

# Using AL-Go for GitHub

## Speakers

Krzysztof Bialowas  
Freddy Kristiansen



- Save time getting started
- Improve collaboration
- Fix bugs faster...
- Improve quality
- Improve security
- Know before your customers
- Advanced scenarios
- Future investments

**Save time  
getting started**



People saying that they are  
too busy to setup DevOps  
is like woodcutters saying they  
are too busy to sharpen their saw

# Save time getting started

The goal is that AL-Go for GitHub will support 100% of the functionality needed by 90% of the partners

Yes, there is a learning curve, but you won't have to develop and maintain workflows for normal DevOps processes.

Developing and maintaining DevOps is a time consuming task that require a special skillset

Your resources can focus on providing customer value

“World” of GitHub – lot's of innovations happening every day...

.dev

App

VS Code Ext

Desktop App

CodeSpaces (coming)

# DEMO

Create a new repo with full CI/CD in minutes  
No development, no maintenance, no worry

# Getting started

## A GitHub Organization

- Owner of the code

- Pays the bill

- Share secrets, code, GitHub packages and more

- Can contain any number of AL-Go repositories

## An AL-Go repository

- The release vehicle

- The security model

- Can contain any number of AL-Go projects

## An AL-Go project

- The shipping vehicle

- Defines dependency resolution

- Can contain any number of apps being built and tested together



# Improve collaboration





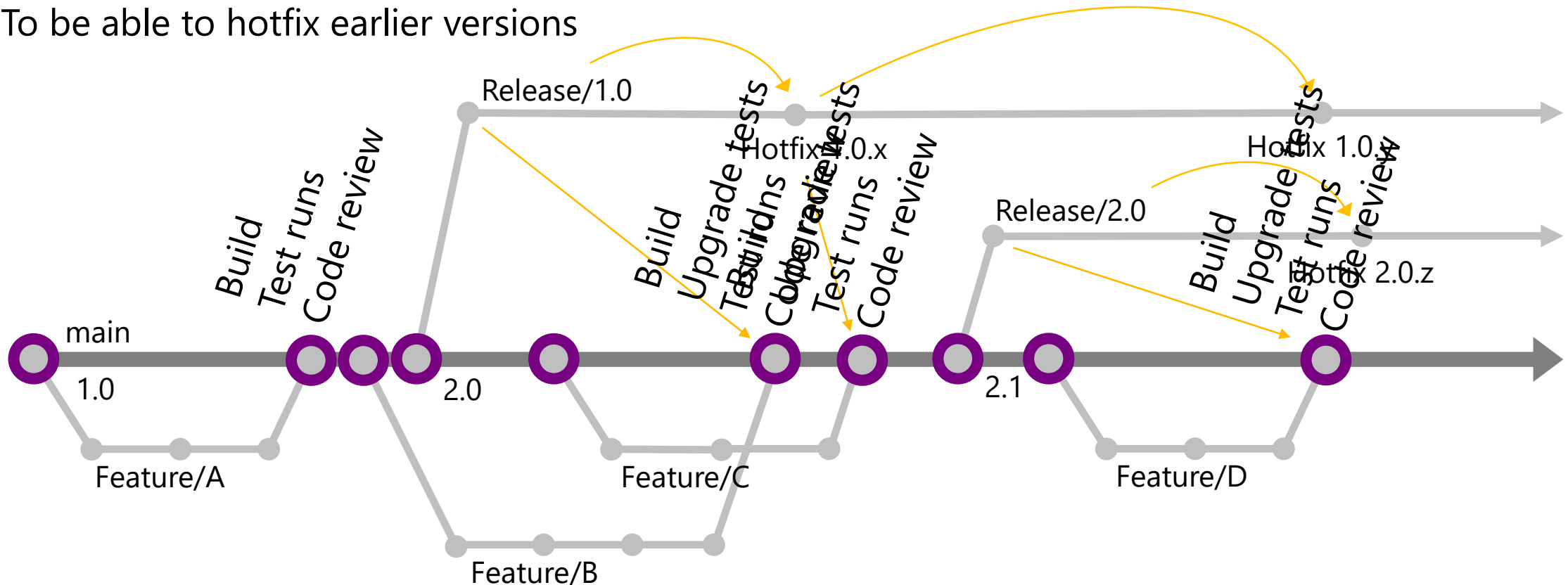
# Improve collaboration

## Use feature branches and Pull Requests

Keep main branch "shippable at all times"

