Progress®

OpenEdge DevOps Framework

Seamlessly build your OpenEdge applications using OpenEdge DevOps Framework (OEDF) and Gradle

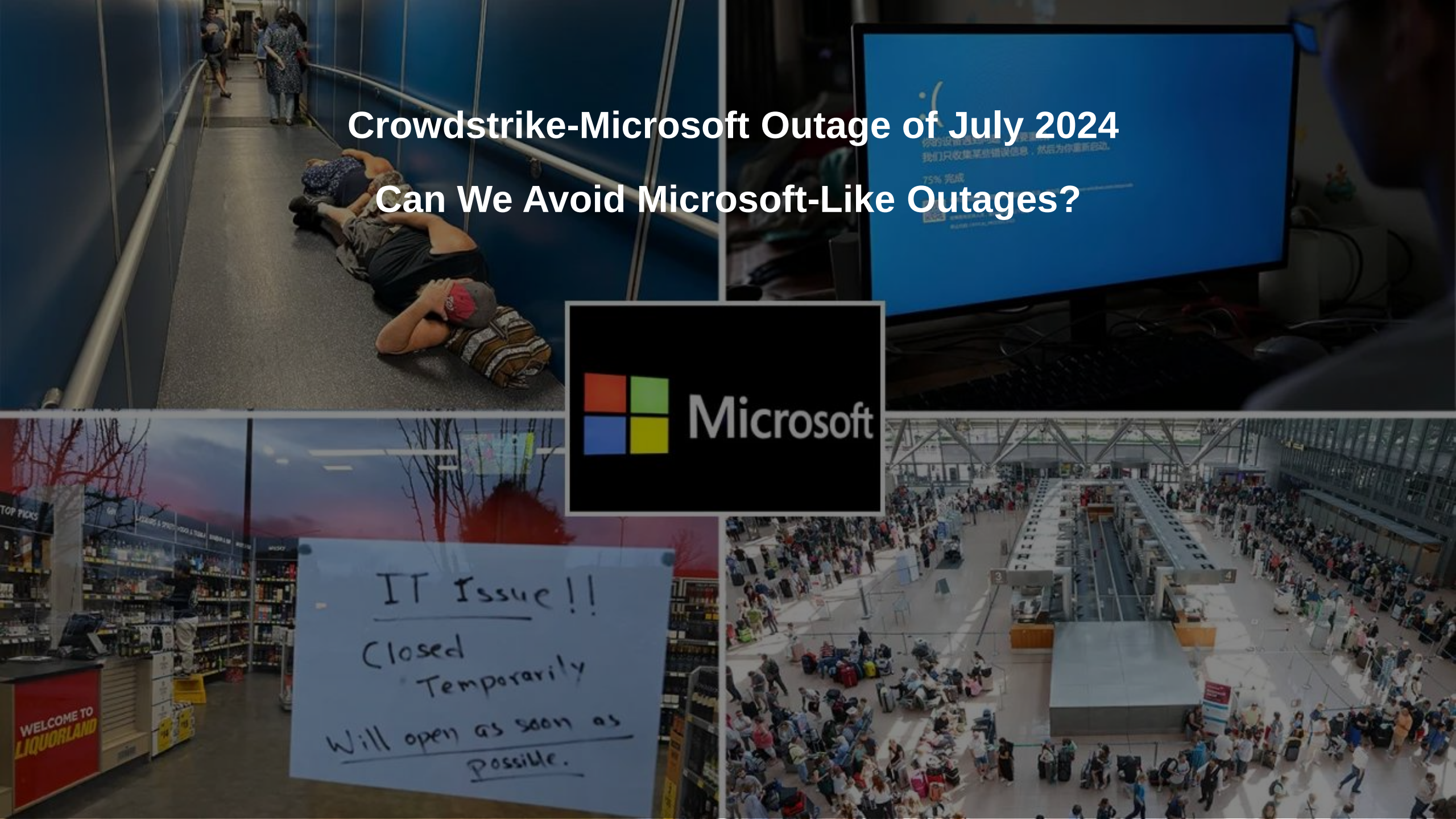
Anil Kotha

Software Engineer, Principal

Oct 2, 2024

Crowdstrike-Microsoft Outage of July 2024

Can We Avoid Microsoft-Like Outages?



Agenda

01
DevOps and CI/CD

02
Overview of OEDF

03
Build with Gradle

04
See it in Action

What is DevOps?

