

**NAV
TECH
DAYS**
2018

mibuso.com

CI/CD for Business Central

GUNNAR GESTSSON, KAMIL SÁČEK, AREND-JAN KAUFFMANN

Introduction

- Gunnar Gestsson
 - Advania, Iceland
 - MVP since 2013
- Kamil Sáček
 - Navertica, Czech Republic
 - MVP since 2005
- Arend-Jan Kauffmann
 - Cloud Ready Software, Netherlands
 - MVP since 2013



Agenda

- What is CI/CD
- Create repository
- Create build pipeline
- Branch policy
- Delivery pipeline
- Deployment pipeline



**Continuous
Irritation
&
Continuous
Distraction**



What is CI / CD?

Continuous Integration

- Merging all developer work to a shared main branch
- Developer's changes are validated by creating a build and running automated tests against the build

Continuous Delivery

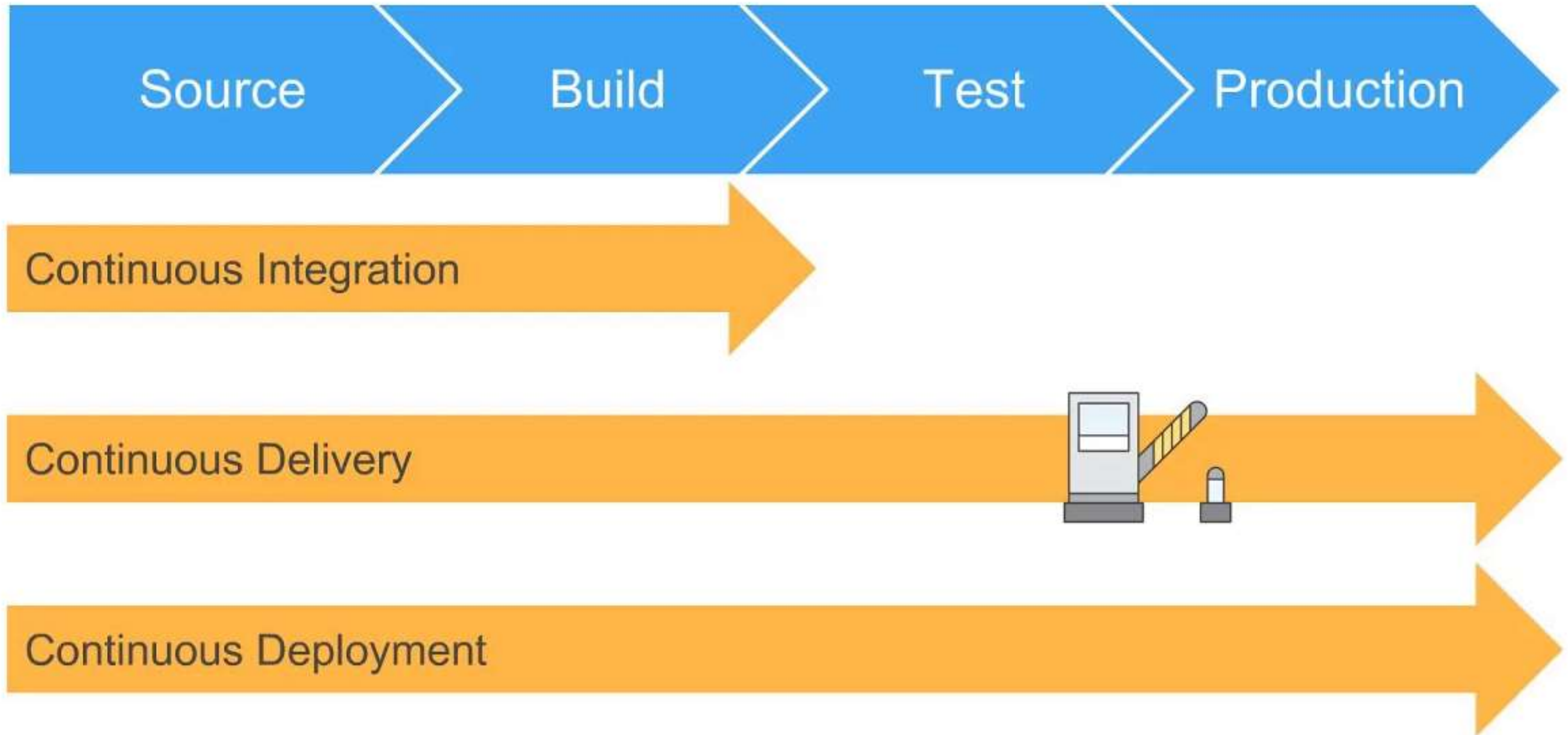
- Produce software in short cycles, ensuring it can be reliably released at any time
- Release process on top of continuous integration
- Possible to deploy at any point of time

