



Agile Rollout Principles

Overview

This document lays out the underlying principles of agile rollout: how to quickly roll out collective action initiatives and programs, creating dynamic and agile partnership frameworks that will produce greater impact at lower cost in less time. By following the principles and guidance laid out below, collaborators can accelerate and amplify impact of their programs.

Agile Principles

As a group, collaborators commit to:

- Falling in love with problems before falling in love with specific solutions
- Valuing relationships and collaboration over rote processes and bureaucracy
- Quickly delivering functional programs instead of comprehensive but delayed programs
- Integrating with existing efforts instead of creating new or isolated programs
- Adapting to change instead of following a fixed plan

Ruthless prioritization is essential for a multi-stakeholder collaborations to remain effective and avoid mission dilution and mission creep. It is the responsibility of the conveners and collaboration managers to enforce priorities, which will entail de-prioritizing tasks as necessary.

S.M.A.R.T. Outcomes

When organizing work, each collaborative team sets its own outcomes that align with the overarching collective objectives. These outcomes should follow the S.M.A.R.T. system:

- **Specific**: Specify precisely what value the outcome will provide, who does the work, and who benefits. Using only one action verb per outcome is preferred.
- **Measurable**: Outcomes are testable propositions that can be measured against a defined target.
- **Assignable**: Outcomes are clearly assigned to a responsible party or parties.
- **Realistic**: Outcomes are attainable within the given time frame and resources allotted.
- **Time-based**: Outcomes include a timeframe for completion.

Example:

- **Vague outcome**: “Training module is designed.”
- **S.M.A.R.T. outcome**: “Training module for airline personnel on identifying illegal wildlife trafficking is designed by the Training & Awareness Working Group by October 1, 2015.”

Maximizing Learning through Iteration

An iterative design and implementation process maximizes the amount of information learned about target beneficiaries per dollar spent, ultimately leading to better outcomes. As an example, rather than trying to develop the perfect, end-to-end system or training program, teams should quickly produce a specific module or class for a subset of early adopters to try. We then gather feedback, analyze and learn from the data, and incorporate it into the next iteration of the design. This process is known as “rapid prototyping.”

Rapid prototyping accelerates adoption by including target communities at every step in the process, ensuring both a better and a more readily-adopted solution. Crucially, teams must remain flexible and open to big changes to the design, even late in the game.

Figure 1: Traditional Project Process

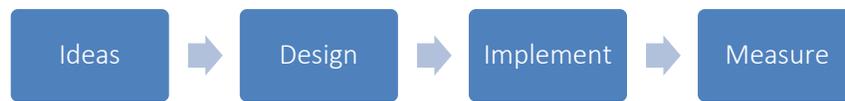
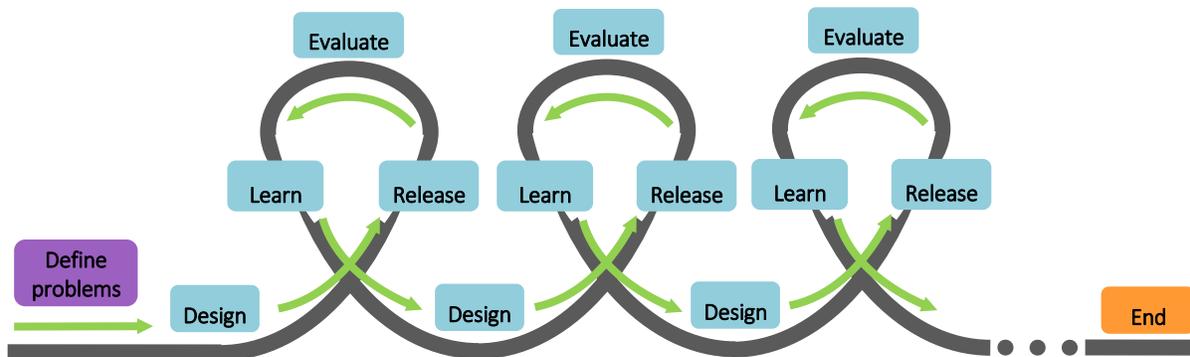


Figure 2: Rapid Prototyping Process



Short timescales are an integral aspect of iterative design. After laying out 12-18 month goals, teams should break work down into "design sprints" with deliverables due in three to six month increments. Deliverables may include a range of internal and external items with soft or hard outcomes (see Table 1 for examples).

Table 1: Types of Deliverables

	Internal	External
Soft	Memorandum with recommendations to Core Team	MOU with an airline to provide guidance on training of airline employees
Hard	Downloadable training manuals for use by members	New training program