Playbook Methods Repository

# **Gated Rollouts**

Release product versions through phased and/or parallel deployments to a subset of users and/or infrastructure so production risks can be mitigated by safe rollbacks before a full-scale launch. Every deploy phase can be gated based on incremental feedback.

### Remote Agility: **•** High

### Linked Tactic(s): Release Management

## Why we do it:

Maintain a high quality service and user experience by reducing the risk of releasing bugs, performance degradations, and breakages. Release changes incrementally so you can monitor the effects of each change and rollback if needed. Use automated checks to reduce manual work in the process and allow for quicker iterations. See also [Playbook Methods: CI/CD](https://docs.google.com/document/d/1m0IktMM2gJB_jAk3M8ePEND46LS-H8hN_RlYrGIjOQQ/edit#).

## 

## When to apply it:

* Release Management

## Best Practices & Considerations:

* Determine which phases you want to have, who they are released to, and what to check to move on to the next phase. Some sample phases may be:
  + Internal users, Alpha, Beta, Everyone
  + Internal users, 1% users, 10% users, Everyone
* You’ll want to balance release quality and speed. What’s the minimum we can do to have confidence in the release?
* Rollbacks are a critical part of this process. If a release has an issue, how can we quickly (automatically?) get back to a good state?
* Alerts are also critical. When can the system recover itself and when does it need manual intervention?

## Responsible roles:

* Software Engineer to implement the deployment pipeline and gates. This may be a more specialized engineer such as a DevOps engineer.
* QA Engineer to support any manual testing gates and automated testing gates as needed.
* Product Manager to define requirements for the release process and gate checks.

## Tools:

* Online tools/platforms/services
  + [AWS CodePipeline](https://aws.amazon.com/codepipeline/)
  + [Google Cloud Deploy](https://cloud.google.com/deploy)
  + [Azure Pipelines](https://docs.microsoft.com/en-us/azure/devops/pipelines/?view=azure-devops)
* Websites
  + xx
* Databases
  + xx
* Other
  + xx

## 

## Thoughtworks Examples - Linked

* Client working docs, airtable, miro/mural boards
  + xx
* Client polished presentations/deliverables
  + xx
* Internal assets - clinic materials / guild docs
  + xx

## 

## Learn more: How we do this?

* Templates (docs, decks, sheets, miro, etc.)
  + xx
* How-To Resources (external or internal)
  + [Set Up a CI/CD Pipeline on AWS](https://aws.amazon.com/getting-started/hands-on/set-up-ci-cd-pipeline/)
* Outside References (articles, books, etc.)
  + [Release deployment control using gates](https://docs.microsoft.com/en-us/azure/devops/pipelines/release/approvals/gates?view=azure-devops) - overview of the concept from Azure Pipelines
  + [How release canaries can save your bacon](https://cloud.google.com/blog/products/gcp/how-release-canaries-can-save-your-bacon-cre-life-lessons) - deep dive on canary releases from Google Cloud
* Sub-set Activities
  + xx

## 