Continuous Deployment

- It is one step further than continuous delivery
- Automatic deployment to customer environment



CI / CD in a picture



Continuous Integration

What you need

- Automated tests
- Process to monitor main repository and run tests automatically
- Developers need to merge their changes as often as possible

What you get

- Les bugs
- Building the release is easy
- Developers are alerted as soon as they break the build
- Testing costs are reduced

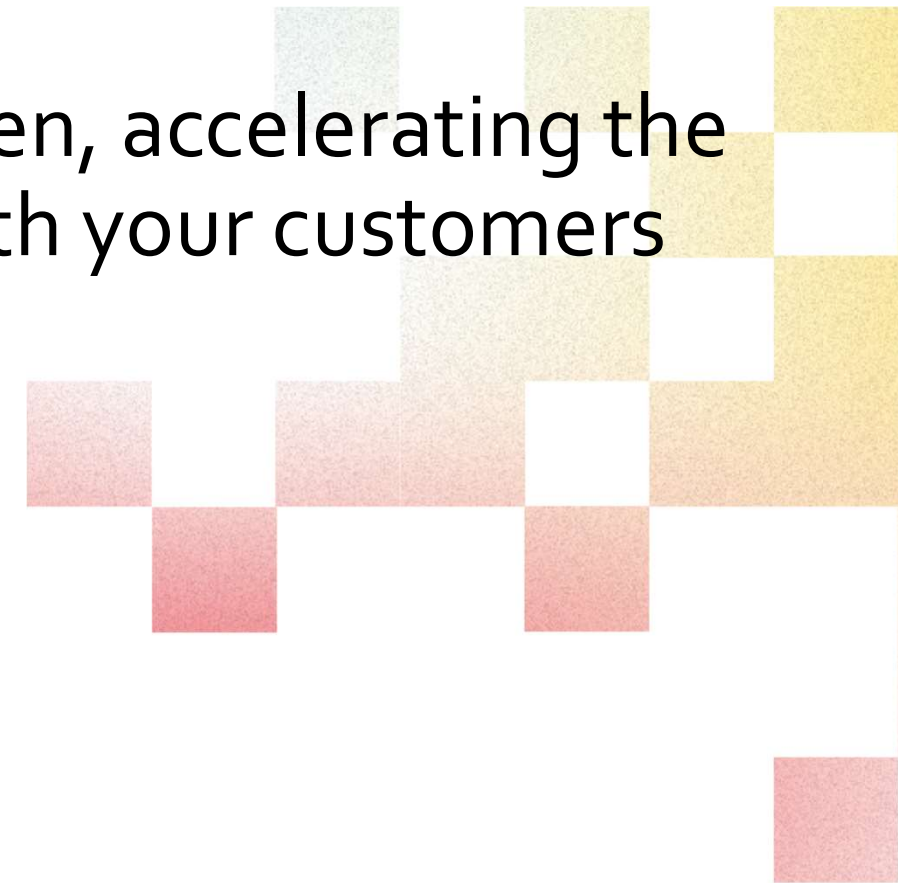
Continuous Delivery

What you need

- Continuous integration process and a test suite that covers enough of your codebase
- Deployment is manually triggered, but the process itself can be automated

What you get

- Complexity of deploying software goes away, preparing a release is easier
- Release more often, accelerating the feedback loop with your customers



