Iterative and Incremental Standard for the Digital Public Services Policy

Last review: Jan 2024
Next review: Jan 2025

Digital Citizen Experience, Ministry of SaskBuilds and Procurement

This document outlines the Iterative and Incremental Standard as a component of the Digital Public Services Policy. These standards inform Government of Saskatchewan service design for public-facing digital services.

Background

Iteration involves subsequent progression where **incremental** means sections of work are done based on available resources and not necessarily in a linear or waterfall way. Several completed increments can be chosen for the next iteration, where other partially completed increments may make it into a future iteration.

Those responsible for digital solutions need to consider these concepts in preliminary stages of planning. Iterations can be seen as “releases" where increments can be seen as various work items done in “sprints.”

This standard allows for flexibility as projects may use various design and development practices or methodologies.

Standard Framework

A clear outline of the iterative approach should be built into RFP language and defined at the start of each project with the purpose of outlining the base requirements for the Minimal Viable Product (MVP). Future iterations can be added as a continuous improvement plan.

Iterative Approach

Iteration is an approach designers and developers use when building and improving services. Each iteration builds onto the last, getting closer to the ideal solution over time.

The overall vision/concept and actual delivered solution converge over numerous iterative releases. The planning and deliverables of each iteration are revised as feedback or environmental factors change. Plans are tweaked on a small scale rather than reworking an entire project. This is less costly than fixing or improving solutions that use a waterfall approach.

The first release usually deploys the primary functions of a service as an MVP in a longer scheduled iteration. Users and stakeholders see results early and often in subsequent iterations.

An iteration involves several steps that may differ depending on the implementation and project constraints, but always involves service design planning, retrospectives (what worked and did not), regression testing, change management and documentation. Feedback and analytics of that iteration are used to plan the following one.

Incremental Approach

Incremental development adds new features with an iteration. Developers from different teams may be actively coding various sections of the product, backlog requests or bug fixes. These are usually selected though grooming user stories, estimating effort of each story and a list of prioritized items are assigned for development.

Functionality is incremental (through user story completion). An iteration is made up of several increments chosen by the business.

Some incremental pieces can also be enhancements developed during slower workloads and

Iterative and Incremental Standard for the Digital Public Services Policy

Digital Citizen Experience, Ministry of SaskBuilds and Procurement

This document outlines the Iterative and Incremental Standard as a component of the Digital Public Services Policy. These standards inform Government of Saskatchewan service design for public-facing digital services.

may be held back for an iteration further into the roadmap (in part due to prioritization or dependency-related factors).

Iterations usually are time-boxed ranging from one to eight weeks before a release, and during this time, some incremental pieces (also usually time-boxed) may be incomplete where others have already been waiting in a finished state for inclusion.

Incremental development spreads out workloads of the project teams and creates less waste. One team is not waiting for another team to finish before beginning their work.

Increments that have dependencies on other increments need to be identified during user story grooming.

Resources

Iteration and incremental work involve cross-functional teams working together.

This includes a mix of planning, analysis, design, prototype, technical architecture, code development, testing, release and change management.

The solution may have a stream of project staff assigned to handle iterations, and a separate stream of staff handling the operation of the solution.

Benefits of this Approach

- Identifies defects early in the process by including user and customer feedback with each iteration.
- More efficient at finding code defects, inconsistencies, or gaps in design.
- Ongoing revisions to unclear user stories or backlogs are addressed to ensure they remain groomed, updated, small and included in roadmap planning.
- Saves time on documentation and knowledge management, ensuring training material and technical documentation remain current.
- Interfaces get improved as feedback comes in, and as platforms/browsers change over time (requiring the interface elements to be adjusted).
- Security scans and patches become part of the process, ensuring the solution is always updated with the most recent frameworks and plugins that are also scanned before each release.
- Your initial iteration reaches the market more quickly and is responsive to changing requirements as user expectations and needs evolve. Users also see continual improvements that lead to increased trust with the solution.