- A methodology in the software development and IT industry consisting of a set of practices and tools
- Integration and automation of work done for software development (Dev) and IT operations (Ops) to improve and shorten a product's development life cycle

Four Key Principles of DevOps

Automation

- Automatic build and deployment process
- Utilize Infrastructure as Code (IaC)
- Reusable on-demand tests

Iteration

- Repeatable build and testing
- “Shift left” to find issues earlier in the process

Monitoring

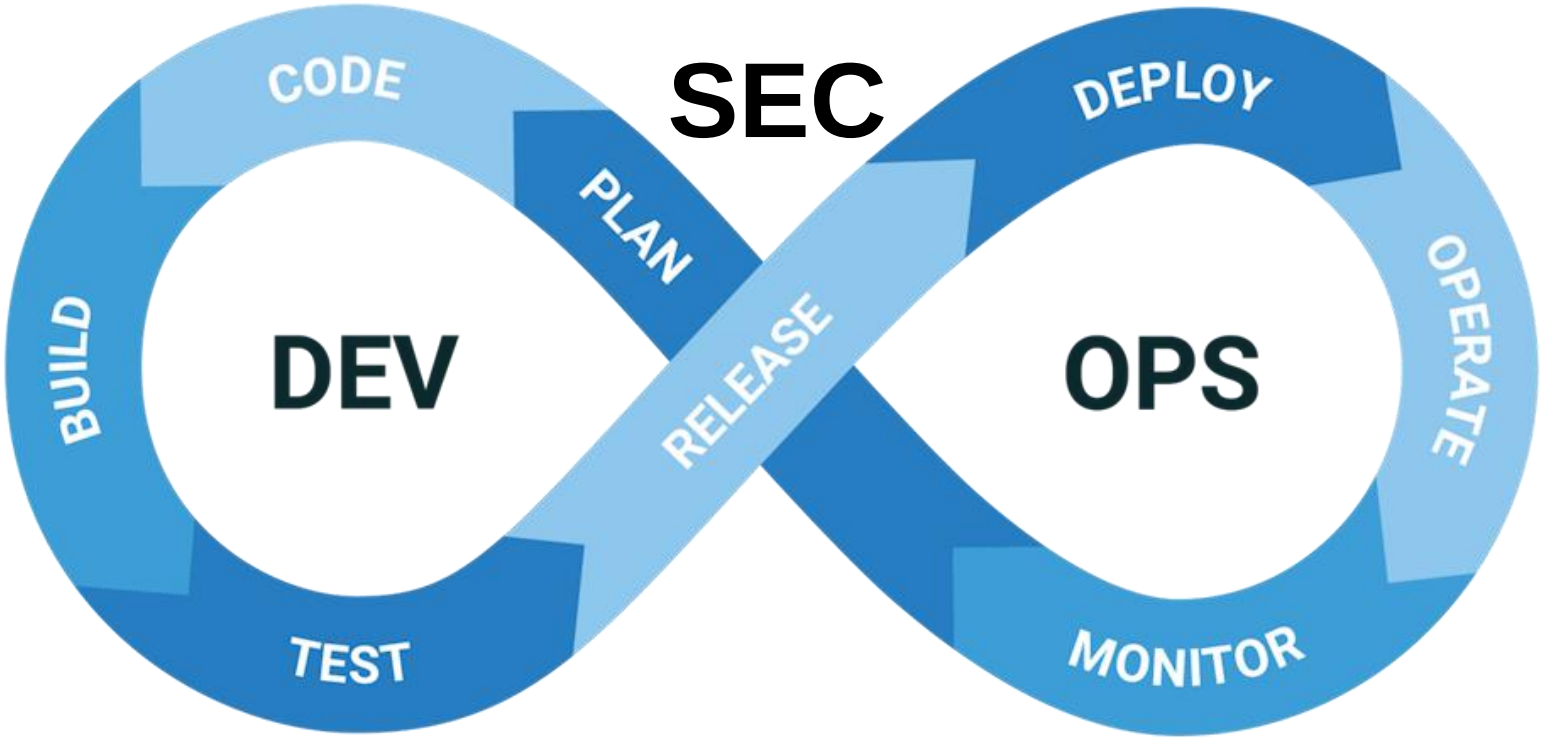
- Continuous feedback and improvement
- Review the metrics
- Shorten feedback loop

Collaboration

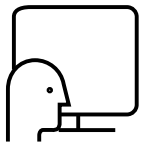
- Dev and operations teams must work together on the process and results
- Provide a positive environment with no “blaming”

DevOps + Security = DevSecOps

Automated Process to Manage Product Lifecycle



Benefits of DevOps

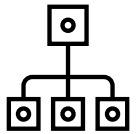


Developers

- Automation allows focus for programming rather than repetitive tasks

Product Managers

- Faster turnaround for features
- Predictable delivery schedule



System Administrators

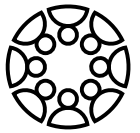
- Repeatable build process
- Infrastructure as Code (IAC)
- Less complex to manage

