Why we should (not) care about Pipelines!?!

Enabling engineering to continuously build, validate, and deploy secure solutions to delight our customers.

WorkSafeBC

DevSecOps

Said Akram

Developer
@saidakram007

Kevin Schwantje

Security Engineer
@604kev

Willy Schaub

Cloud Systems Engineer
@wpschaub

WHY

Pipeline?

WorkSafeBC

DevOps is the union of people, process, and products to enable continuous delivery of value to our customers.

Donovan Brown,
donovanbrown.com/post/what-is-devops

AGILE

DEVOPS

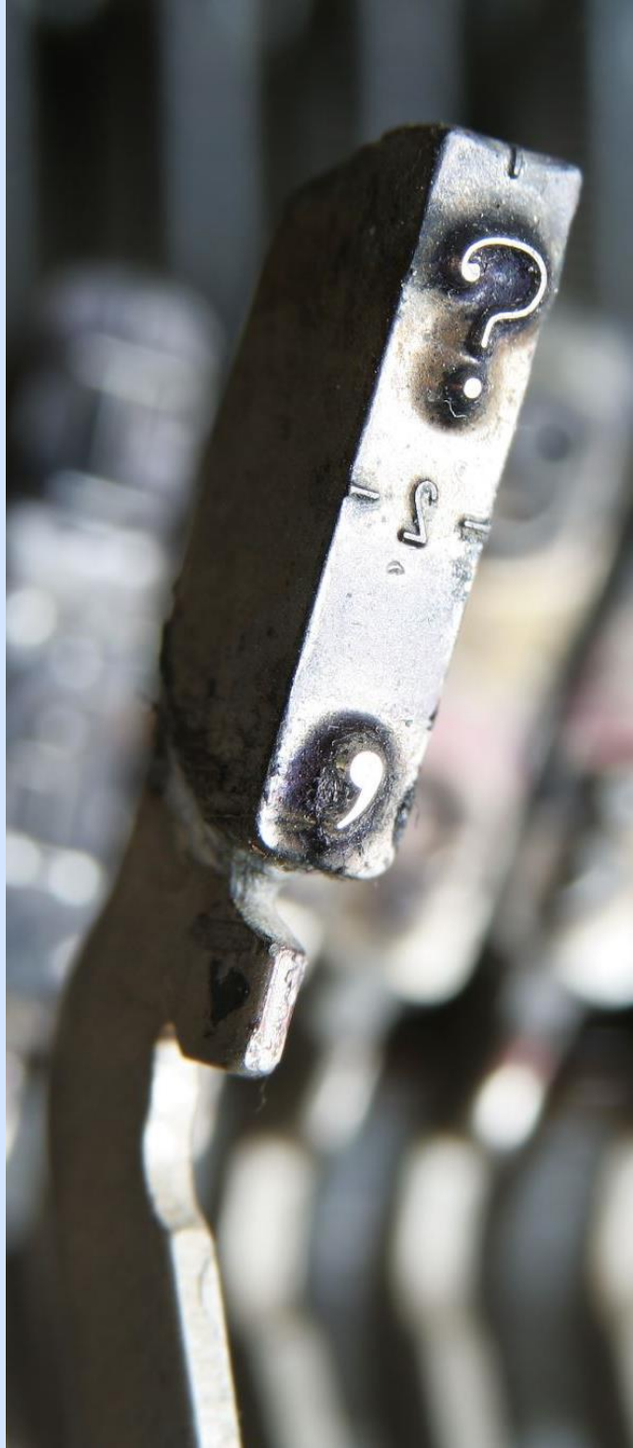
IDEATE | CREATE | RELEASE | OPERATE

WHY checklist

Automate everything automatable

Move repetitive tasks to machines

Enable engineering to focus on delighting customers



Improve beyond the limits of today's processes

Strive to always innovate and improve beyond repeatable processes and frameworks.

No new silos to break down silos

Inspire and share collaboratively instead of becoming a hero or creating a silo.

Feedback from stakeholders is essential

Focus on our stakeholders and their feedback rather than simply changing for the sake of change.

Knowing your customer means cross-organization collaboration

Measure performance across the organization, not just in a line of business.

Inspire adoption through enthusiasm

Promote a culture of learning through Lean quality deliverables, not just tools and automation.

