

AD04 Testing Times: Implementing A CI/CD model for SAS

14th November 2022

Paul Shannon
Amadeus Software Ltd.





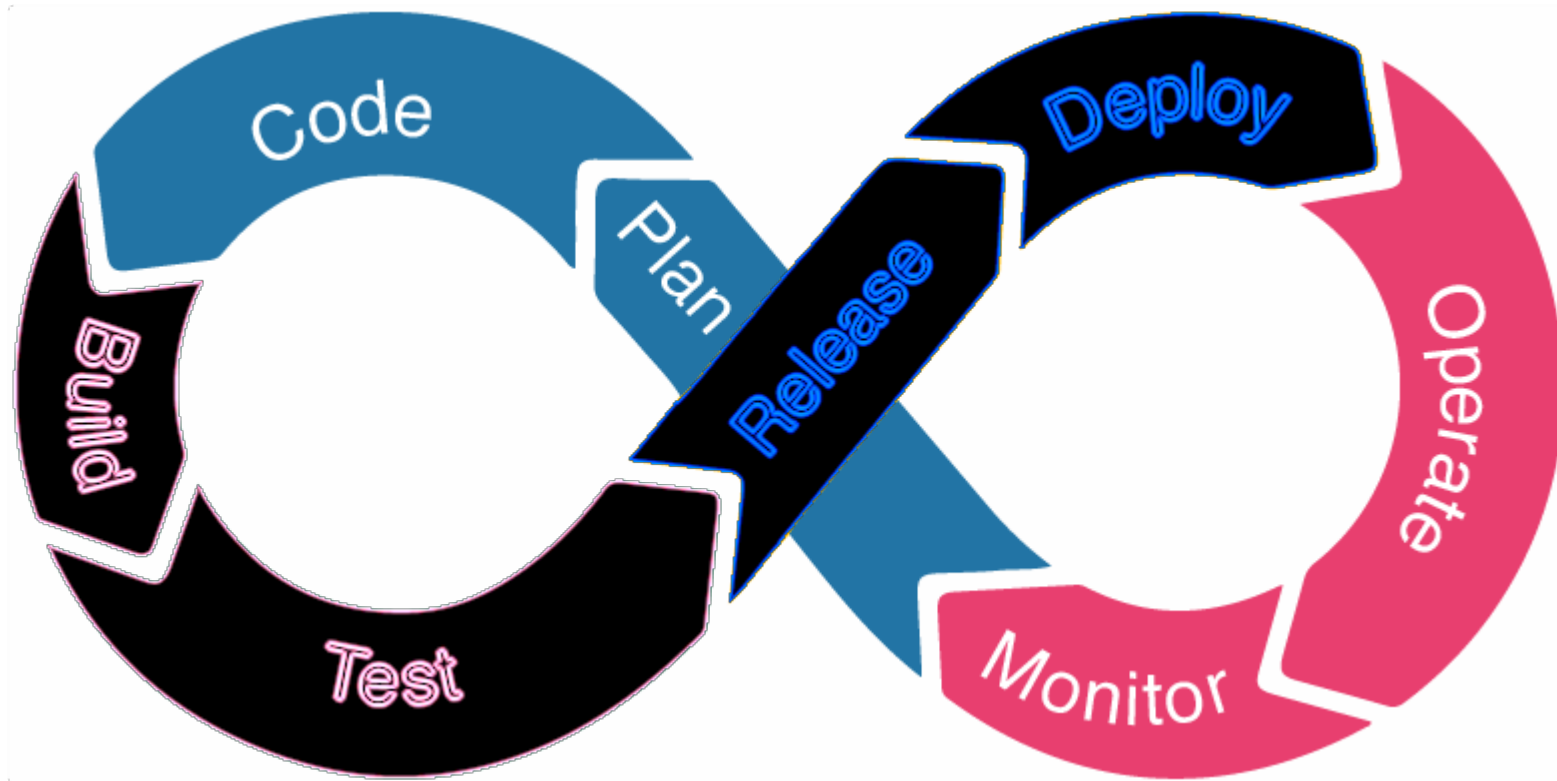
Introduction

- Background
- CI/CD in the DevOps Lifecycle
- The role of Git in CI/CD
- Continuous Integration
- Continuous Delivery
- Next steps: Artifacts and Continuous Deployment
- Summary

Can't we just batch submit a job?... CI/CD gives more:

Fully automated	End-to-end testing, removes risk of human error.
Standardisation	Same process used in SAS, R, Python etc.
Security Improvement	Individual users don't need permission to run jobs.
Reporting	Comprehensive logs and history of test runs.

CI/CD in the DevOps Lifecycle



DevOps lifecycle starts with Plan, Code, Build, Test, etc.

CI covers the Build and Test steps and CD is Release and Deploy.

In this presentation we will take existing unit tests, automate their execution and the deployment of the source programs.