Test Engineers

- Consistent test results
- Faster resolution with increased visibility

Customer Service

- Fewer bugs, focus on quality
- Integrated into the process

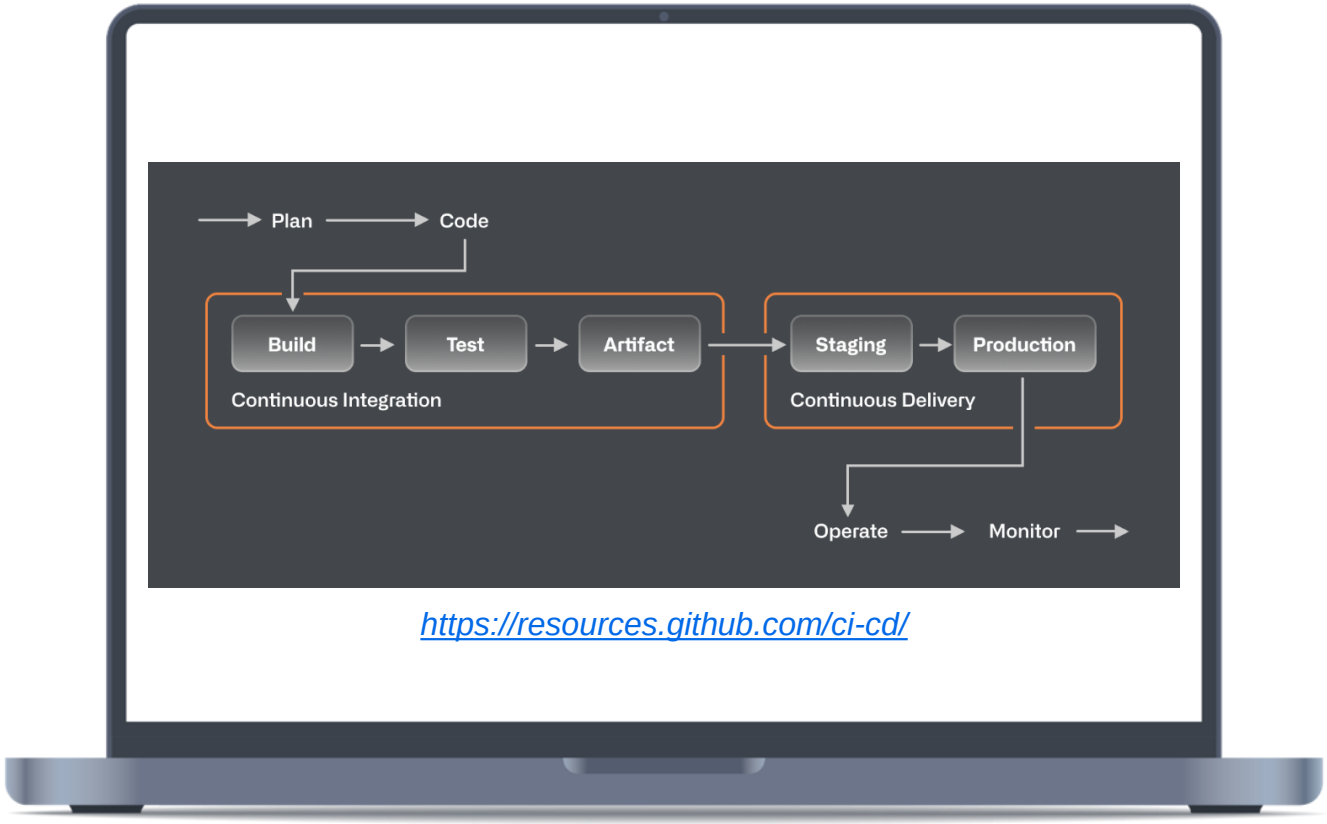


End users

- More stable software
- Consistent deployment dates

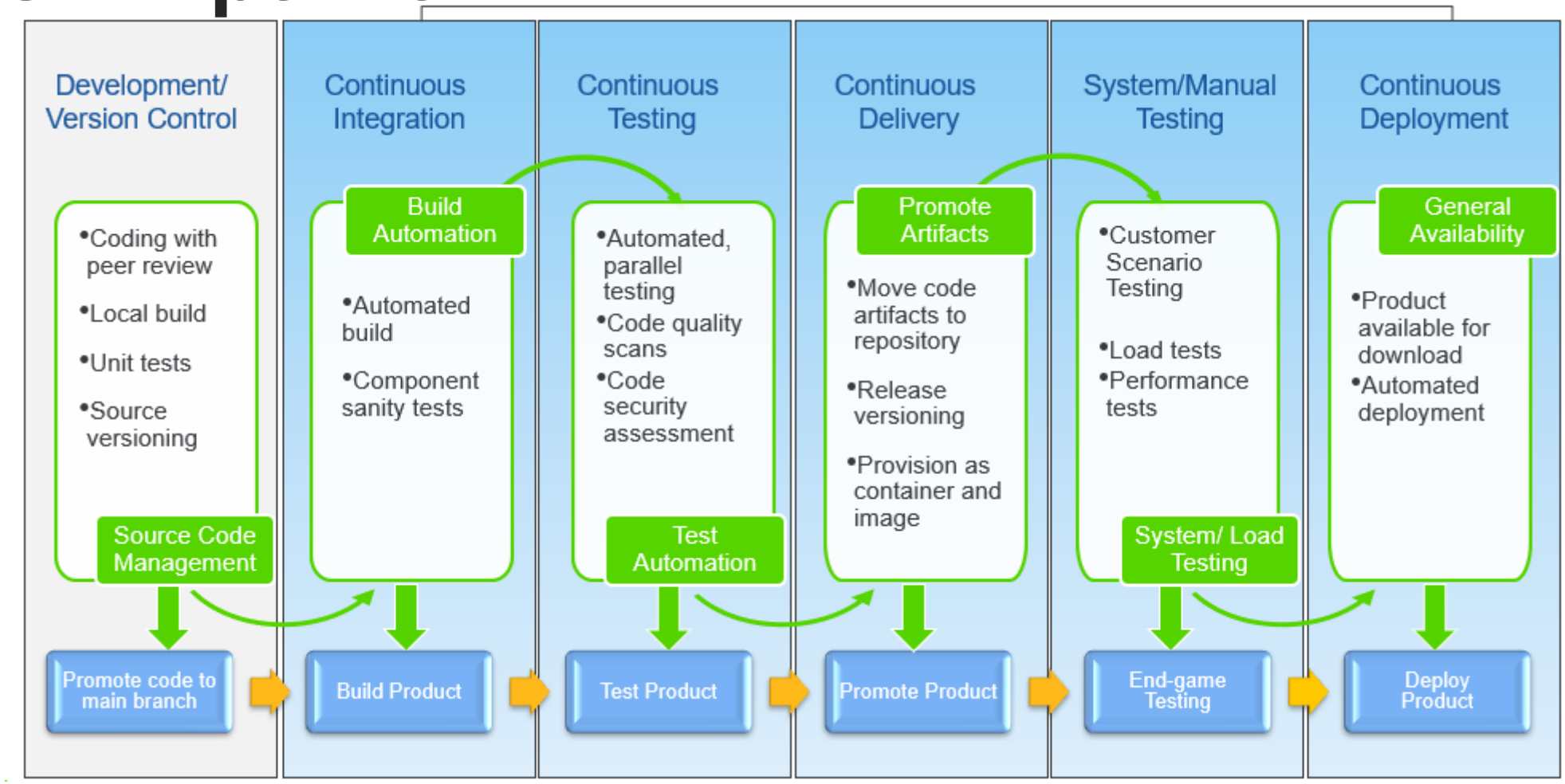
CI / CD Pipeline

- Continuous integration (CI): Automatically builds, tests, and integrates code changes within a shared repository
- Continuous delivery (CD): automatically delivers code changes to production-ready environments for approval
- Continuous deployment (CD): automatically deploys code changes to customers directly



https://resources.github.com/ci-cd/

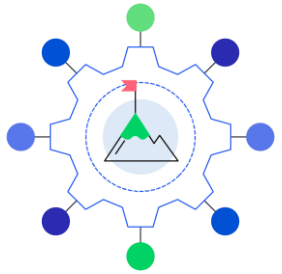
CI/CD Pipeline



Why CI / CD?

- Reduce maintenance cost
- Find and address issues early
- Increase development productivity
- Transition seamlessly from development to integration environment
- Reduce time to market

Speed



Quality

Security

Research suggests that CI/CD can save 20% of your time, EVERYDAY!

https://android.jlelse.eu/how-ci-cd-can-save-your-team-up-to-20-of-time-research-into-75k-builds-674a6306844

Agenda

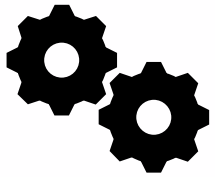
01
DevOps and CI/CD

02
Overview of OEDF

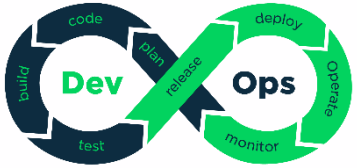
03
Build with Gradle

04
See it in Action

OpenEdge DevOps Framework (OEDF)