## Use release branches

To be able to hotfix earlier versions



# DEMO

Create a Pull Request

Code Review

Merge

# Fix bugs faster



# Fix bugs faster

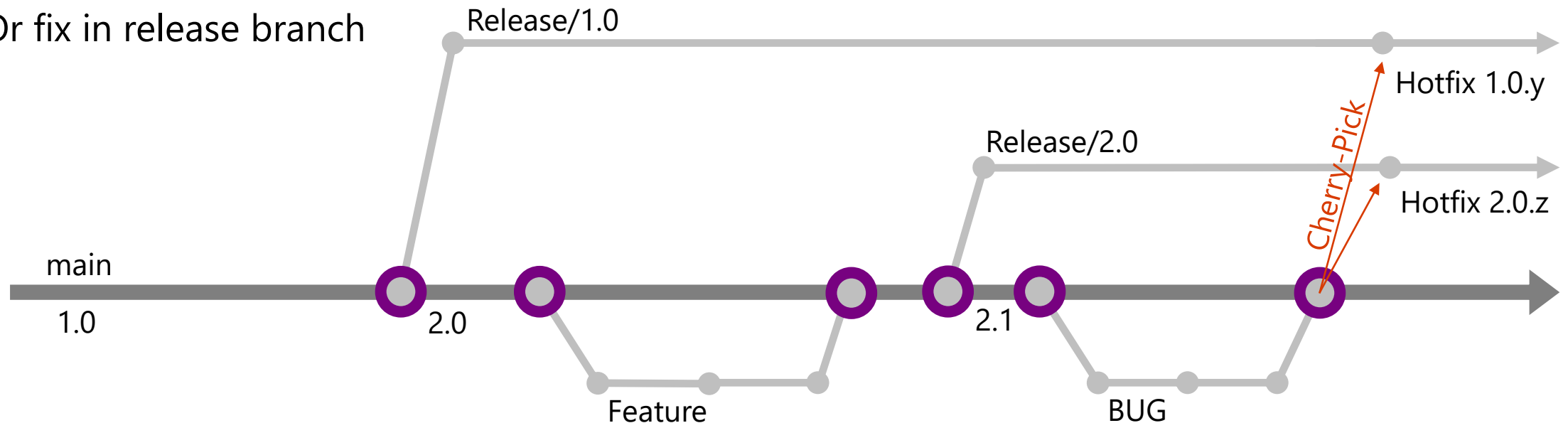
Supporting a well-known (and proven) branching strategy