Agents of Chaos
DevOps Core Values

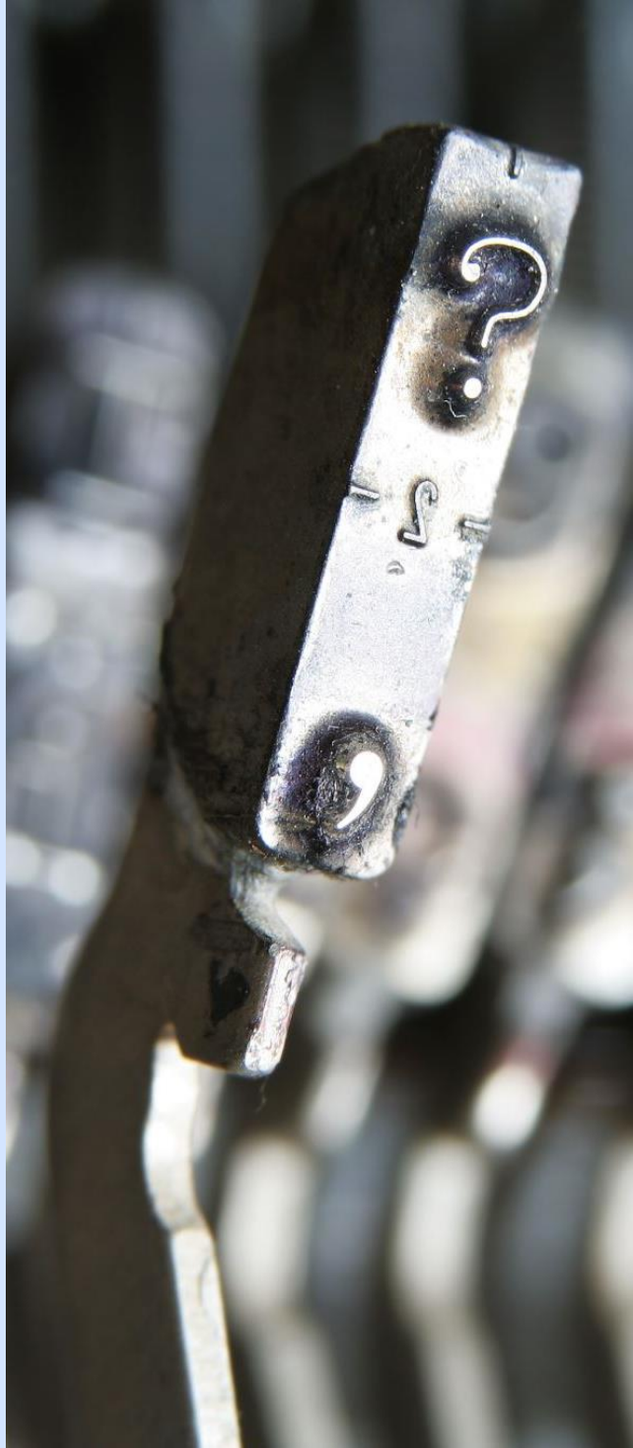
WHY checklist

Automate everything automatable

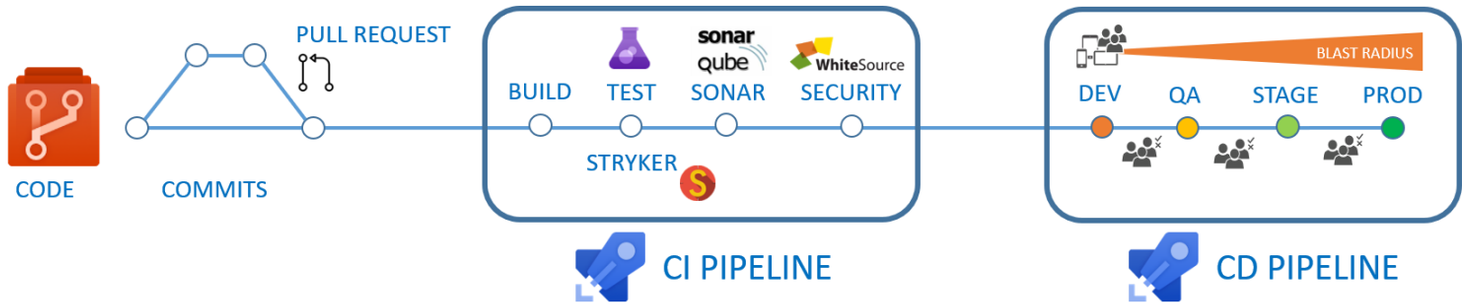
Move repetitive tasks to machines

Enable engineering to focus on delighting customers

Innovate and improve beyond repeatable processes



Unified Pipeline

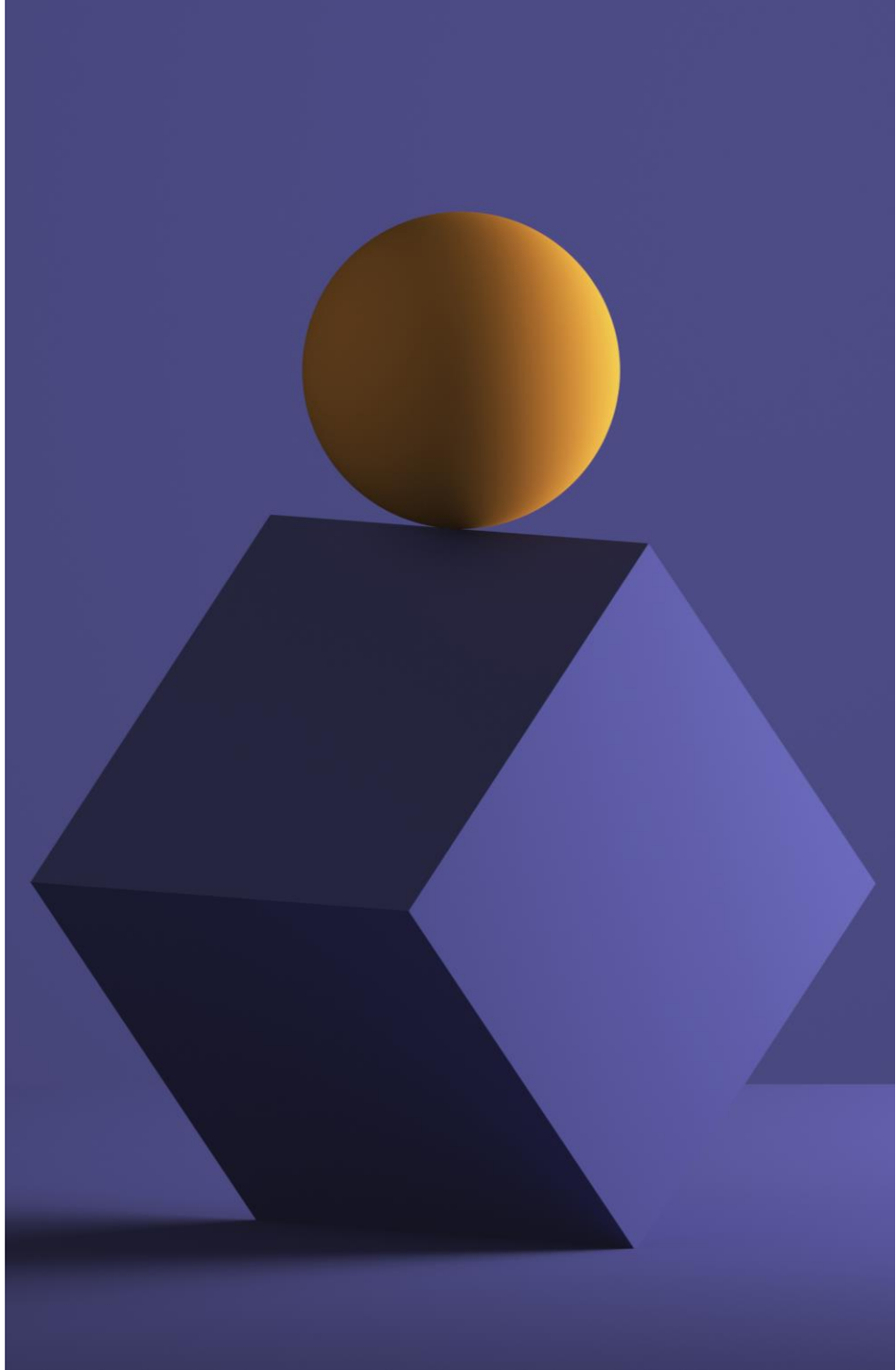


Azure Pipelines

Build once

Deploy the same build artifacts

Streamlined approvals



3000+ pipeline definitions

Snowflakes

Vulnerabilities

Classic warning

WHY checklist

Automate everything automatable

Move repetitive tasks to machines

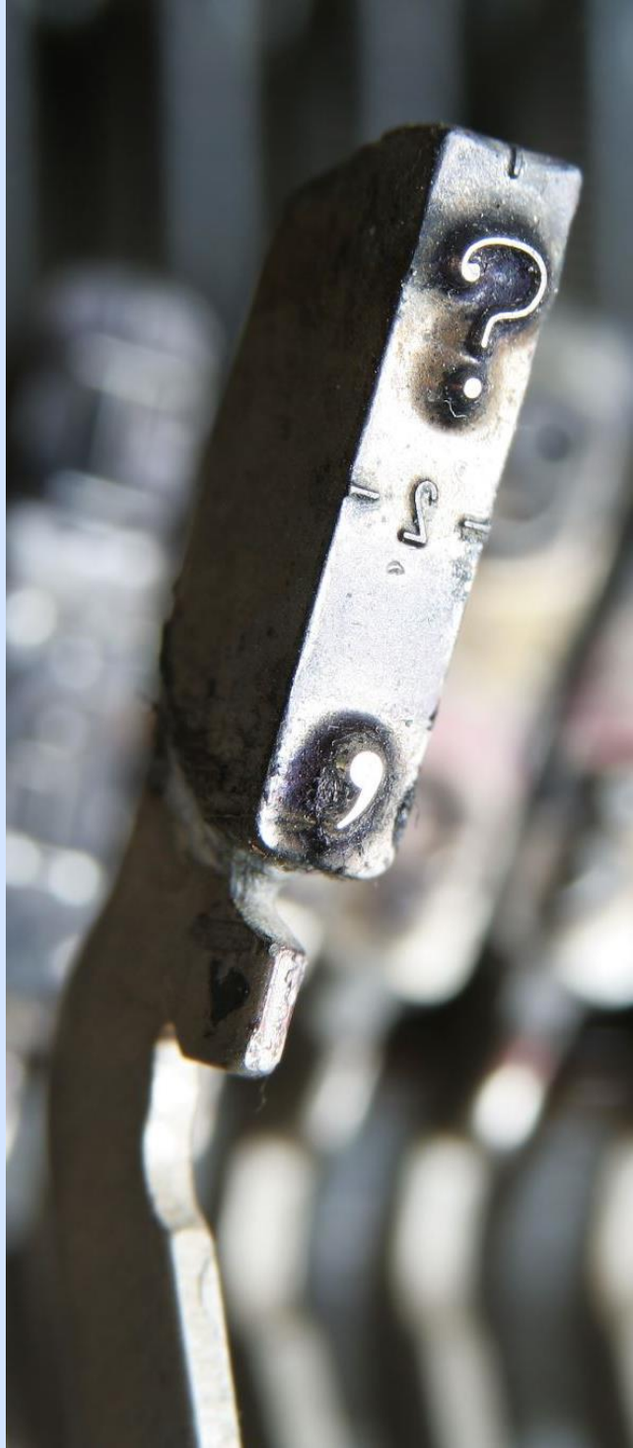
Enable engineering to focus on delighting customers

Innovate and improve beyond repeatable processes

Alignment, Consistency, Simplicity, and Security guardrails

Flexibility and ability to “keep up with technology”

Transparency over standardization

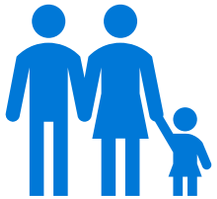


WHAT

Did we do?