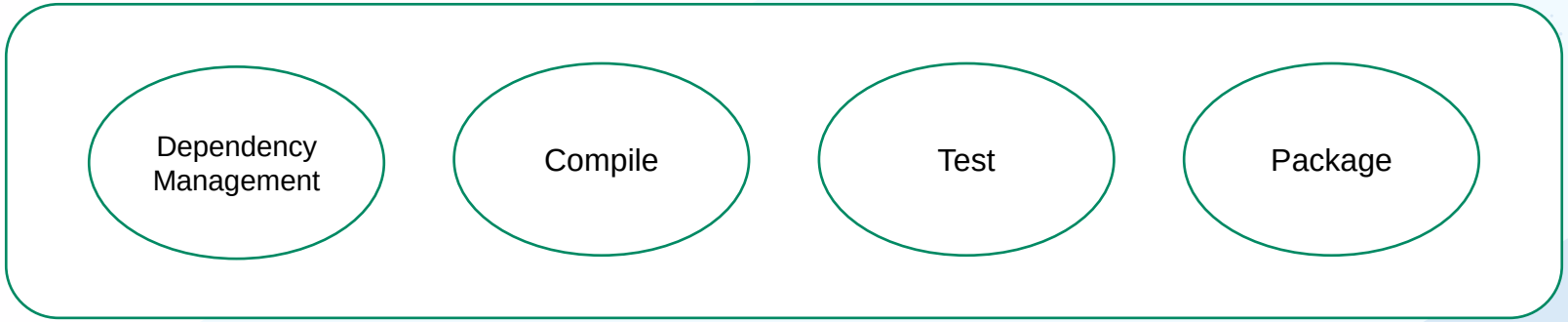


Automation Framework



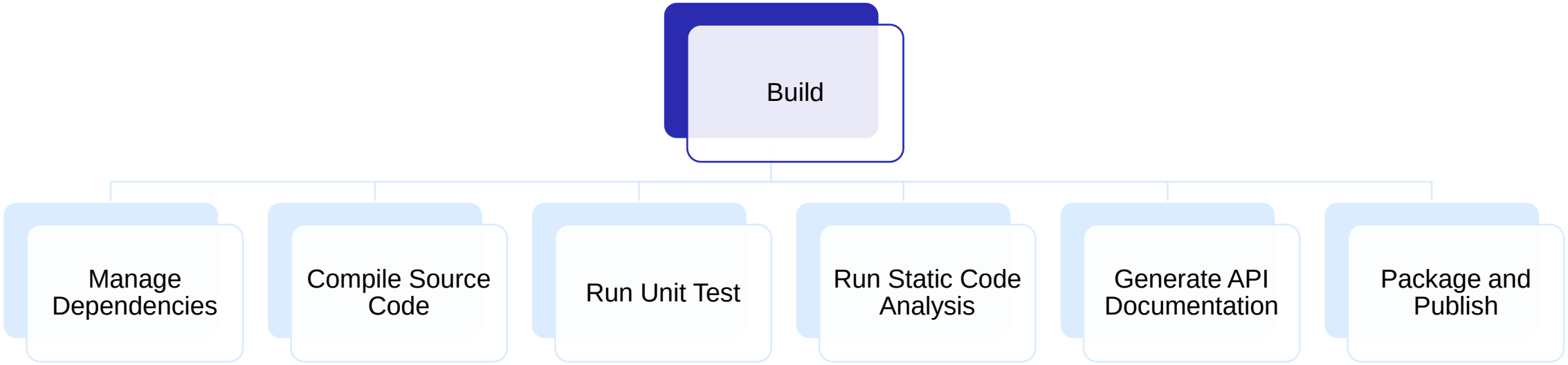
Enables CI/CD

Gradle Plugins



[Learn About the OpenEdge DevOps Framework (progress.com)]

Build process of an ABL application



OpenEdge DevOps Framework

- OEDF designed to help implement a CI pipeline that handles:
  - Repository Integrations
  - Compilation
  - Testing
  - Packaging

OEDF Plugins

Gradle Plugin	ID	Description
ABL Base Plugin	progress.openedge.abl-base	▪ Provides task types (e.g., ABLCompile, ABLUnit, PL, OEWar) ▪ Provides extension: Global settings available to all tasks ▪ Users should write tasks

ABL Main Plugin	progress.openedge.abl	▪ Applies ABL Base Plugin
		▪ Provides default tasks
		  ▪ build
		  ▪ clean
		  ▪ compileAbl
		▪ Mandatory files
		  ▪ Config file
		  ▪ Propath file
		▪ Identifies PDSOE project