Use release branches

Easily locate the source for the version your customer is running

Fix bugs in main, cherry pick to release branches

Or fix in release branch



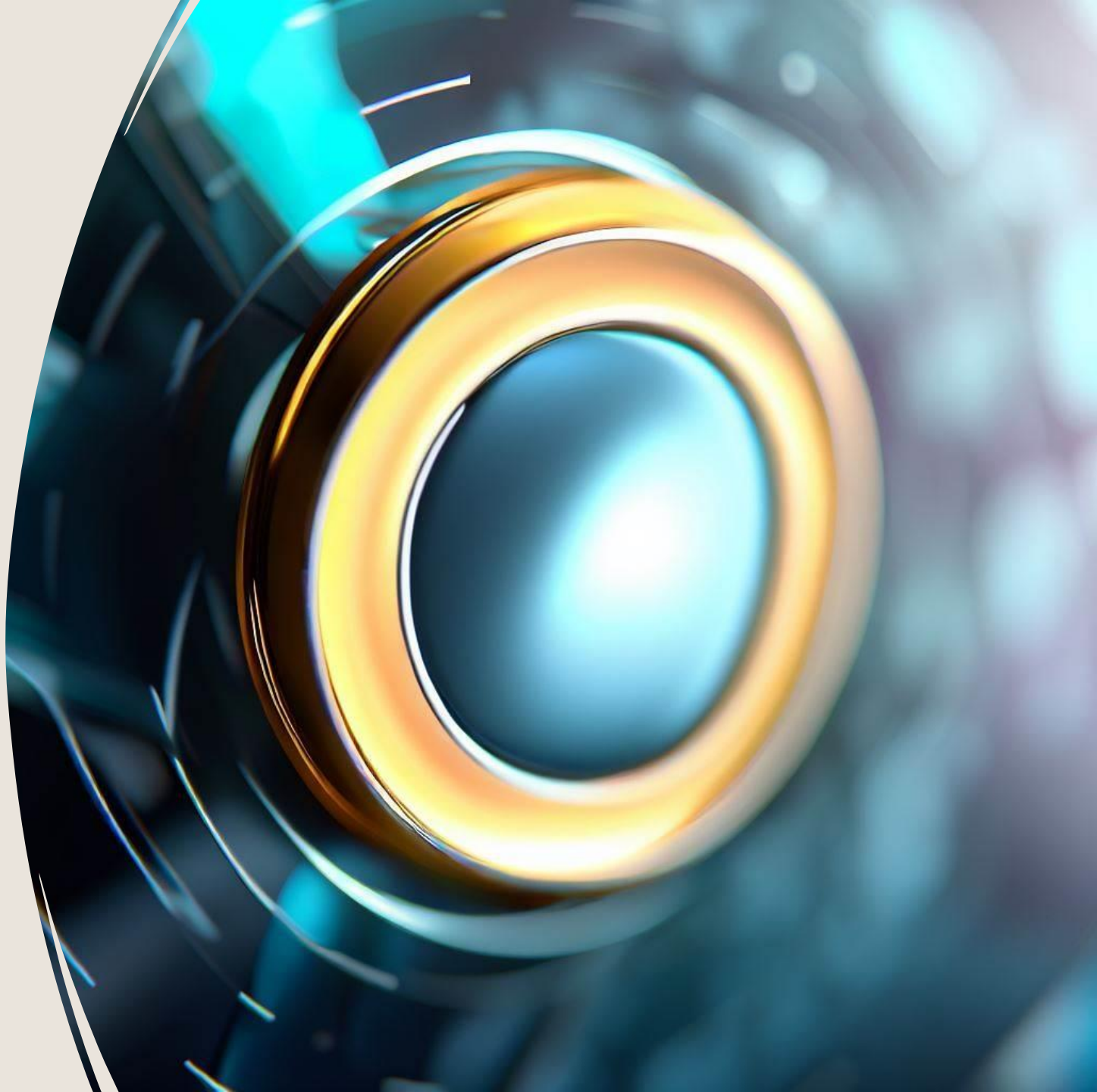
# DEMO

Fix a bug in main

Move bug fix to release branch

Release hotfix

# Improve quality



# Improve quality

Run automated tests as part of every pull request

Run upgrade tests as part of every pull request

Locate the correct "previous version" of your app

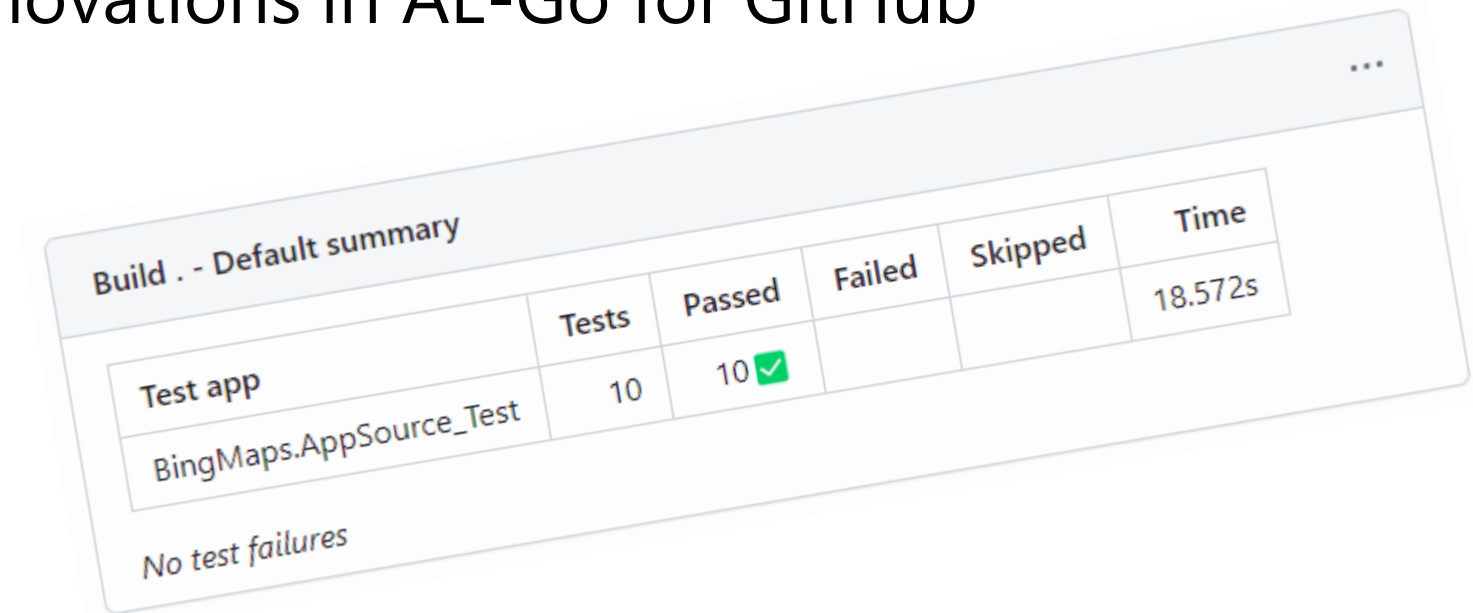
## Notifications

Subscribers automatically get notified about changes, failures, etc.