WorkSafeBC

YAML

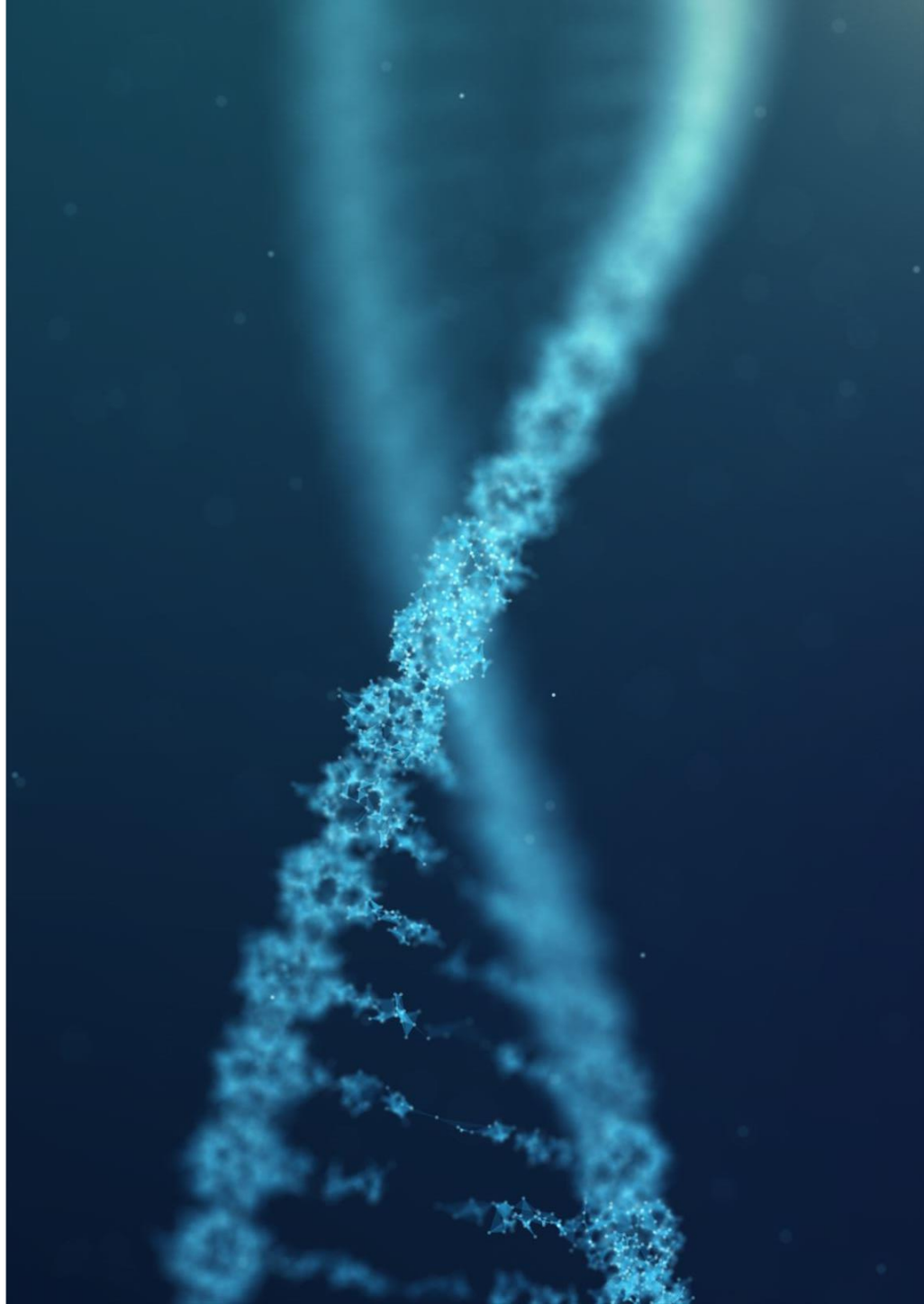


-readable language

Clark Evans 2001

Yet another markup language

GitHub fork of AzDO pipelines



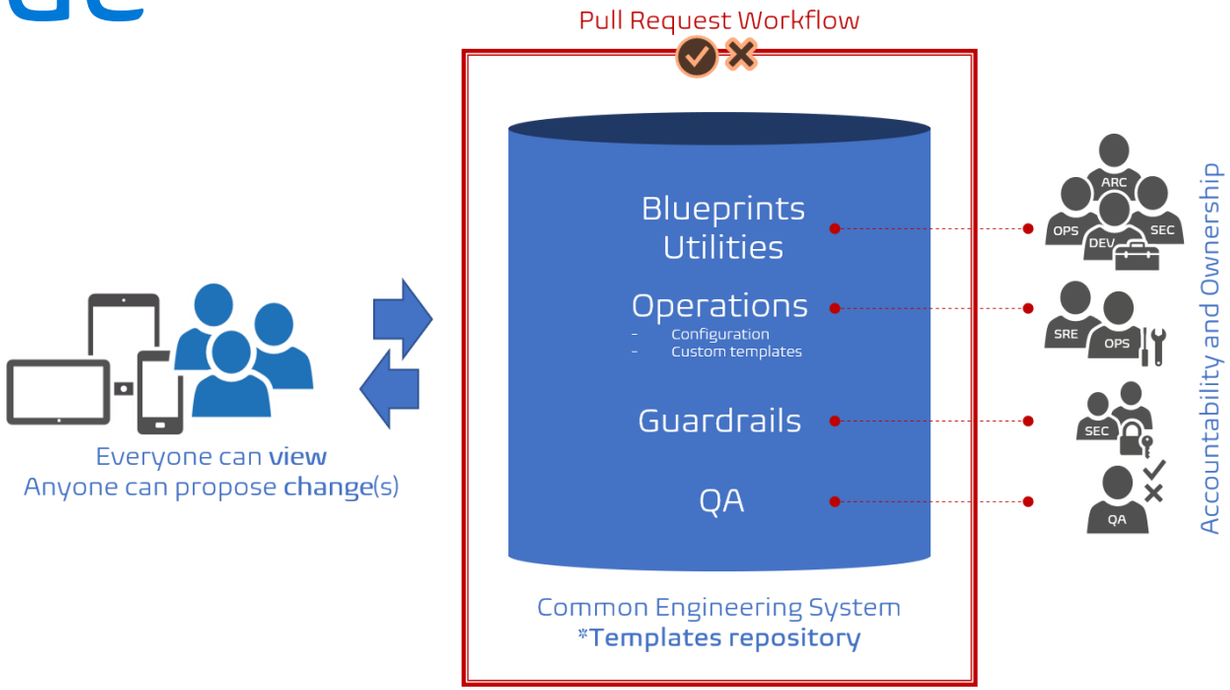
Pipeline as code

Golden fleece

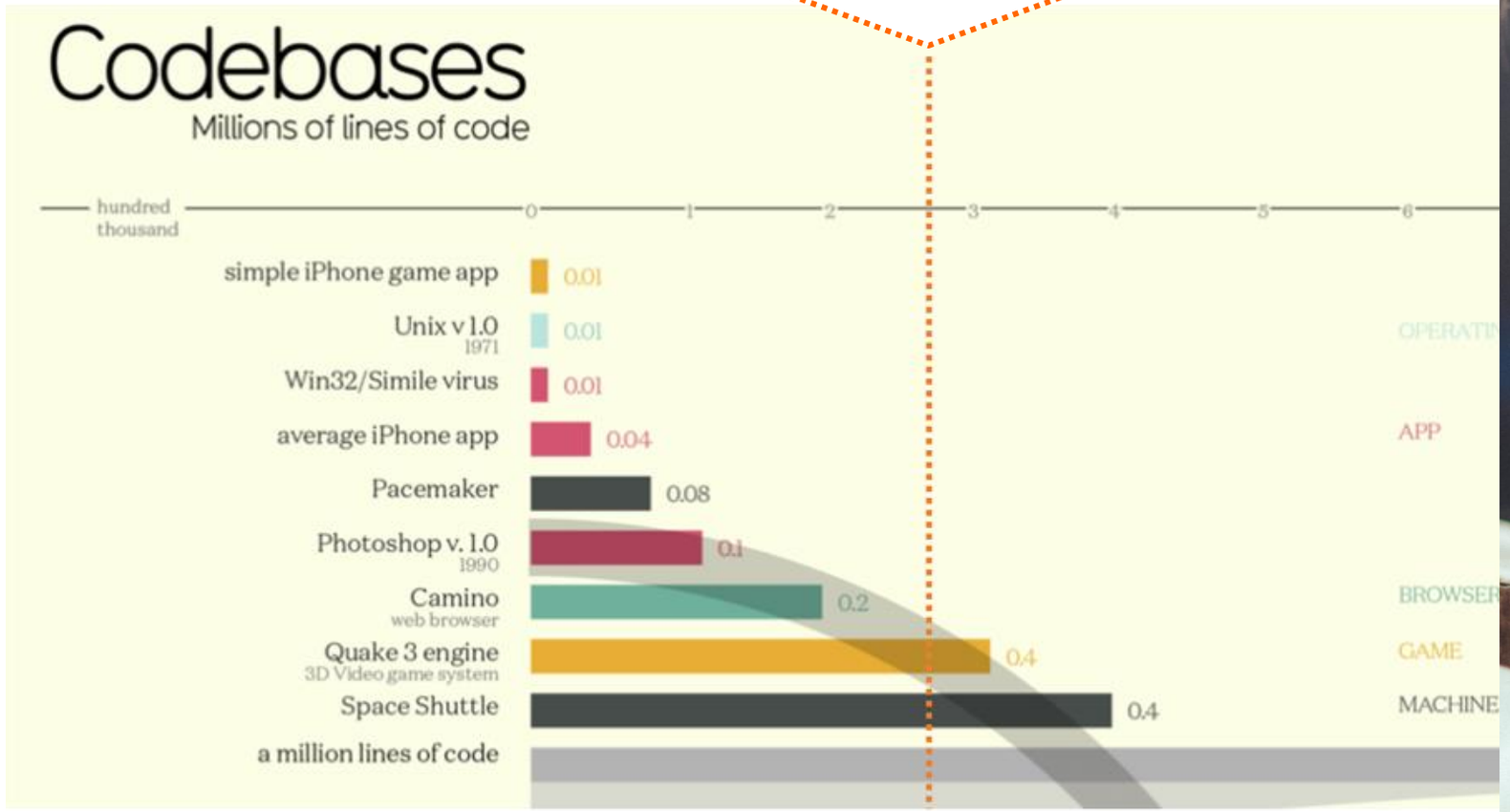
Transparency

Everyone can contribute

Automation



927 * 300 = 278,100



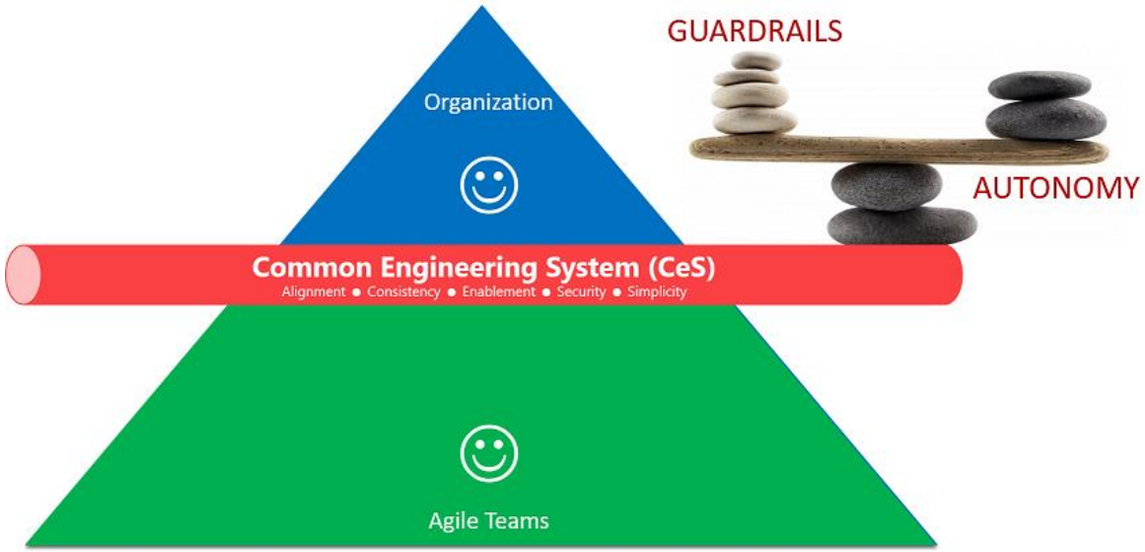
https://datavizblog.com/2017/02/21/infographic-codebases-millons-of-lines-of-code/

113,094

25,956 ▼ ... 90.67% less waste!

