continue to test and adjust the policy, or its implementation, even after launch. The learning should never end.

When Agile Works Best

Agile is a powerful methodology for development, but it is most successful in policymaking when teams know they have:

- Clarity on policy objectives
- Access to users and cross-sectoral teams
- Stakeholder tolerance for uncertainty
- Flexible timelines and sequencing
- No clearly-defined final product
- Tolerance for public engagement and failure

Barriers to Agile in Policymaking

There are many circumstances when Agile will have limited use in policymaking. In some cases, stakeholders will have limited tolerance for uncertainty, making Agile inappropriate once the policy has been released. Some stakeholders who could suffer in the case of volatility might demand stability; in such cases, Agile is only appropriate during policy design.

Government is also structured to operate with different agencies responsible for discrete portfolios, often with clear delineation between policy and implementation units. These divisions limit opportunities for creating cross-functional teams of policy experts, implementers, analysts, and stakeholders.

The rigidity of budgetary and planning cycles restricts exploratory processes. It is difficult to attract and allocate resources when managers cannot be specific about the exact solution, process, and timeline an agency plans to follow in developing a policy.

That said, there is increasing use of Agile in government policymaking and it is achieving some success. The general principles of understanding the user, working closely with implementers, user-testing assumptions, and iterating until the problem is solved, are all policymaking best practices. I will highlight some examples of Agile in action in the public sector in my next post.

The views expressed in the Government Innovators Network blog are those of the individual author(s) and do not necessarily reflect those of the Ash Center for Democratic Governance and Innovation, the John F. Kennedy School of Government, or of Harvard University.

 (https://www.printfriendly.com/print?url=https%3A//www.innovations.harvard.edu/blog/new-public-policymaking-frontier-blockchain%3Futm_source%3DPANTHEON_STRIPPED%26utm_medium%3DPANTHEON_STRIPPED%26utm_campaign%3DPANTHEON_STRIPPED)

RELATED TOPICS

Public Management (/blog/topic/206)   Governance & Politics (/blog/topic/163)