Take advantage of future innovations in AL-Go for GitHub

Improved test capabilities

API testing



The image shows a screenshot of a GitHub Actions build summary. The title is 'Build . - Default summary'. Below the title is a table with the following data:

	Tests	Passed	Failed	Skipped	Time
Test app	10	10 ✓			18.572s
BingMaps.AppSource_Test					

At the bottom of the summary, it says 'No test failures'.



# DEMO

Automated upgrade test  
Automated testing

# Improved security



# Improve security

## Continuous deployment to a QA environment

Allowing customers access to the latest and greatest

Get feedback early

## Automated deployment to production environments

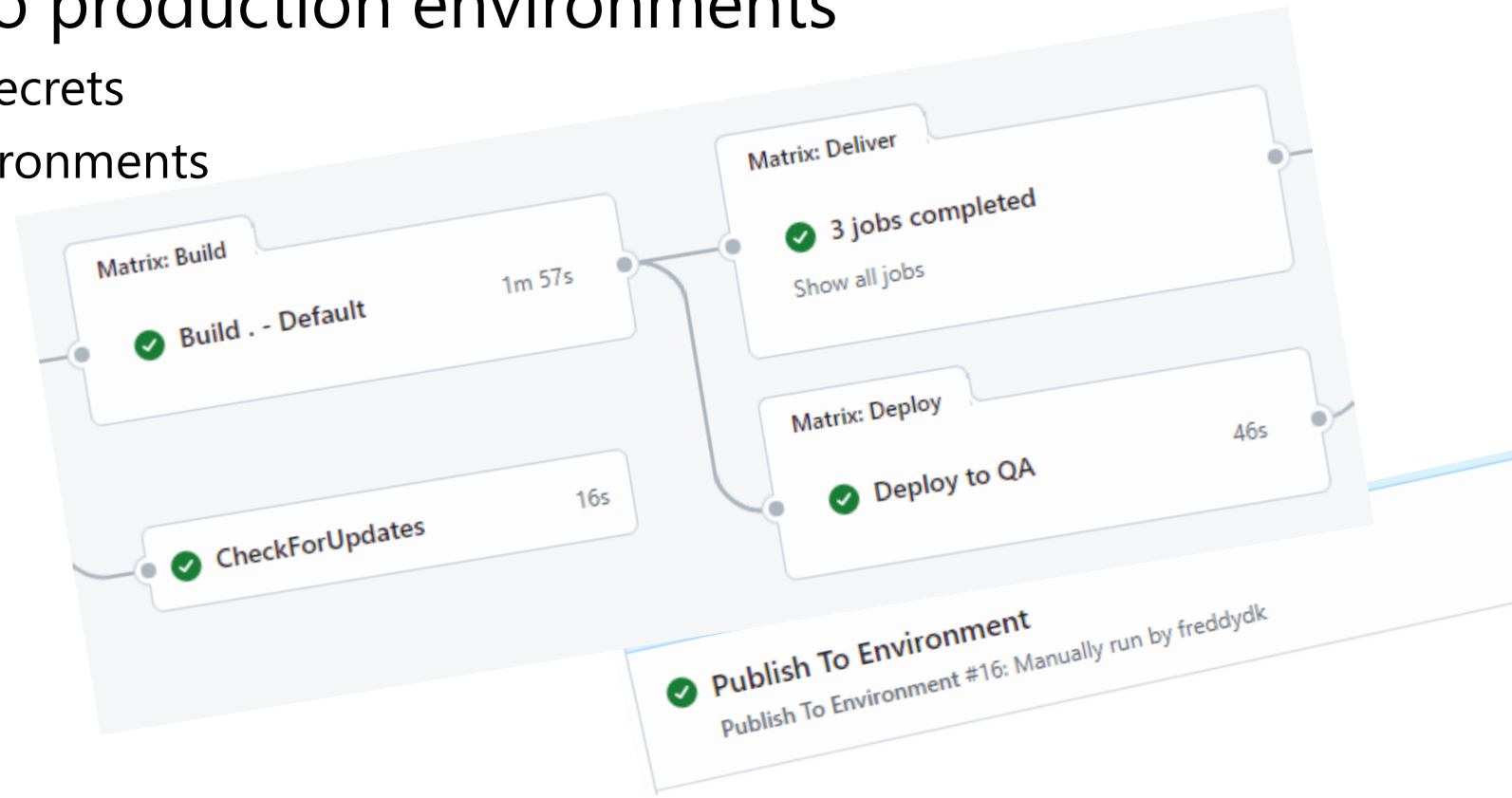
Authentication stored in GitHub secrets

No direct access to customer environments

## Avoid silly mistakes

Selecting the wrong file

Not having the latest source



# DEMO

Continuous Deployment to Sandbox environment  
Deploy to production environment



# Know before your customer



# Know before your customer

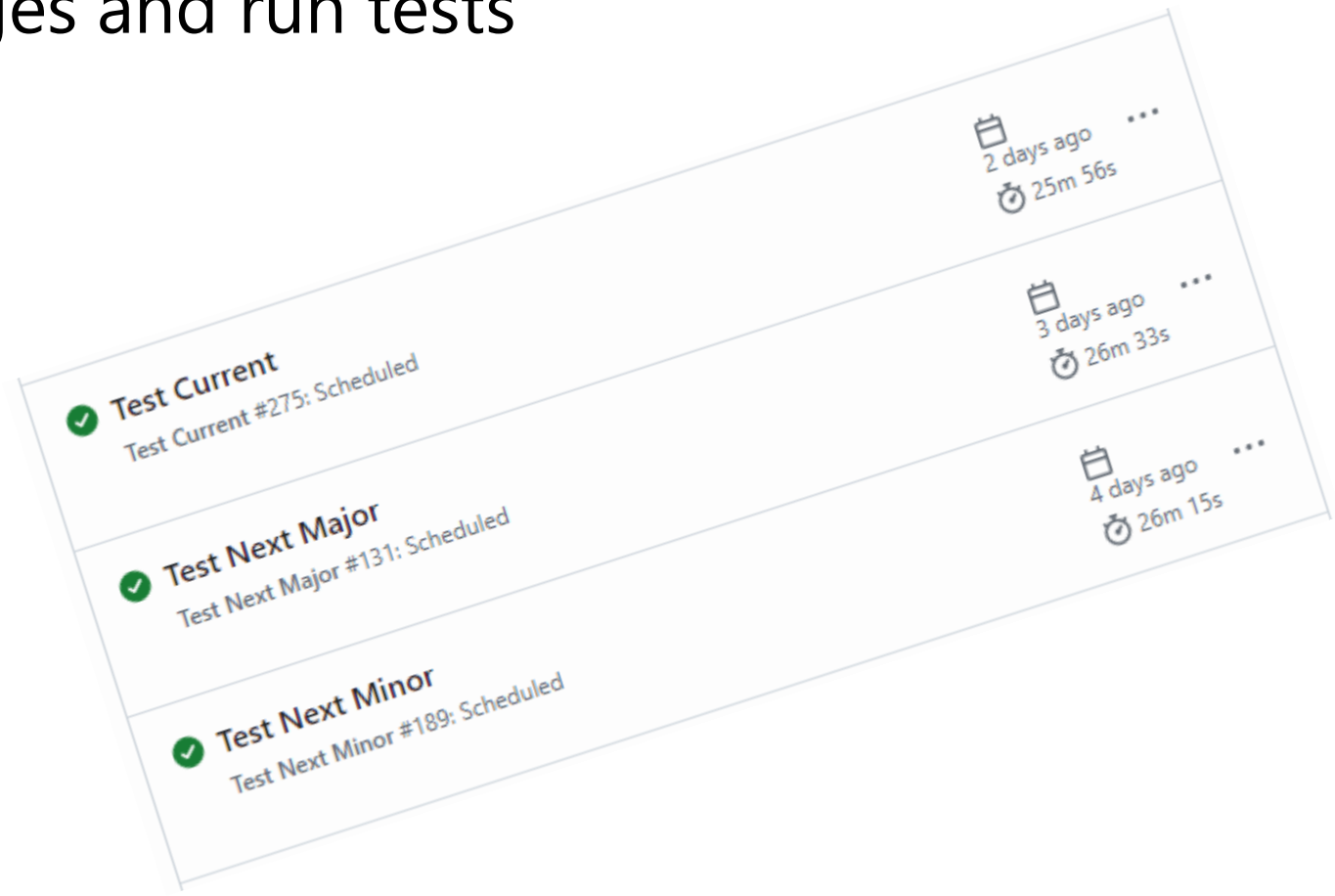
Scheduled tests of your app running against

