


Manage change in an incremental migration effort

04/04/2019 • 2 minutes to read • Contributors 

In this article

[Release backlog](#)

[Iteration backlog](#)

[Next steps](#)

This article assumes that migration processes are incremental in nature, running parallel to the [govern process](#). However, the same guidance could be used to populate initial tasks in a work breakdown structure for traditional waterfall change management approaches.

Release backlog

A *release backlog* consists of a series of assets (VMs, databases, files, and applications, among others) that must be migrated before a workload can be released for production usage in the cloud. During each iteration, the Cloud Adoption team documents and estimates the efforts required to move each asset to the cloud. See the "Iteration backlog" section that follows.

Iteration backlog

An *iteration backlog* is a list of the detailed work required to migrate a specific number of assets from the existing digital estate to the cloud. The entries on this list are often stored in an agile management tool, like Azure DevOps, as work items.

Prior to starting the first iteration, the Cloud Adoption team specifies an iteration duration, usually two to four weeks. This time box is important to create a start and finish time period for each set of committed activities. Maintaining consistent execution windows makes it easy to gauge velocity (pace of migration) and alignment to evolving business needs.

Prior to each iteration, the team reviews the release backlog, estimating the effort and priorities of assets to be migrated. It then commits to deliver a specific number of agreed-on migrations. After this is agreed to by the Cloud Adoption team, the list of activities becomes the *current iteration backlog*.

During each iteration, team members work as a self-organizing team to fulfill commitments in the current iteration backlog.

Next steps

After an iteration backlog is defined and accepted by the Cloud Adoption team, [change management approvals](#) can be finalized.

Approve architecture changes prior to migration