Re-usable Templates

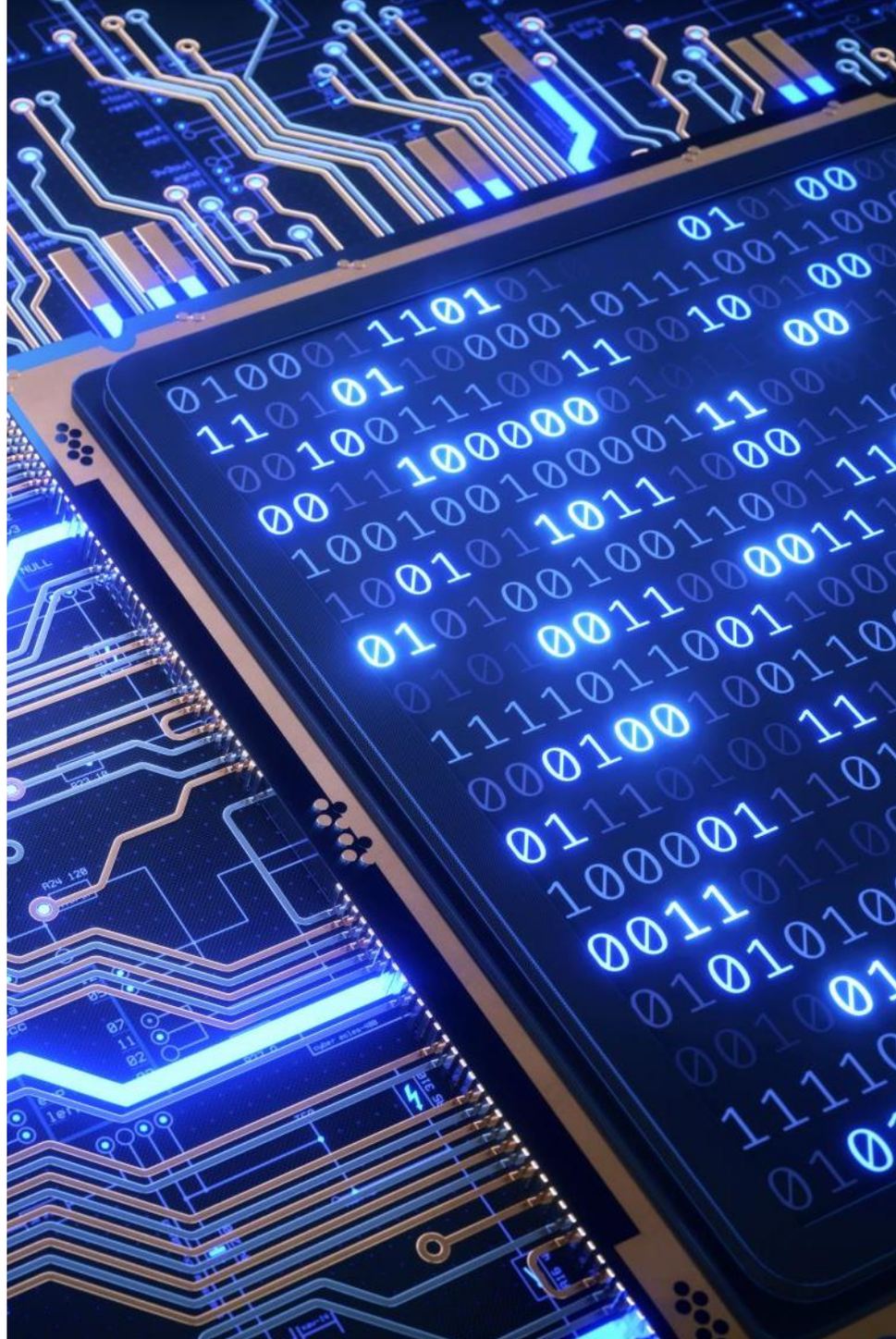
Parts 3, 4, 5



SonarQube, WhiteSource

Building Code

Telemetry

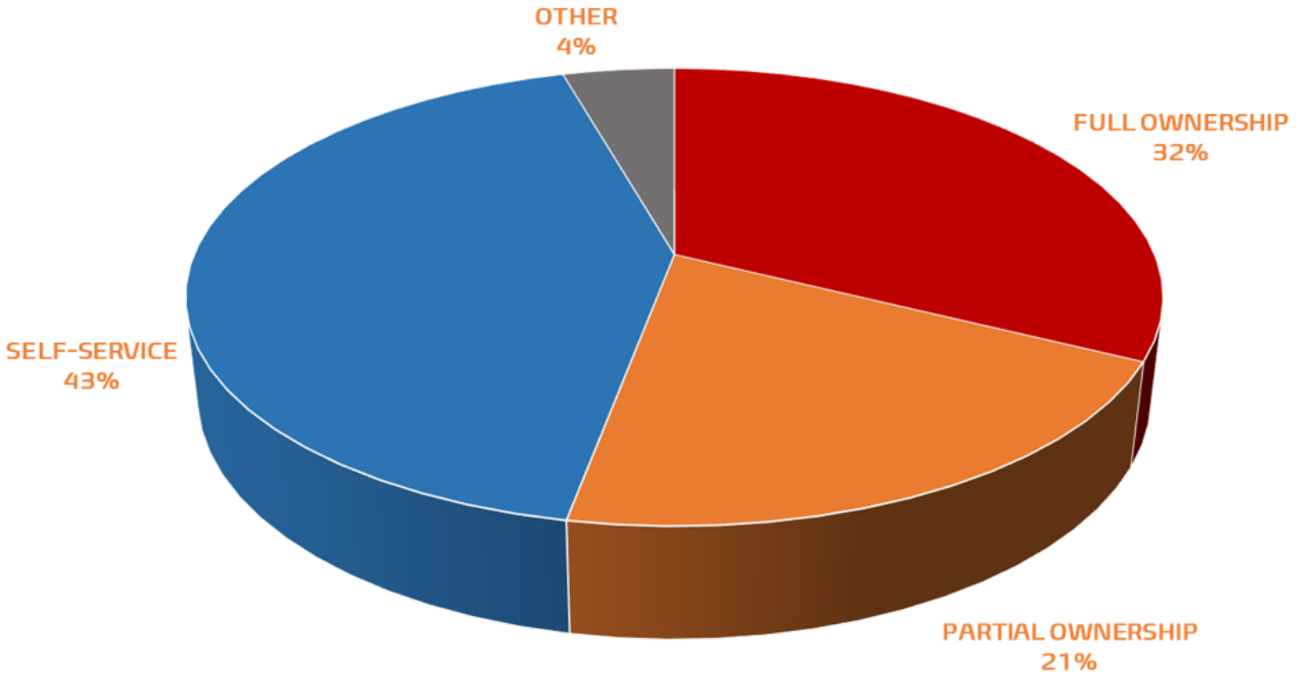


1st Gen App-Type Blueprints

Part 7

Abstract 90%+ of CI pipe/code

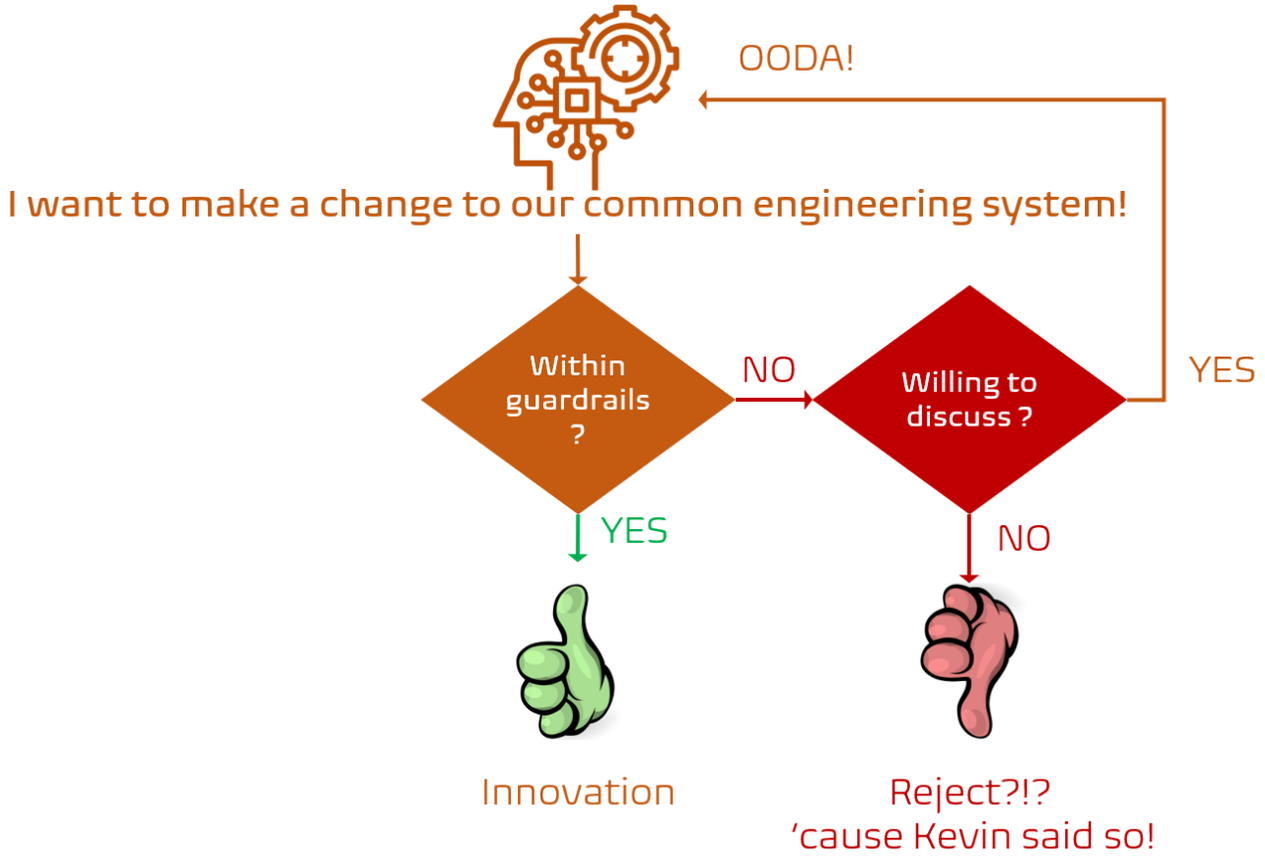
Extend template



2nd Gen App-Type Blueprints

Part 8

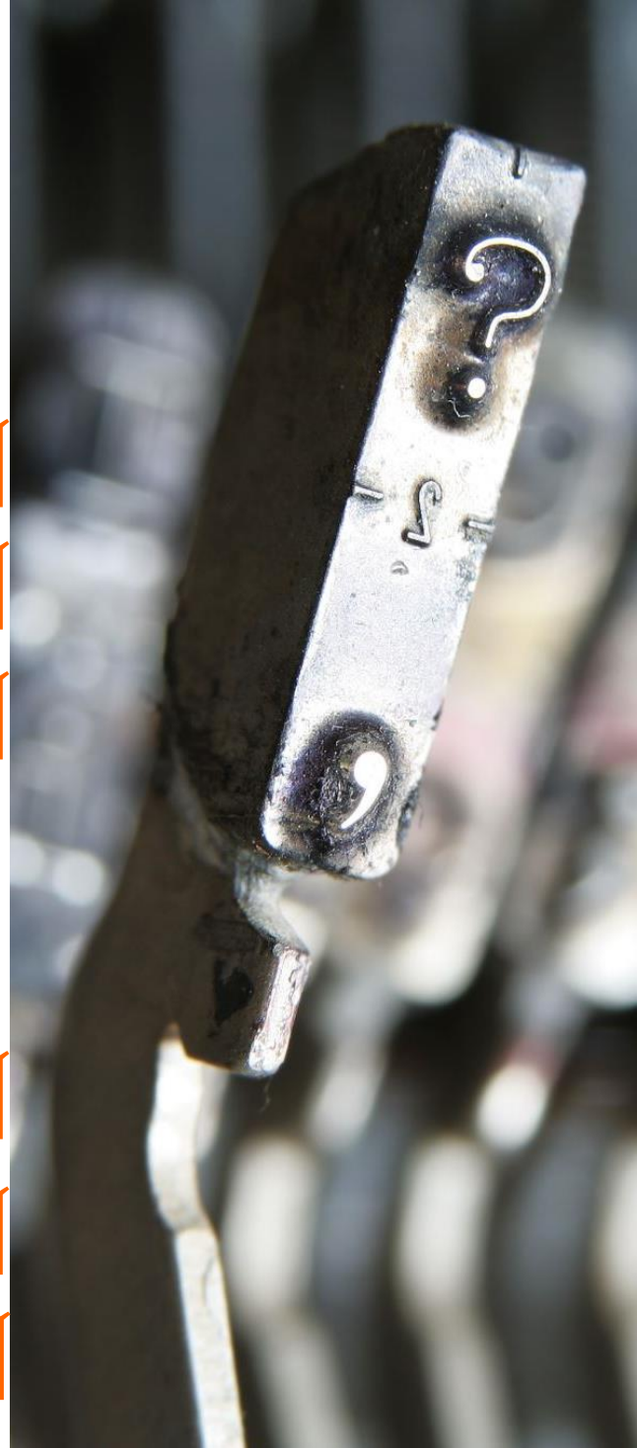
Abstract 90%+ of CI/CD pipe/code



WHY checklist

- Automate everything automatable ☐
- Move repetitive tasks to machines ☑
- Enable engineering to focus on delighting customers ☑
- Innovate and improve beyond repeatable processes ☑

- Alignment, Consistency, Simplicity, & Security guardrails ☑
- Flexibility and ability to “keep up with technology” ☑
- Transparency over standardization ☑