Next Major, Next Minor and Current version of Business Central

Build, check for breaking changes and run tests

Be proactive

Notify your customers early



# DEMO

Test Next Major,  
Test Next Minor and  
Test Current Business Central version



# Advanced scenarios

Project Dependencies  
GitHub Packages



# Advanced scenarios – GitHub Packages

AL-Go for GitHub supports using GitHub packages for internal dependency resolution.

Just add a secret called **GitHubPackagesAuthContext** to enable. CI/CD workflows will now automatically deliver to GitHub Packages. Other repositories/projects can now find these apps.

DEMO:

<https://github.com/FreundDKorg/app1>

<https://github.com/FreundDKorg/app2>

# Advanced scenarios – Project Dependencies

AL-Go for GitHub supports building projects by dependency order. Just add a repository setting called **useProjectDependencies**: true and run Update AL-Go System Files.

Projects will now be build and tested in dependency order.

DEMO: <https://github.com/BusinessCentralApps/buildorder>



# Future investments

Translations

NuGet

Linux, Performance and more

Code Coverage

Page scripting execution

Leapworks

Codespaces

PowerPlatform support



# Translations – Current thinking

Investigate how partners are working with translations today

Create 1-3 supported translation strategies, which fits most needs

no promises  
no dates

# NuGet – Current thinking

Symbols for all AppSource apps will be available from a NuGet feed  
Runtime packages (or full apps) will be available from a NuGet feed  
for partners who opt-in

AL-Go for GitHub will automatically trust the AppSource NuGet server  
AL-Go for GitHub can trust other (partner) NuGet feeds/servers for  
dependency resolution.

NuGet packages can be a single app or a project

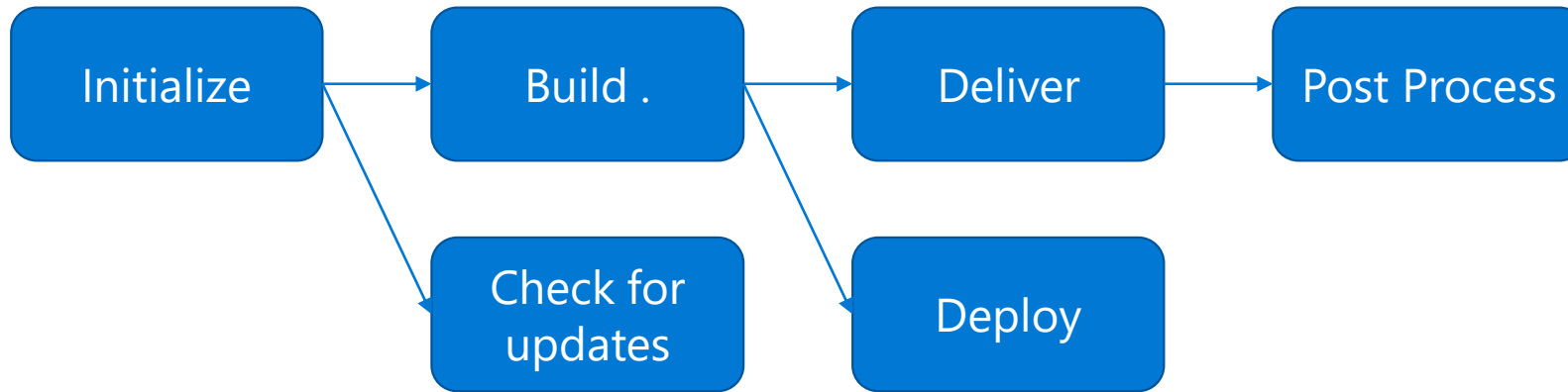
NuGet packages can have dependencies to other NuGet packages  
(from trusted servers)