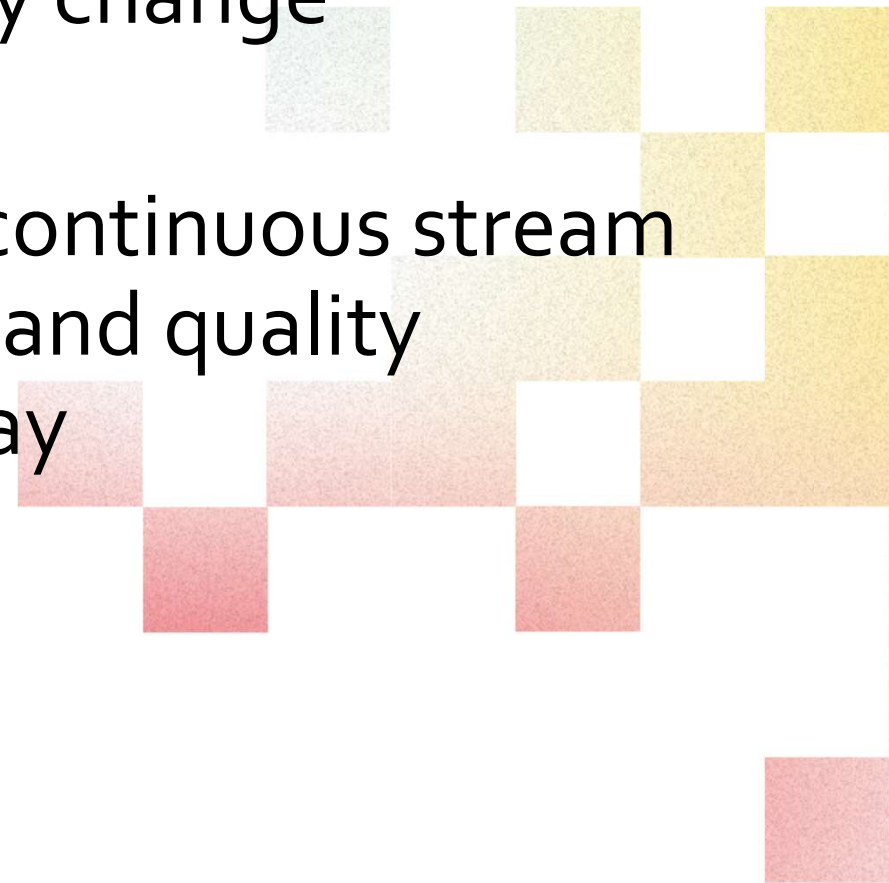
Continuous Deployment

What you need

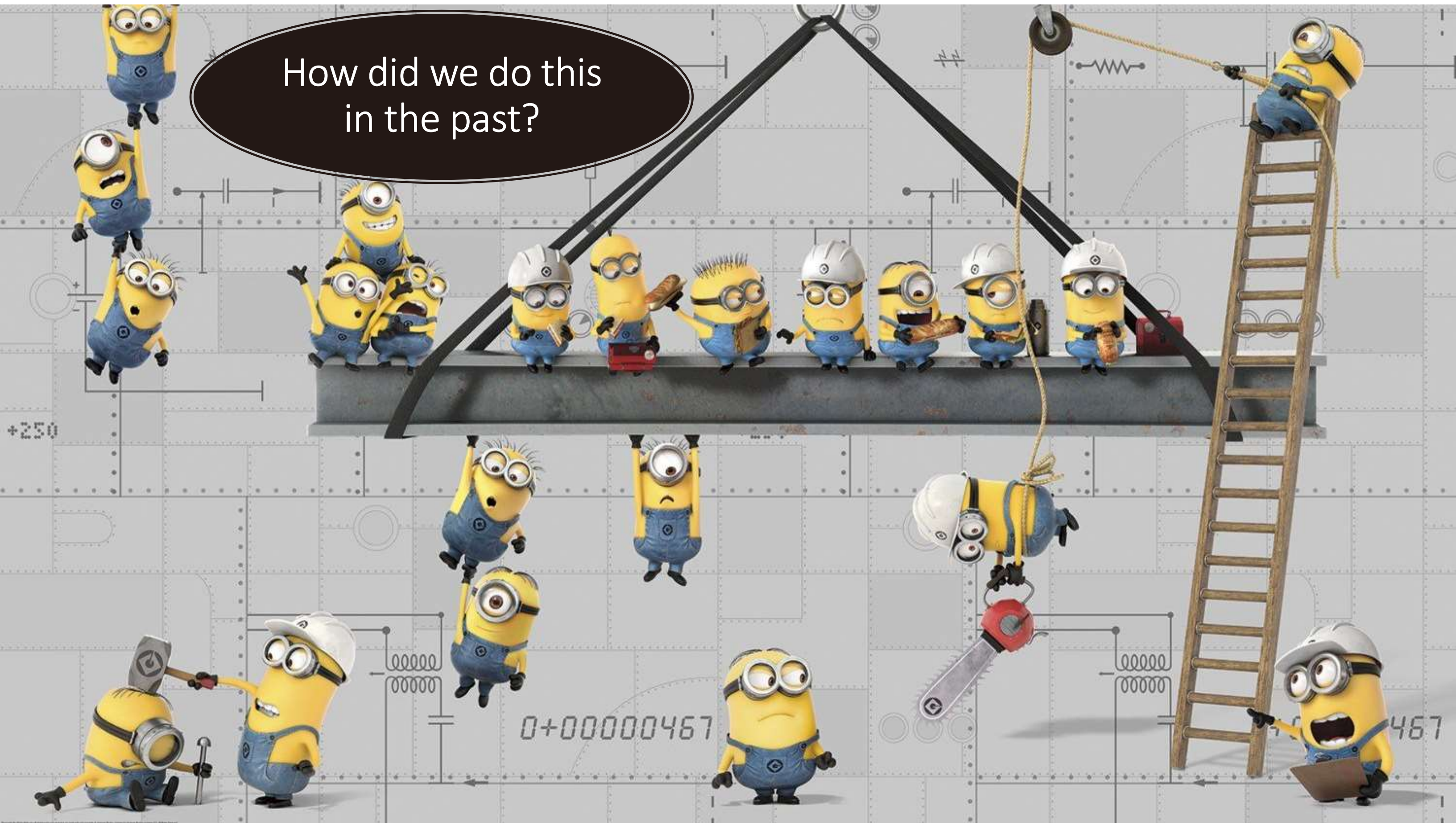
- Testing culture needs to be at its best. The quality of the test suite determines the quality of the release
- Documentation process needs to keep up with the pace of deployments

What you get

- Develop faster, no need to pause development for releases. Deployment pipelines are triggered automatically very change
- Customers see a continuous stream of improvements and quality increases every day



How did we do this
in the past?



Importing fob files

DynamicsNAV80 - Microsoft Dynamics NAV Development Environment - [Import Worksheet]

Archivo Edición Ver Herramientas Ventana ?

✂ 📄 📁 🔍 ✖ 🏠 📊 🔄 🔄 🔄 🔄

Type	No.	Name	New Object	New Obje...	Existing ...	W...	Action	Existing ...	Existing Object Version List	New Obje.
Table	36	Sales Header		✓	✓	!	Merge: Existing<-New	✓	NAVW 18.00.00.40459, NAVES...	
Table	38	Purchase Header		✓	✓	!	Merge: Existing<-New	✓	NAVW 18.00.00.40459, NAVES...	
Table	5994	Service Cr.Memo Header		✓			Replace		NAVW 18.00, NAVES8.00	
Table	10752	SII Doc. Upload State	✓				Create			
Codeunit	10750	SII XML Creator	✓				Create			
Codeunit	10752	SII Doc. Upload Management	✓				Create			

	Name	Modified	Size	Date	Time	Version list
Existing	Purchase Header	✓	320820	24/08/17	10:32:24	NAVW 18.00.00.40459, NAVES8.00.00.40459, INIK, SINASE
New	Purchase Header		310372	02/08/17	12:00:00	NAVW 18.00.00.49000, NAVES8.00.00.49000

Aceptar Cancelar Replace All Save As... Ayuda

Action: Merge: Existing<-New SINASE\jfigueras INS

Merging objects

2017-12-06 All Base Mods RM365 Merged5.txt <--> 2017-12-06 All Base Mods RM365 Compare.txt* - Text Compare - Beyond Compare

Session File Edit Search View Tools Help [New version available...](#)