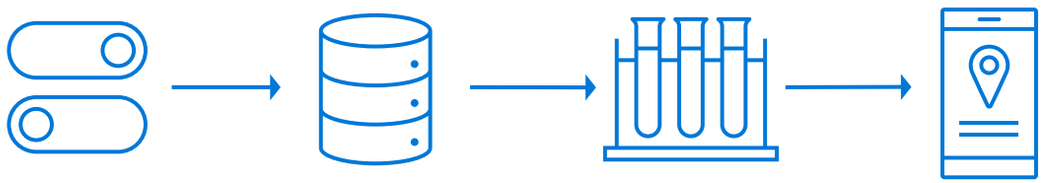
Self-Service

Blueprints are enablers

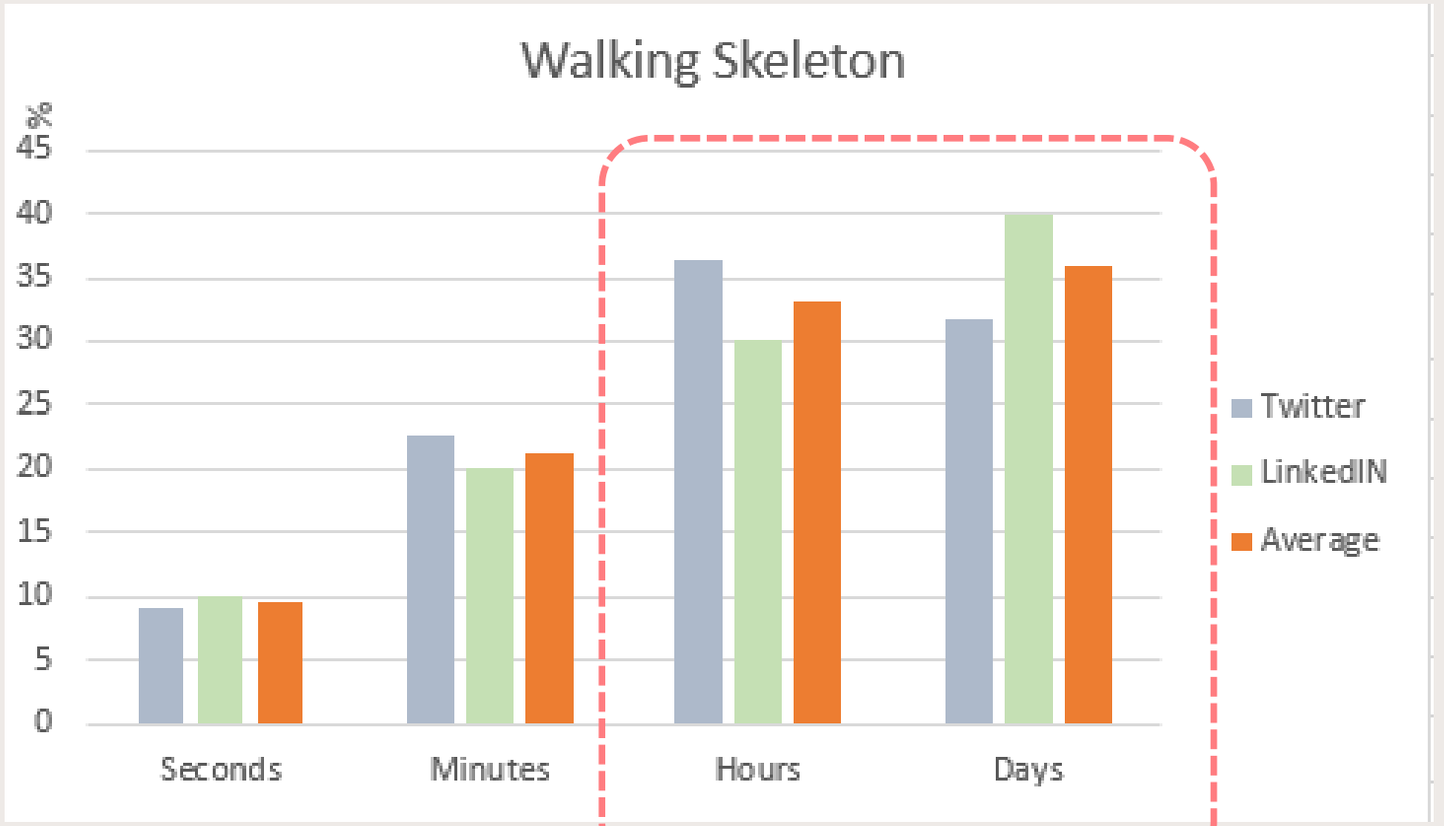
Consistency is the key

Simplicity makes it a reality

Automation working group

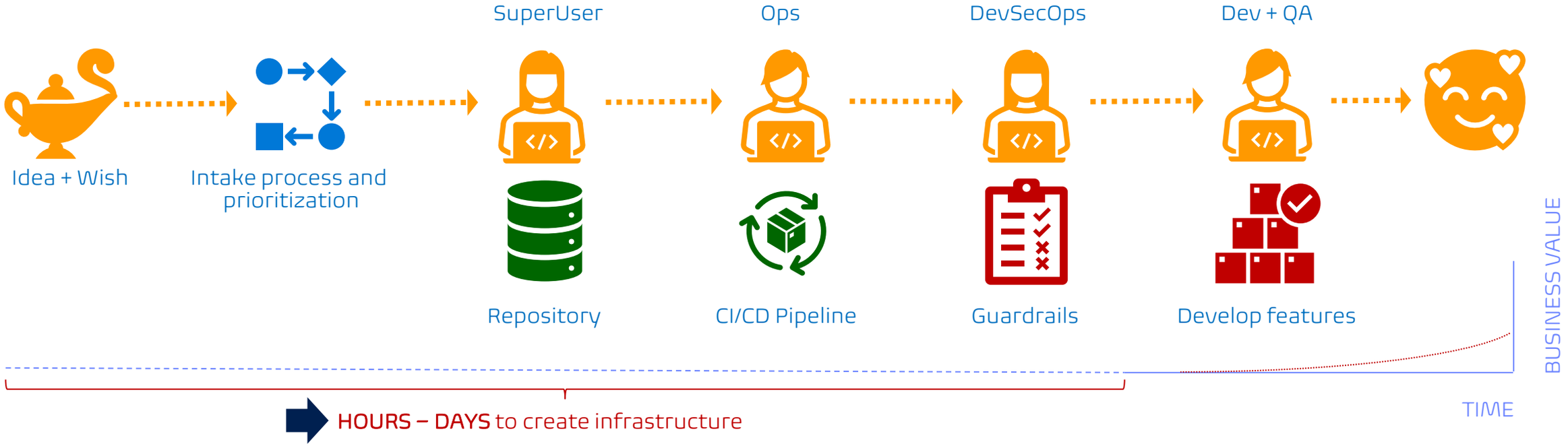


Click • <1min • Hello World

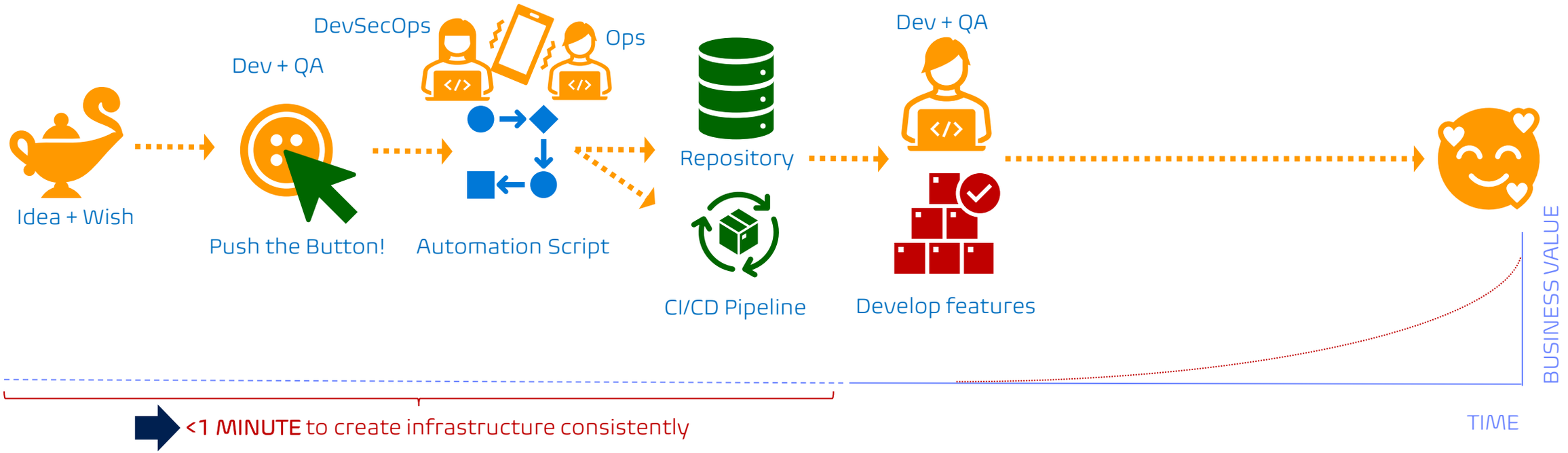


UNACCEPTABLE!

Manual


SuperUser
Ops
DevSecOps
Dev + QA
Idea + Wish
Intake process and prioritization
Repository
CI/CD Pipeline
Guardrails
Develop features
BUSINESS VALUE
TIME
HOURS – DAYS to create infrastructure

Auto


DevSecOps
Ops
Dev + QA
Dev + QA
Idea + Wish
Push the Button!
Automation Script
Repository
CI/CD Pipeline
Develop features
BUSINESS VALUE
TIME
<1 MINUTE to create infrastructure consistently

Parameters
You can specify parameters in templates and in the pipeline, as shown.
They can be restricted to a subset of values.
- key:value pairs
- string : any
- ${{}} to dereference @ **queue** time

Triggers
Use triggers to run a pipeline automatically.
- Continuous integration (CI) trigger
- Pull request validation (PR) trigger
- Comment triggers (GitHub only)
- Scheduled triggers
- Pipeline triggers (YAML)
- Build completion triggers (Classic Build)

Variables
Variables are different from runtime parameters, which are typed and available during template parsing.
- key:value pairs
  - string : string
  - $(key) to dereference @ **run** time
- Inline variables
  - hardcoded
- Variable groups
- Template variables
- Pipeline variables
- Predefined variables
  - $(Build.SourceBranch)
  - https://docs.microsoft.com/en-us/azure/devops/pipelines/build/variables

```yaml
variables:
  name: WSBC

steps:
- script: echo "Hello $(name)!"
```

Variables switch to **key:value** if mixing with groups, templates

```yaml
variables:
# a regular variable
- name: myvariable
  value: myvalue
# a variable group
- group: myvariablegroup
# a reference to a variable template
- template: myvariabletemplate.yml
```

Resources
Resource is anything used by a pipeline that lives outside the pipeline.
- CI/CD pipelines that produce artifacts (Azure Pipelines, Jenkins, etc.)
- Code repositories (Azure Repos Git repos, GitHub, Bitbucket Cloud)
- Container image registries (Azure Container Registry, Docker Hub, etc.)
- Package feeds (GitHub packages)