NuGet packages can contain tests, which also will be run

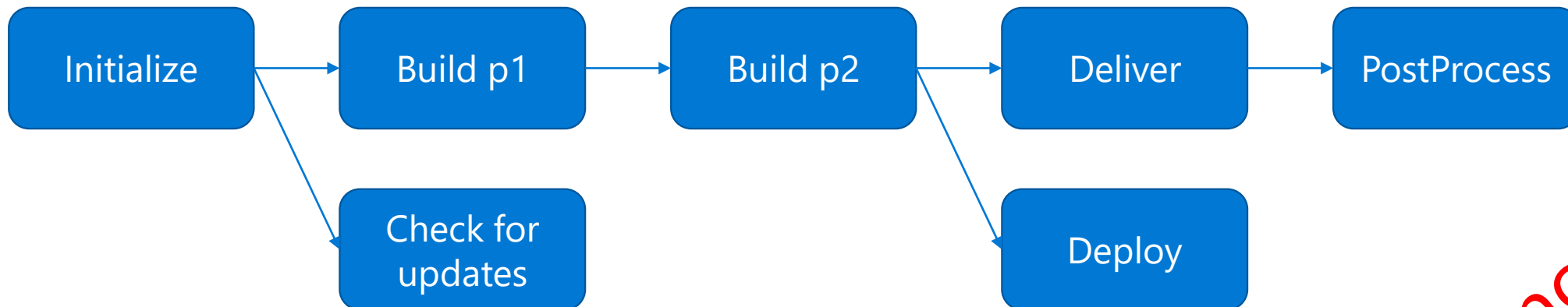
no promises  
no dates

# Linux, performance and more – current thinking

Today, I CI/CD workflow looks like this:



or:

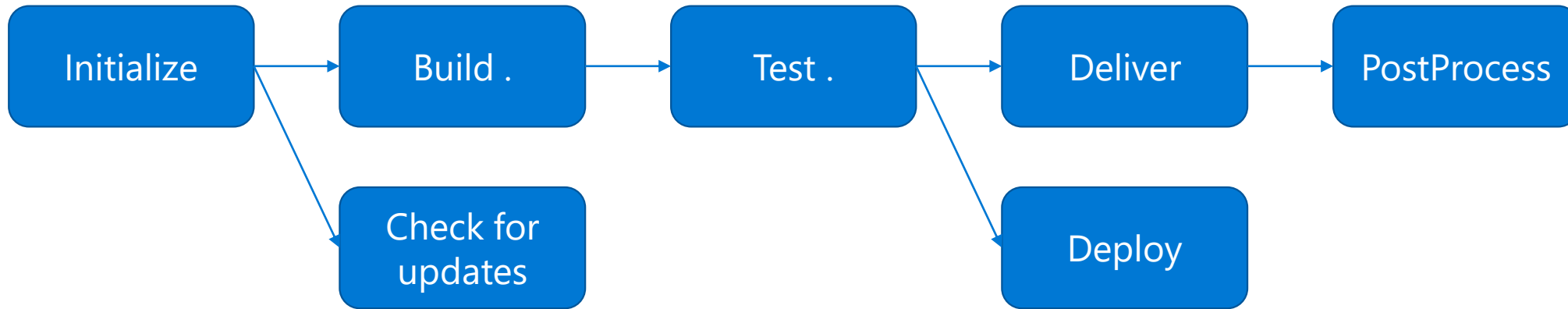


no promises  
no dates

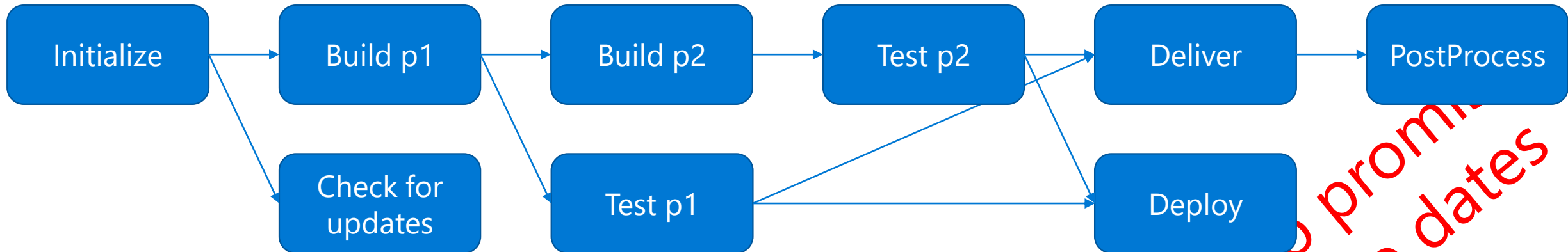


# Linux, performance and more – current thinking

Very soon, the CI/CD workflow will look like this:



or:



no promises  
no dates

# Linux, performance and more – current thinking

Do test environments need to be a docker container in an agent?