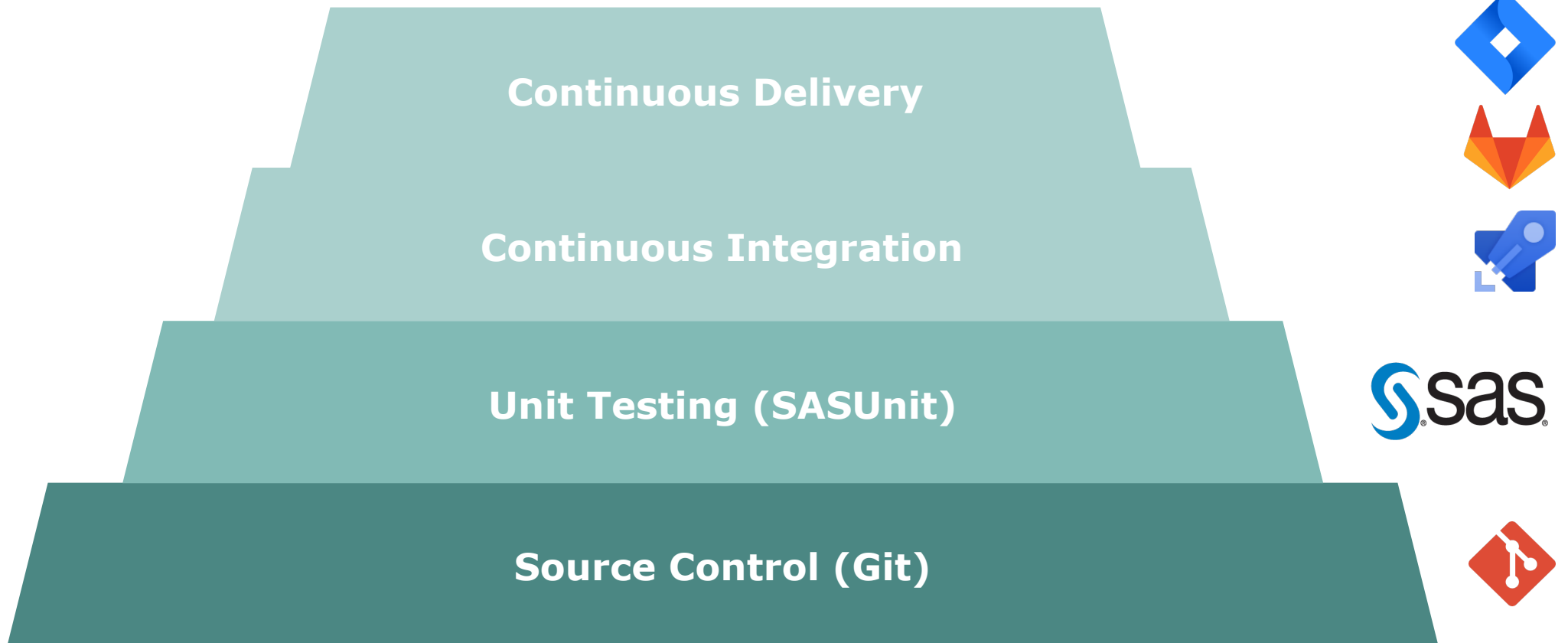


CI/CD in the DevOps Lifecycle

Unit Testing:

- A prerequisite for CI/CD is that you have an automated Unit Testing system in place.
- This is not a topic to cover today, but there are multiple papers on Unit Testing in SAS, including ones presented at PhUSE.
- Various tools are available for Unit Testing in SAS, we will be using SASUnit in this demonstration, but any unit testing solution is compatible with this methodology.

The Role of Git in CI/CD





Continuous Integration

Continuous integration **refers to the build and test stages of the DevOps process.**

Every revision that is committed triggers an automated build and test.

Start with a Git Repository. Required for our DevOps model.

We will create a Git Pipeline to program the build and execution of unit tests, using the YAML language

The pipeline runs automatically each time a code change is made, defining the “continuous” in Continuous Integration.

What is a Git Pipeline?

- A set of instructions to build, test or release content in a repository.
- Written in a syntax called YAML.
- Supported by all DevOps software solutions.

What is YAML?

- Yet Another Markup Language
- Human-readable
- Similar syntax to Python
- Indentation nesting
- Save YAML file(s) in your Git repository

Continuous Integration: SAS

- In SAS there is no need to compile or build programs
- Method of Continuous Integration is simple:
 - Copy Git repository to a SAS server
 - Execute unit tests
- Different environments have different methods for executing unit tests:
 - Remote PowerShell execution for Windows
 - Secure Shell (SSH) for Linux environments

PhUSE 2022 +

Overview

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Artifacts

Project settings <<

Connect

Select

Configure

Review

New pipeline

Where is your code?



Azure Repos Git

YAML

Free private Git repositories, pull requests, and code search



Bitbucket Cloud

YAML

Hosted by Atlassian



GitHub

YAML

Home to the world's largest community of developers



GitHub Enterprise Server

YAML

The self-hosted version of GitHub Enterprise



Other Git

Any generic Git repository



Subversion

Centralized version control by Apache

[Use the classic editor](#) to create a pipeline without YAML.