Templates
Define reusable content, logic, and parameters.
- Variable template
- Logic template

YAML Generic Blueprint-based Pipeline Quick Reference

WORK SAFE BC
WORKING TO MAKE A DIFFERENCE

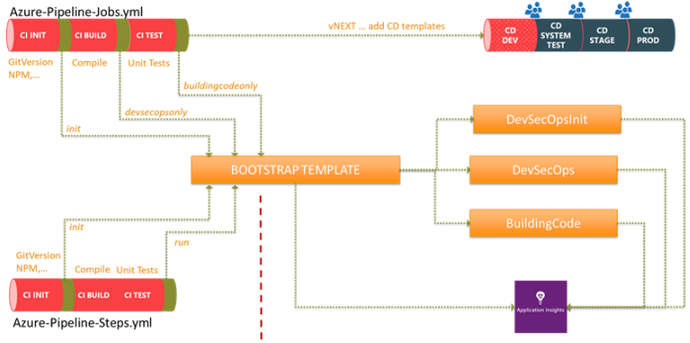
2021.04.30 v1.2 WS - See https://aka.ms/yaml for Azure DevOps Pipeline documentation

azure-pipeline-single-job **Blueprint** Extract

```yaml
# ----------------------------------------------------------------------------------------------------
# PIPELINE INITIALIZATION
# ----------------------------------------------------------------------------------------------------
# batch    - Batch if pipeline is running.
# '*'      - Trigger on all branches. Artifact filters will be used in CD.
# paths    - Exclude pipeline/ folder which contains pipeline yml files.
# ----------------------------------------------------------------------------------------------------
parameters:
- name: image
  displayName: 'Pool Image'
  type: string
  default: windows-latest
  values:
  - windows-latest
  - ubuntu-latest
  - macOS-latest

trigger:
  batch: true
  branches:
    include:
    - '*'
  paths:
    exclude:
    - pipeline

# Semantic version as per Azure DevOps Naming Conventions.
name:
$(portfolioName)_$(productName)_$(GITVERSION_MAJORMINORPATCH)_$(date:yyyyMMdd).$(date:HHmmss).$(Build.SourceBranchName)

# Configure the default agent pool and image to use for your pipeline
pool:
  vmImage:              $(image)

# VARIABLES
variables:
  BuildConfiguration:   Release
  BuildPlatform:        'Any CPU'
  templateVersion:      1.0.7
  portfolioName:        'TODO REPLACE WITH PORTFOLIO NAME'
  productName:          'TODO REPLACE WITH PRODUCT NAME'
  productGuid:          'TODO REPLACE WITH A NEW GUID WITHOUT BRACKETS'

#Repository resources
resources:
  repositories:
    # ==========================================================================
    # SKULL & CROSS-BONES - DO NOT COMMENT OUT, OR REMOVE THIS REPOSITORY
    - repository: CDTemplates
      type: git
      name: 'Common-Engineering-System/AzureDevOps.Automation.Pipeline.Templates'
    # ==========================================================================

# ----------------------------------------------------------------------------------------------------
# START OF BUILD and TEST STAGE
# - GitVersion task looks at your Git history and works out the semantic
#   version of the commit being built.
# ----------------------------------------------------------------------------------------------------
stages:
- stage: ContinuousIntegration
  displayName: Continuous Integration
  jobs:
  - job: ContinuousIntegration
    steps:
    - task: GitVersion@5
      inputs:
        runtime: core
        updateAssemblyInfo: false

# ----------------------------------------------------------------------------------------------------
# PREREQUISITES
# - Run jobs/steps that have to run before the build here, for example NuGet
# ----------------------------------------------------------------------------------------------------
#   TODO Insert your scripts, steps, and tasks here and remove this comment

# ==========================================================================
# BOOTSTRAP VALIDATION, STAGE: CI_BOOTSTRAP_INIT
# SKULL & CROSS-BONES - DO NOT COMMENT OUT, OR REMOVE THIS SECTION
# ==========================================================================
    - template: Templates/Bootstrap.yml@CDTemplates
      parameters:
        bootstrapMode:    'init'
        applicationType:  'TODO REPLACE WITH APP TYPE' # dotnet, angular
        applicationGuid:  $(productGuid)
        portfolioName:    $(portfolioName)
        productName:      $(productName)
        sourcesDirectory: $(Build.SourcesDirectory)
# ==========================================================================

# ----------------------------------------------------------------------------------------------------
# CONTINUOUS INTEGRATION BUILD
# - Run jobs/steps/tasks to build your solution here.
# - Move initialisations (NPM, NuGet,...) to PREREQUISITES section
# ----------------------------------------------------------------------------------------------------
#   TODO Insert your scripts, steps, and tasks here and remove these comments
#    - task: MSBuild@1
#      inputs:
#        solution: '**/*.sln'

# ----------------------------------------------------------------------------------------------------
# CONTINUOUS INTEGRATION TEST
# - Run jobs/steps/tasks to test your solution here
# ----------------------------------------------------------------------------------------------------
#   TODO Insert your scripts, steps, and tasks here and remove this comment

# ----------------------------------------------------------------------------------------------------
# PUBLISH
# - Publish the build and test artifacts
# ----------------------------------------------------------------------------------------------------
#   TODO Insert build and test artifact publication tasks

# ==========================================================================
# BOOTSTRAP VALIDATION, STAGE: CI_BOOTSTRAP
# SKULL & CROSS-BONES - DO NOT COMMENT OUT, OR REMOVE THIS SECTION
# ==========================================================================
    - template: Templates/Bootstrap.yml@CDTemplates
      parameters:
        bootstrapMode:    'run'
        applicationType:  'TODO REPLACE WITH APP TYPE' # dotnet, angular
        applicationGuid:  $(productGuid)
        portfolioName:    $(portfolioName)
        productName:      $(productName)
        sourcesDirectory: $(Build.SourcesDirectory)
# ==========================================================================
```

Inject at **queue** time

Inject at **queue** time



Bootstrap.xml Extract

```yaml
# ----------------------------------------------------------------------------------------------------
# WorkSafeBC Multi-Stage YAML-Based CI Pipeline Design Practice
# ----------------------------------------------------------------------------------------------------
# MODES
#   init                Runs initialisation steps
#   run                 Runs validation steps
#   runDevSecOpsOnly    Runs only DevSecOps validation steps
#   runBuildingCodeOnly Runs only Building Code validation steps
# ----------------------------------------------------------------------------------------------------
parameters:
  bootstrapMode:    'invalid'
  applicationType:  'unknown'
  applicationGuid:  'unknown'
  portfolioName:    'unknown'
  productName:      'unknown'
  sourcesDirectory: '$(Build.SourcesDirectory)'
  verbose:          'false'
  forceCheck:       'false'
  forceToolbox:     'false'
  previews:         'false'

steps:

# ==========================================================================
# BOOTSTRAP TOOLBOX
# ==========================================================================
- ${{ if or(eq( parameters.bootstrapMode, 'init' ), eq( parameters.forceToolbox, 'true' )) }}:
  - task: UniversalPackages@0
    displayName: Bootstrap download toolbox
    inputs:
      command: 'download'
      downloadDirectory: '$(System.DefaultWorkingDirectory)/Toolbox'
      feedsToUse: 'internal'
      vstsFeed: '9e070486-593d-44c7-8840-2e82011555b6/60506877-90cb-47b3-aa4c-cdd1b05db462'
      vstsFeedPackage: '84827d02-43dd-4518-8292-b17e681cb764'
      vstsPackageVersion: '*'

# ==========================================================================
# Alert if we supress
# ==========================================================================
- ${{ if and(ne(variables['Build.SourceBranchName'], 'merge'), eq( parameters.forceCheck, 'false')) }}:
  - script: echo Bootstrap Validations are supressed!
    displayName: Bootstrap Validation Supression Alert

# ==========================================================================
# BOOTSTRAP INITIALISATION STAGE
# ==========================================================================
- ${{ if and(eq( parameters.bootstrapMode, 'init' ),
             or(eq(variables['Build.SourceBranchName'],'merge'), eq( parameters.forceCheck, 'true'))) }}:
  - template: DevSecOps/SecurityScansInit.yml
    parameters:
      applicationType:        ${{parameters.applicationType}}
      portfolioName:          ${{parameters.portfolioName}}
      productName:            ${{parameters.productName}}
      sonarQubeCliSources:    ${{parameters.sourcesDirectory}}
      sonarQubeService:       'SonarQube Production'
      sonarQubeProjectsKey:   SQr2.${{parameters.portfolioName}}.${{parameters.productName}}
      sonarQubeProjectName:   ${{parameters.portfolioName}}.${{parameters.productName}}
      verbose:                ${{parameters.verbose}}
      previews:               ${{parameters.previews}}

...
```

Conditional template injection
Only run DevSecOps and Building Code validations for pull request validation builds, or if forceCheck is 'true'