Support

- OpenEdge 12.2 onwards
- All OpenEdge supported platforms

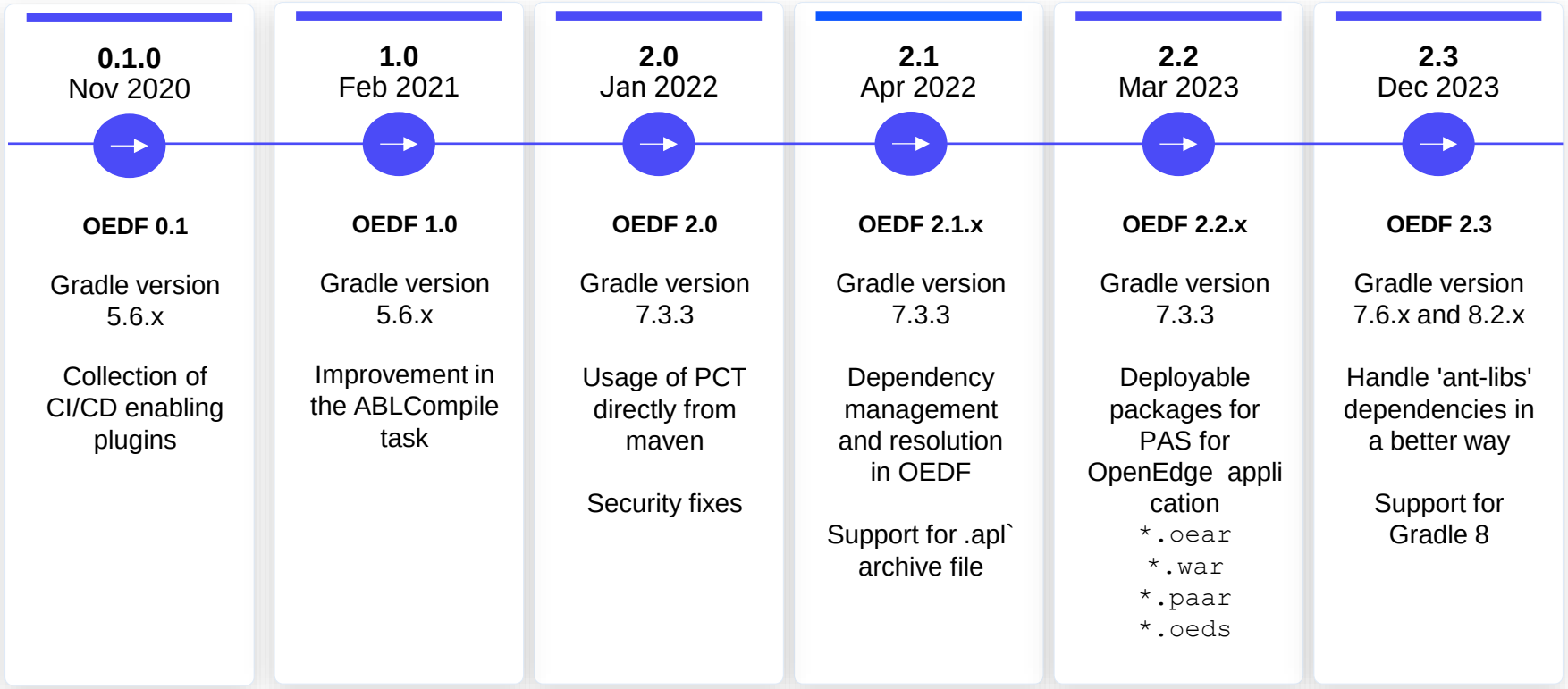
OpenEdge 12.2.x

- Windows
- Linux
- Aix
- Solaris

OpenEdge12.8.x

- Windows
- Linux

OEDF Features - Summary



Progradle

Agenda

01
DevOps and CI/CD

02
Overview of OEDF

03
Build with Gradle

04
See it in Action

Build with Gradle

- Open-source build automation tool
  - Takes your code and packages it into deployable unit
  - Applies to small or large projects
- Uses domain-specific language based on Groovy for project configuration
- Determines which parts of a code base have not changed, builds and executes only the changed parts

Key Gradle Concepts

- Plugins
  - Packages up reusable pieces of build logic
  - Can be used across many different projects and builds
  - Gradle can run custom plugins

Key Gradle Concepts (Cont.)

build.gradle

- Is the Gradle build script file
  - Written in Groovy DSL
  - Lives at the top level of your project

*build.gradle ×

```
12
13 import com.progress.gradle.abl.tasks.*
14
15 plugins {
16     id 'base'                // Gives base ta
17     id 'distribution'        // for zip creat
18     id "com.dorongold.task-tree" // provide
19     id "progress.openedge.abl-base" // base
20     id "progress.openedge.abl"      // link
21 }
```

Task

- Define a unit of work
- See available tasks by running ./gradlew tasks
- Invoked from the command line ./gradlew build
- Tasks have dependencies on other tasks

Key Gradle Concepts (Cont.)