YAML Code: Inspection

```
1 # Buld Pipeline for SAS programs
2 # Uses SASUnit for executing unit tests
```

```
4 trigger:
5   - integration
```

Identifies the branch that will trigger this pipeline

```
7 pool:
8   vmImage: windows-latest
```

Set the O/S to be used within Azure DevOps. Should match your SAS environment

```
10 steps:
11 # Azure file copy
12 # Copy files to SAS deployment
13 - task: AzureFileCopy@5
14   inputs:
15     SourcePath: '*'
16     azureSubscription: 'Visual Studio Enterprise Subsci
17     Destination: 'AzureVMs'
18     storage: 'phusecicdxfer'
19     resourceGroup: 'PHUSE2022-PREP'
20     MachineNames: '$(SASServer)'
21     vmsAdminUserName: '$(AdminUser)'
22     vmsAdminPassword: '$(DestPW)'
```

Script the steps. Each step comprises a task name and a list of inputs.

The first is to copy files to an Azure server

```
23 Target: $(TestFolder)
24 enablePS: true
25 skipCopy: false
26 BlobPrefix: 'phuse'
27 Clean: true
28   beforeCopy: true
```

Use PowerShell to run SASUnit (Windows)

```
29 # Run PowerShell script on SAS deployment
30 # Runs SASUnit test suite
31 - task: PowerShellOnTargetMachines@3
32   inputs:
33     Machines: '$(ServerWinRM)'
34     UserName: '$(AdminUser)'
35     Password: '$(DestPW)'
36     InlineScript: '.\sasunit.ps1'
37     ErrorActionPreference: 'silentlyContinue'
38     WorkingDirectory: '$(TestFolder)'
39     RunPowershellInParallel: false
```

Use variables to avoid storing credentials in your Git repository.



Continuous Delivery

- Continuous delivery automates the entire software release process. Every revision triggers an automated flow that builds, tests, and then stages the update.
- The final decision to deploy to a live production environment is triggered by the developer.
- For SAS programs, use another Git Pipeline.
- Important to only deploy/release assets intended for production.
- Exclude unit test files, data, YAML files, etc.



Continuous Delivery: SAS

- For SAS programs, the CD Pipeline is very similar to CI:
- Method of Continuous Delivery :
 - Copy **source files** from Git repository to a SAS server
 - Exclude build/test files such as YAML files, or SASUnit
- Configure this Pipeline only when the CI pipeline runs successfully.



PhUSE 2022



Overview



Repos



Pipelines



Pipelines



Environments



Releases



Library



Task groups



Deployment groups



Artifacts



Project settings



Pipelines

Recent

All

Runs

New pipeline



Filter pipelines

Recently run pipelines

Pipeline

Last run



Release and Deploy

#20221023.5 • Update release.yml for ...

Build completion for main

1h ago

1m 24s



Build and Test

#20221023.13 • add fullsTimer to auto...

Individual CI for integration

1h ago

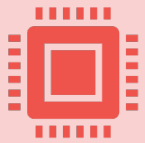
1m 46s

Continuous Delivery: YAML

```
1  trigger:
2  - integration
3
4  pool:
5    vmImage: windows-latest
6
7  steps:
8  # Azure file copy
9  # Copy files to Azure Blob Storage or virtual machines
10 - task: AzureFileCopy@5
11   inputs:
12     SourcePath: '$(Build.Repository.LocalPath)\src\*'
13     azureSubscription: 'Visual Studio Enterprise Subscription'
14     Destination: 'AzureVMs'
15     storage: 'phusecicdxfer'
16     resourceGroup: 'PHUSE2022-PREP'
17     MachineNames: '$(SASServer)'
18     vmsAdminUserName: '$(AdminUser)'
19     vmsAdminPassword: '$(DestPW)'
20     TargetPath: '$(ReleaseFolder)'
21     enableCopyPrerequisites: true
22     skipCACHeck: true
23     BlobPrefix: 'phuse'
24     CleanTargetBeforeCopy: true
```

- Very similar YAML to the CI script. Differences are:
- The SourcePath value only select files in “\src\” folder
- Different Target Path
- Configuring this Pipeline to run on successful run of CI pipeline is not saved in the YAML file.

Where Next?



Infrastructure as code



Apply CI/CD to cloud-based servers, scripting an entire SAS deployment in YAML.



Automate end-to-end testing of a platform deployment



Automation validation and re-validation of a *Validated Statistical Environment*

Summary

Using CI/CD in SAS is best practice.

Utilises the power of Git source control.

Many DevOps tools are available to support create CI/CD pipelines for SAS programs.

Sensitive values such as passwords are securely controlled.

Supports cloud or on-premise environments.

CI/CD is being expanded into SAS environments through Infrastructure as Code.



Do you have any questions?



Paul Shannon
Amadeus Software Ltd.

Tel: +44 (0)1993 848010
E-mail: paul.shannon@amadeus.co.uk
Web: www.amadeus.co.uk