Sessions

D:\ODTRM\RM365 V2.0\2017-12-06 Compare\2017-12-06 All Base Mods RM365 Merged5.txt 12/6/2017 2:05:45 PM 1,003,935 bytes <default> ANSI PC

D:\...\2017-12-06 Compare\2017-12-06 All Base Mods RM365 Compare.txt 12/6/2017 12:02:39 PM 1,241,408 bytes <default> ANSI PC

```
{ 10017; ;Provincial Tax Area Code;Code20
CaptionML=ENU=Do Not Use For Tax Calculation }
;TableRelation="Tax Area" WHERE (Country/Region=CO
OnValidate=BEGIN
IF "Do Not Use For Tax Calculation"
"Provincial Tax Area Code" := '';
END;

CaptionML=ENU=Provincial Tax Area Code }
{ 70014200;;Allow Rentals ;Boolean ;CaptionML=ENU=Allow Rentals;
Description=RM36510.00 }
{ 70014201;;Temporary Location ;Boolean ;CaptionML=ENU=Temporary Location;
Description=RM36510.00 }
}
KEYS
{
{ ;Code ;Clustered=Yes }
{ ;Name }
{ ;Use As In-Transit,Bin Mandatory }
```

```
{ 10017; ;Provincial Tax Area Code;Code20
CaptionML=ENU=D
;TableRelation="
OnValidate=BEGI
IF
END;

CaptionML=ENU=P
}
KEYS
{
{ ;Code ;Clustered=Yes }
{ ;Name }
{ ;Use As In-Transit,Bin Mandatory }
```

464: 1 Default text Modified

.....{.70014200;;Allow.Rentals.....;Boolean.....;CaptionML=ENU=Allow.Rentals;}

18 difference section(s) Important Left Orphan Insert 0.14 seconds

Merging objects next level

The image shows a screenshot of a development environment with Visual Studio Code and Windows PowerShell ISE. In Visual Studio Code, the 'COD80.CONFLICT' file is open, displaying a JSON-like structure for 'Sales-Post' (Codeunit 80). The structure includes 'OBJECT-PROPERTIES', 'PROPERTIES', 'CHANGES', and 'CODE'. The 'RESULTS' pane on the left lists various text files, with 'COD80.CONFLICT' highlighted. Overlaid on top of the Visual Studio Code window is the 'Administrator: Windows PowerShell ISE' window. It shows a script titled '2 - retrieve and merge objects.ps1' being executed. The script contains a comment: '# Step 5 - merge ORIGINAL, MODIFIED and TARGET objects into the RESULTS folder'. Below the script, the output of the command is displayed in a blue console window. The output shows that 860 application objects were processed, with 744 merged, 115 in conflict, 1 inserted, 0 deleted, 0 unchanged, and 0 failed.

```
1 OBJECT Modification "Sales-Post"(Codeunit 80)
2 {
3   OBJECT-PROPERTIES
4   {
5     Date=;
6     Time=;
7     Version List=;
8   }
9   PROPERTIES
10  {
11    Target="Sales-Post"(Codeunit :
12  }
13  CHANGES
14  {
15    { CodeModification ;Target=F
16      Conflict
17    { CodeModification ;Target=C
18      Conflict
19  }
20  CODE
21  {
22    BEGIN
23    END.
24  }
25  }
26 }
```

```
74 # Step 5 - merge ORIGINAL, MODIFIED and TARGET objects into the RESULTS folder
75 $Mergeresult = Merge-NAVApplicationObject -originalPath $originalFolder
76 -ModifiedPath $modifiedFolder
77 -TargetPath $targetFolder
78 -ResultPath $resultsFolder
79 -VersionListProperty FromTarget
80 -DateTimeProperty FromTarget
```

Details:

Category	Count	Description
Processed	860	application object(s):
Merged	744	objects - with changes in MODIFIED that were successfully merged with any changes from TARGET into RES
Conflict	115	objects - with changes in both MODIFIED and TARGET that could only be merged partially. Partially merged objects and corresponding .CONFLICT files are added to RESULT. This also includes objects that are deleted in MODIFIED/TARGET and changed in TARGET/MODIFIED
Inserted	1	objects - in MODIFIED that do not exist in TARGET and are inserted into RESULT.
Deleted	0	objects - that exist in ORIGINAL, but do not exist in MODIFIED and are unchanged in TARGET.
Unchanged	0	objects - in TARGET which are not changed in MODIFIED and are copied from TARGET to RESULT. This also include objects deleted in both MODIFIED and TARGET and objects unchanged in MODIFIED
Failed	0	objects - that could not be imported, such as an object that is not valid or that contains unsupported features.