No, we will support other test environments, like:

- Cosmo Alpaca build containers
- Azure container instances
- Online sandboxes

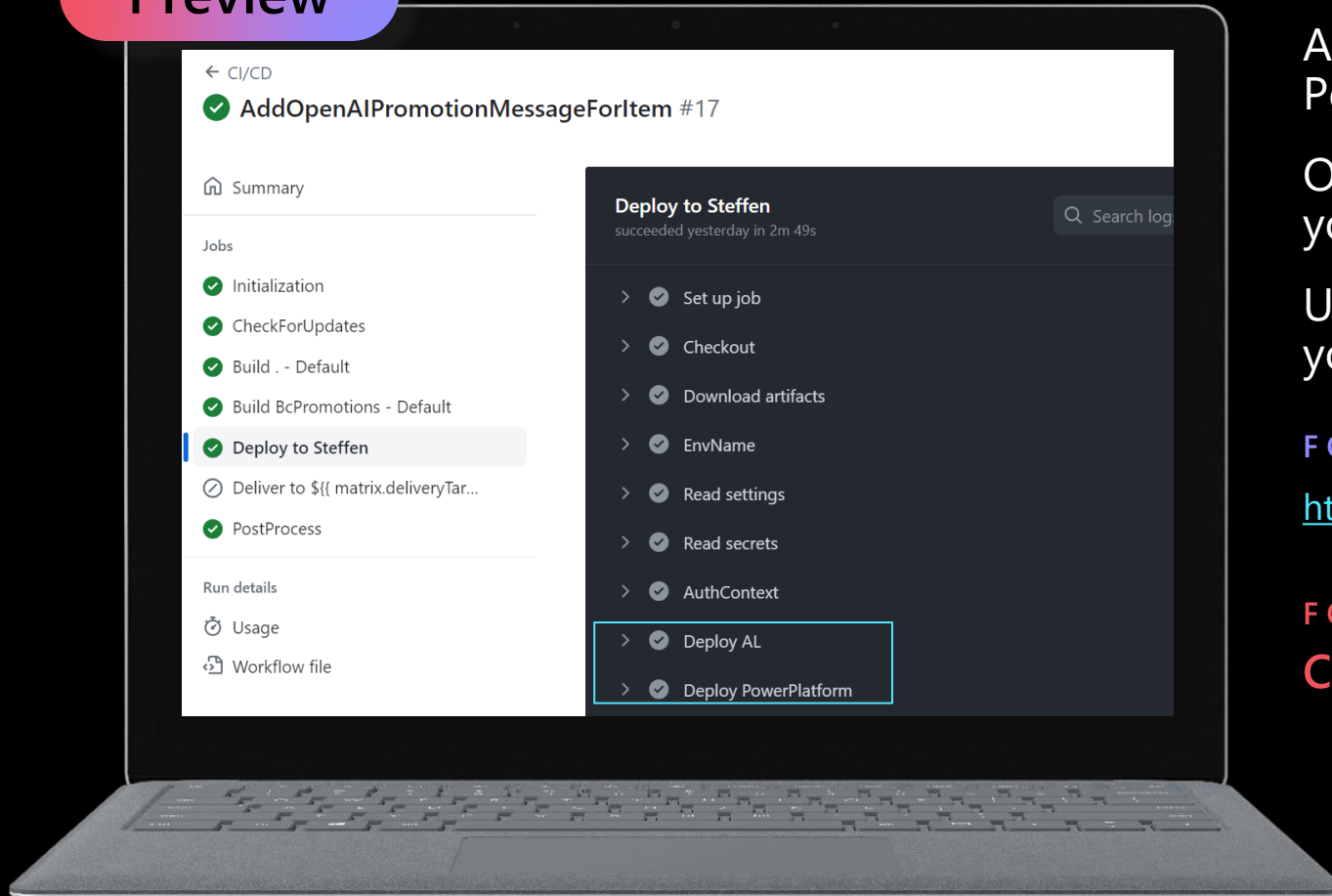
Idea is that the all workflows will be able to run on Linux for better performance and pricing

no promises  
no dates

# AL-Go for GitHub and Power Platform

Full ALM support for Power Platform solutions for Per Tenant Extensions

## Preview



A single repository for your AL and Power Platform artifacts

One set of configuration for all your environments

Unified management and deployment of your solutions across BC and Power Platform

FOR VARS

<https://github.com/microsoft/AL-Go-PTE-PPPreview>

FOR ISVS

Coming soon

Try the preview with your AL and Power Platform code today

**QUESTIONS?**