- Wrapper
  - Script used to invoke Gradle and run task
  - Committed into version control
  - Contains a specific version of Gradle for your project
- Properties
- Settings

ABL Base Plugin (progress.openedge.abl-base)

A set of Gradle tasks

- ABLCompile
- ABLRun
- ABLUnit
- BackupDB
- CreateDB
- DBConnection
- ExtractPL
- LoadDBSchema
- PL
- APL
- Oear
- OEWar
- OESvcZip
- Oeds
- Paar

ABL Plugin (progress.openedge.abl)

- Depends on the Progress Developer Studio project structure
- Provides various predefined tasks required to build an ABL project
- Configured in *build.config* file that is part of each Developer Studio project

Software Packaging Options

ABL Libraries

- .pl – procedure library
- .apl – archive library

ABL Application (PASOE)

- .oear – application archive (deploy using ***tcman*** command)

Web Applications (PASOE)

- .war – web application archive (deploy using ***tcman*** command)

Services (PASOE)

- .oeds – data service archive
- .paar – REST service definitions (deploy using ***deployREST*** command)
- .wsm – SOAP service descriptors (deploy using ***deploySOAP*** command)
- .zip – incremental service zip (deploy using ***deploysvc*** command)

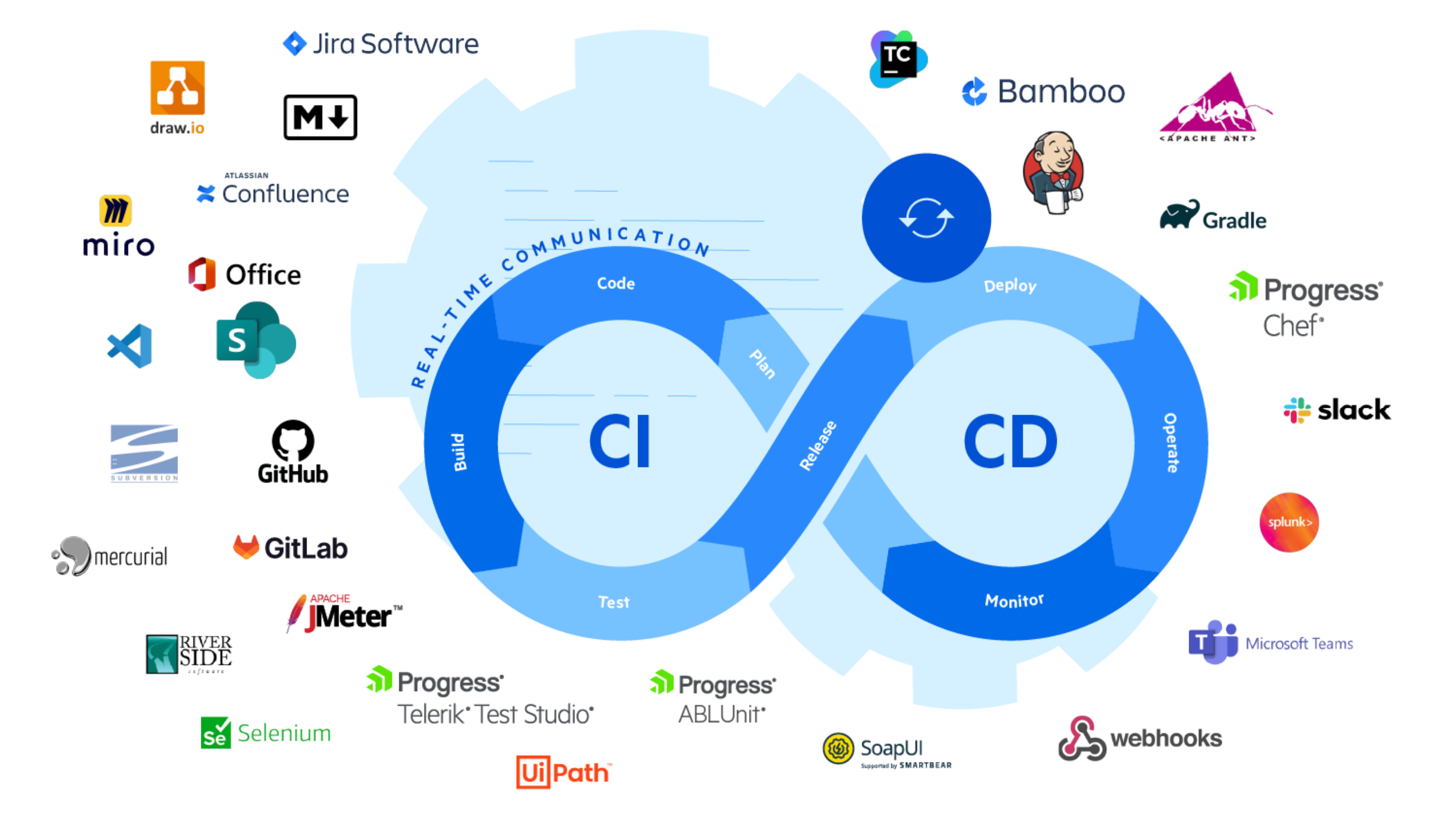
Agenda

01
DevOps and CI/CD

02
Overview of OEDF

03
Build with Gradle

04
See it in Action


Jira Software
draw.io
ATLASSIAN
Confluence
miro
Office
GitHub
GitLab
mercurial
APACHE
JMeter™
RIVER
SIDE
Selenium
Progress
Telerik Test Studio
UiPath
Progress
ABLUnit
SoapUI
Supported by SMARTBEAR
webhooks
Microsoft Teams
splunk>
slack
Progress
Chef
Gradle
Bamboo
<APACHE ANT>
REAL-TIME COMMUNICATION
Code
Plan
Build
Test
CI
Release
