# GitHub Migration Roadmap

## Summary

PepsiCo is transitioning its codebase from Azure Repos to GitHub. Aligning CI/CD with GitHub Actions allows seamless integration, workflow reusability, and AI-driven development with GitHub Copilot. This document outlines the strategic differences, a structured migration timeline, and key benefits for PepsiCo.

## 1. Why Move from Azure DevOps to GitHub Actions

* Azure DevOps allows unlimited projects, but Microsoft recommends keeping the count under 1,000. As project numbers grow, managing secrets, templates, and access becomes increasingly complex. GitHub avoids this by scaling through organizations and repositories.
* Reusable workflows reduce repeated effort
* Org-wide governance and access control is simpler
* Marketplace actions(1000 +) support faster integrations which avoids use of custom scripts for common tools.
* Native Copilot integration offers productivity boosts

## 2. ADO – Github Migration Timelines Automation (Phase-wise)

|  |  |  |
| --- | --- | --- |
| Timeline | Phase | Key Activity |
| June 16 | Repo Migration | Pilot with DSX apps |
| June 25 | Secret & Variable Setup | Recreate ADO variable groups, environments as GitHub secrets |
| June 30 | Runners | Reuse existing AKS-based self-hosted runners in GitHub Actions |
| July 9 | Workflow Conversion | Translate ADO YAML pipelines into modular GitHub workflows |
| July 14 | Integration | Integrate toolchains like SonarQube, ACR, and AKS |
| July 20 | Semgrep SAST Integration | Integrate Semgrep into GitHub Actions workflows for static code analysis as part of the CI pipeline. |
| July 31 | Initial Validation | Run GitHub Actions workflows and compare with ADO pipelines |
| August 29 | Pipeline Reusable template | Create reusable GitHub Actions workflow templates.**This activity will start along with initial migration step for DPA approved tech stacks.** |
|  |  |  |
| September 12 | E2E Validation | Validate full pipeline across migrated services |
| TBD | Rollout | Begin phased migration by app criticality |

## Note: These timelines are subject to change incase of any access blockages.

## Migration Phase Details

### Repo Migration

We use the GitHub Importer CLI to automate the migration of repositories from Azure DevOps to GitHub while preserving branches, commit history, and tags. Automation is achieved using shell scripts that takes input of one Azure Repos url , invoking the GitHub Importer CLI with batch parameters.

### Secret & Variable Setup

Azure DevOps variable groups are converted into GitHub secrets using CLI-based Python scripts. The scripts utilize GitHub CLI (`gh secret`) to create secrets with proper naming conventions and apply them at the repo or environment level.

### Runners

Our existing AKS-based self-hosted runners are registered to GitHub Actions. GitHub REST APIs are used in conjunction with bash scripts to automate tagging, registration, and connectivity validation.

### Workflow Conversion

Azure DevOps YAML pipelines are modularized into GitHub workflows (build, test, deploy) using custom YAML parsing scripts. Validation scripts ensure schema compliance and structural consistency for GitHub workflows.

### Integration

Integration with tools like SonarQube, ACR, and AKS is implemented using GitHub Marketplace actions and custom scripts. Secrets and environment configurations are passed securely using reusable input patterns in GitHub workflows.

### Semgrep SAST Integration

We integrate Semgrep into GitHub Actions pipelines to perform static application security testing (SAST) on pull requests and merges. Automation: Semgrep is configured as a reusable GitHub Action with predefined security rules tailored to each tech stack.

### Initial Validation

GitHub workflows are validated by executing side-by-side with ADO pipelines. Shell scripts compare logs, artifacts, and runtimes using diff tools to ensure output parity.

### Pipeline Reusable template:

Complete Github action workflow which includes all DPA approved techstacks

### E2E Validation

Full pipeline validation includes unit testing, artifact verification, and Helm-based AKS deployments. Automation scripts trigger workflows and collect results using GitHub Actions matrices and test assertion checks.

### Rollout

The rollout begins with non-critical applications and scales based on validation. Automation scripts classify repos by criticality and propagate GitHub workflows with dry-run validations.

## 3. Template Strategy for AKS Deployments GitHub Actions workflows will be optimized for AKS (Azure Kubernetes Service) deployments. The goal is to simplify the current Global Template (GT) setup by reducing the number of templates while preserving flexibility and ease of maintenance.

## 4. Future Roadmap (Enhancements)

* Reusable deployment workflows will be extended to support EKS (Amazon Elastic Kubernetes Service), ensuring cloud-specific config is parameterized.
* Integrate Github native tools like Codeql, Secret Scanning for Security.
* Optional chatbot integrations (Slack or Microsoft Teams) will be explored for deployment alerts.
* Optional hooks for observability (e.g., Prometheus, Azure Monitor) may be included in later pipeline revisions.
* Templates will remain modular and aligned with PepsiCo’s platform engineering roadmap to support scalable, multi-region deployments.