Expressions
Expressions can be evaluated at compile time or at run time.
- Functions
  - and, eq, ge, gt, in, le, lt, ne, not, or, xor
  - coalesce, contains,, containsValue, counter, endsWith, format, join, length, lower, notIn, replace, startsWith, upper
- Job Status
  - Always, canceled, failed, succeeded, succeededOrFailed

```yaml
variables:
  staticVar: 'my value' # static variable
  compileVar: ${{ variables.staticVar }} # compile time expression
  isMain: $[eq(variables['Build.SourceBranch'], 'refs/heads/master')] # runtime expression

steps:
  - script: |
      echo ${{variables.staticVar}} # outputs my value
      echo $(compileVar) # outputs my value
      echo $(isMain) # outputs True
```

START (*-start.yml) Template ← extends — CI (*-ci.yml) Template ← includes — CD (*-cd.yml) Template

Validation Gate
- Environment
- Service Connection

owned by development engineering team

```yaml
trigger:
  batch: true # batch changes if true
  branches:
    include:
    - '*'    # Trigger on all branches

resources:
  repositories:
  - repository: CeSAppTemplates
    type:       git
    name:       'CeS/AzDO.Automation.Pipeline.Templates'

# Semantic version as per Naming Conventions.
name:
  $(portfolioName)_$(productName)_$(GITVERSION_MAJORMINORPATCH)_
  $(date:yyyyMMdd).$(date:HHmmss).$(Build.SourceBranchName)

# VARIABLES
variables:
- name:  portfolioName
  value: 'TODO REPLACE WITH PORTFOLIO NAME'
- name:  productName
  value: 'TODO REPLACE WITH PRODUCT NAME'
- template: /Operations/Config/${{variables.portfolioName}}
            -${{variables.productName}}-config.yml@CeSTemplates
extends:
  template: /Blueprints/AzureFunction/
            azure-pipeline-azure-function-ci.yml@CeSAppTemplates
  parameters:
    portfolioName:  $(portfolioName)
    productName:    $(productName)
    # suppressCD:     true
```

owned by cloud engineering and pipeline working group

```yaml
parameters:
- name:     portfolioName
  type:     string
- name:     productName
  type:     string
# Bootstrap optionals
- name:     verbose
  type:     boolean
  default:  false
- name:     forceToolbox
  type:     boolean
  default:  false
- name:     previews
  type:     boolean
  default:  false
# CI Blueprint optionals
- name:     customCDTemplate
  type:     string
  default:  'blueprint'
- name:     suppressCD
  type:     boolean
  default:  false
- name:     customQATemplate
  type:     string
  default:  'blueprint'
- name:     suppressQA
  type:     boolean
  default:  false
- name:     loadDVTBox
  type:     string
  default:  'no'
```

```yaml
stages:
- stage: CI
  displayName: Continuous Integration

  # Configure the default agent pool and image to use for the pipeline
  pool:
    name:     'Azure Pipelines'
    vmImage:  'Windows-latest'

  jobs:
  - job: ContinuCIousIntegration
    steps:
    - template: /Templates/Utility/git-tools-git-version.yml
      parameters:
        versionSpec: '5.x'

    - task: PowerShell@2
      name: setVersion
      displayName: "Set Version"
      inputs:
        targetType: 'inline'
        script: Write-Host "##vso[task.setvariable
                            variable=semVersion;isOutput=true]
                            $(GitVersion.MajorMinorPatch)"
        failOnStderr: true
        pwsh: true
      continueOnError: false

    - template: /Templates/boot-strap.yml
        parameters:
          bootstrapMode:        'init'
          applicationType:      'universal artifacts'
          applicationBlueprint: 'universal-artifact'
          portfolioName:        ${{parameters.portfolioName}}
          productName:          ${{parameters.productName}}
          sourcesDirectory:     $(Build.SourcesDirectory)
          verbose:              ${{parameters.verbose}}
          forceToolbox:         ${{parameters.forceToolbox}}
          previews:             ${{parameters.previews}}
          loadDVTBox:           ${{parameters.loadDVTBox}}

    - task: PublishBuildArtifacts@1
        displayName: 'Publish Artifact'
        inputs:
          PathtoPublish:  '$(Build.SourcesDirectory)/$(publishFolder)'
          ArtifactName:   'drop'

    - template: /Templates/boot-strap.yml
        parameters:
          bootstrapMode:        'run'
          applicationType:      'dotnet'
          applicationBlueprint: 'universal-artifact'
          portfolioName:        ${{parameters.portfolioName}}
          productName:          ${{parameters.productName}}
          sourcesDirectory:     $(Build.SourcesDirectory)
          verbose:              ${{parameters.verbose}}
          forceToolbox:         ${{parameters.forceToolbox}}
          previews:             ${{parameters.previews}}
          loadDVTBox:           ${{parameters.loadDVTBox}}

    - ${{ if and( ne( parameters.suppressCD, true),
                  ne( lower(parameters.customCDTemplate), 'blueprint' )) }}:
      - template: /Operations/Custom/${{parameters.customCDTemplate}}.yml
    - ${{ if and( ne( parameters.suppressCD, true),
                  eq( lower(parameters.customCDTemplate), 'blueprint' )) }}:
      - template: /Blueprints/ArtifactsUniversal/
                   azure-pipeline-universal-artifact-cd.yml
```

flow variable

```yaml
stages:

- stage:          'Development'
  dependsOn:
  - CI
                                …<snip>…

- stage:          'SecurityScans'
  displayName:    'Security Scans'
  dependsOn:
    - CI
    - Development
  pool:
    vmImage:      $(securityScansStageVmImage)
  jobs:
  - job:          'SecurityScans'
    steps:
    - template:   /Templates/DevSecOps/security-scans-cd.yml

- ${{ if or( eq(lower(variables['Build.SourceBranch']),
               'refs/heads/release'),
               startsWith(lower(variables['Build.SourceBranch']),
               'refs/heads/release/')) }}:
  - stage:          'Production'
  - stage:          'SecurityReview'
    displayName:    'Security Review'
    dependsOn:
    - Development
                                …<snip>…

- ${{ if or( eq(lower(variables['Build.SourceBranch']),
               'refs/heads/release'),
               startsWith(lower(variables['Build.SourceBranch']),
               'refs/heads/release/')) }}:
  - stage:          'Production'
    dependsOn:
    - CI
    - SecurityScans
    - SecurityReview
    variables:
      currentVersion: $[ stageDependencies.CI.CI
                         .outputs['setVersion.semVersion'] ]
    pool:
      vmImage:      $(productionStageVmImage)
    jobs:
    - deployment:   'Production'
      environment:  $(productionStageEnvName)
      strategy:
        runOnce:
          deploy:
            steps:
            - script: echo Toolkit Version = $(currentVersion)
            - task: UniversalPackages@0
              name: Create_PR_Universal_Package
              inputs:
                command: 'publish'
                publishDirectory:       $(Agent.BuildDirectory)/drop
                feedsToUsePublish:      'internal'
                vstsFeedPublish:        $(productionStageFeedPublish)
                vstsFeedPackagePublish: $(productionStagePackagePublish)
                versionOption:          'custom'
                versionPublish:         $(semVersion)
```

Optional parameters

PARAMETER	VALUES	DESCRIPTION
forceCheck	'false', 'true'	Enforce validations, irrespective of source branch.
previews	'false', 'true'	Evaluate new preview features and provide feedback`.
verbose	'false', 'true'	Verbose logging.
forceToolbox	'false', 'true'	Force a reload of the toolbox config and scripts.
loadDVTBox	'no', 'yes'	If 'yes' load the DV toolbox, else load PROD toolbox.
customCDTemplate	'blueprint'	Default to app-type CD blueprint if 'blueprint', else override.
customQATemplate	'blueprint'	Default to app-type CD blueprint if 'blueprint', else override.
suppressCD	true, false	Suppress CD
suppressQA	true, false	Suppress QA Scans

Your Application Repo
All changes should be made through and protected by the pull request workflow

CeS.Automation.Pipelines.Templates Repo
All changes made through and protected by the pull request wortkflow

includes

Variable Template (config)

```yaml
- name:  productionStageEnvName
  value: 'SHARED_PR_Universal_Artifacts'
- name:  productionStageVmImage
  value: 'windows-latest'
- name:  productionStageFeedPublish
  value: '9e070486-893d-44c7-8840-2e82011555b6/60506877-90cb-47b3-aa4c-cdd1b05db462'
- name:  productionStagePackagePublish
  value: 'toolbox'
```

WORK SAFE BC
WORKING TO MAKE A DIFFERENCE

App-type CI/CD Blueprint-based Pipeline Quick Reference

2021.04.30 v1.2 WS - See https://aka.ms/yaml for Azure DevOps Pipeline documentation

Quick Context Switch

HOW did we do it?

Out of scope

Story for another day

However, …

DEMO TIME

Q&A

You have

Questions

we (hopefully) have

Answers

REFERENCES

- WorkSafeBC Technical Blog
- Pipelines - Why bother and what are our nightmares and options?
- Pipeline-as-code wrapped with Pull Requests
- Quick Reference Sheet for YAML and Generic Blueprint-based Pipelines
- Quick Reference Sheet for Application-type Blueprint-based Pipelines
- Magic video