Deploy
Operate
Monitor
CD

What Tools Can I Use with OpenEdge?

Repository Integration

GitHub

RoundTable

Subversion

Mercurial

What Tools Can I Use with OpenEdge?

Functional and Non-functional Testing

ABLUnit

Postman

jMeter

Test Studio

UiPath

What Tools Can I Use with OpenEdge?

Build, Package and Deploy – OpenEdge DevOps Framework

Jenkins

Team City

PCT

Ant

Bamboo

Maven™

Gradle / OE Plug-in

Demo

OEDF Prerequisites

- ABL installed with OpenEdge 12.2 and later
- Gradle

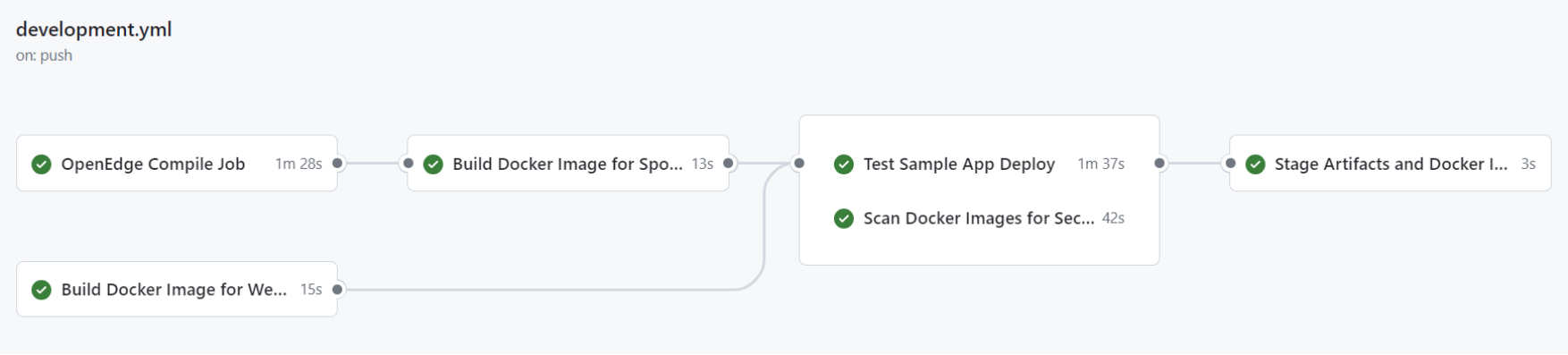
OEDF Setup

- progradle utility

Trigger the build scripts trough a basic CI/CD setup: TeamCity as CI server

- Compile a PASOE project
- Run ABL Unit testing

Example of a CI Workflow



Jobs

- OpenEdge Compile Job
- Build Docker Image for Web UI App
- Build Docker Image for Sports App
- Test Sample App Deploy
- Scan Docker Images for Security Vulne...
- Stage Artifacts and Docker Images for ...
- Test Results - ABL Unit tests for Sports...
- Test Results - Goss tests for Sports App
- Test Results - Goss tests for Web UI App
- Test Results - Sports App Services

CI/CD Best Practices

- Build once, deploy often
- Test, test and test again
- Bundle everything needed for deployment
- Sign, seal and deliver
- Clean up after yourself

Keep things simple!

What did we learn?

- DevOps and CI/CD in general
- OEDF – OpenEdge DevOps Framework
  - OpenEdge implementation of CI/CD
- OEDF plugins
- Gradle concepts
- Gradle tasks for OpenEdge
- Tools to use with OpenEdge

Progress® OpenEdge®

News You Can Use

Resources

- https://github.com/audaciousanil/Build-With-OEDF – Sample ABL example with oedf plugins usage


Progress®