Are you too
busy to
improve?



New Solution

- Create a repository from a template
- Prepare the solution for development
- Push to GIT



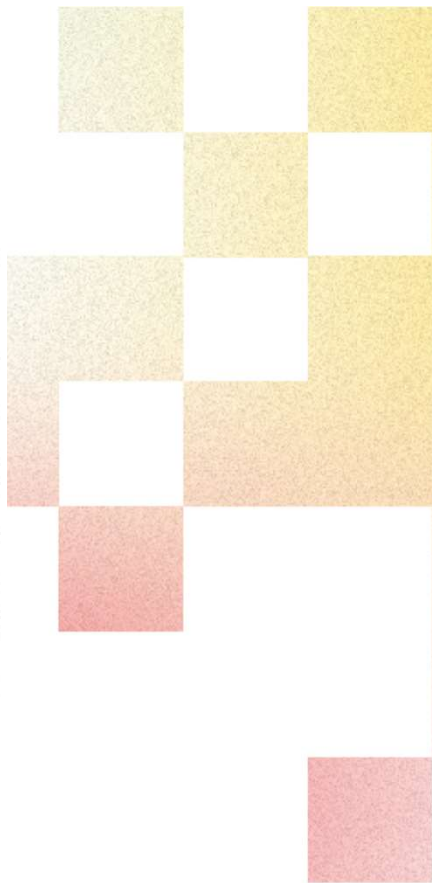
Solution Architect

DEMO



Links

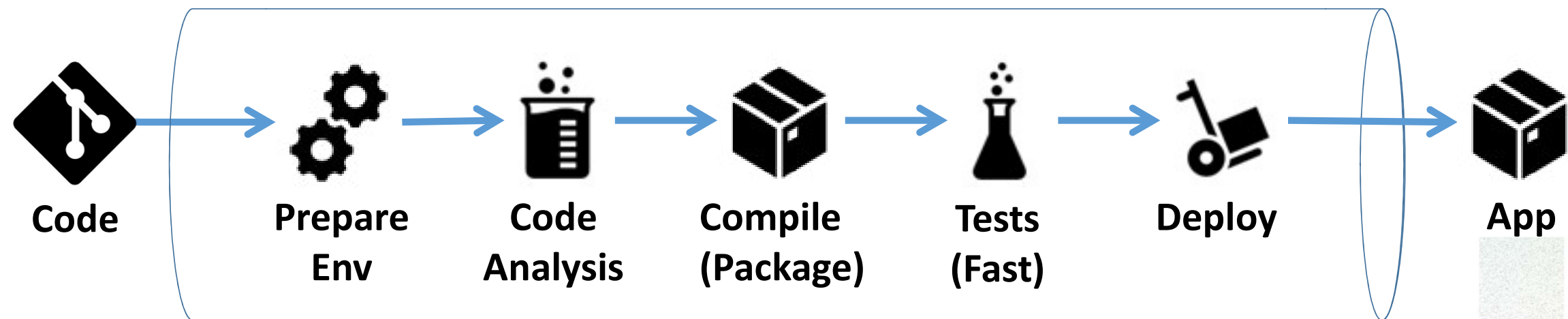
- <https://github.com/gunnargestsson/businesscentral>
 - Templates to use in your Business Central development
- <https://github.com/gunnargestsson/AdvaniaGIT>
 - VS Code Extensions and PowerShell Scripts



We have the code on server. What we will do now?



Build pipeline



Build pipeline as Code

- Imagine that you can create Build pipeline as any other code in your app

- Wishes:
 - I can use SCM for it
 - I do not need to leave VSCode to change it
 - I can use pre-prepared building blocks
 - When I fix bug in it, all Build pipelines will be updated
...and potentially screwed... 😊



Build pipeline as Code

Time to learn YAML

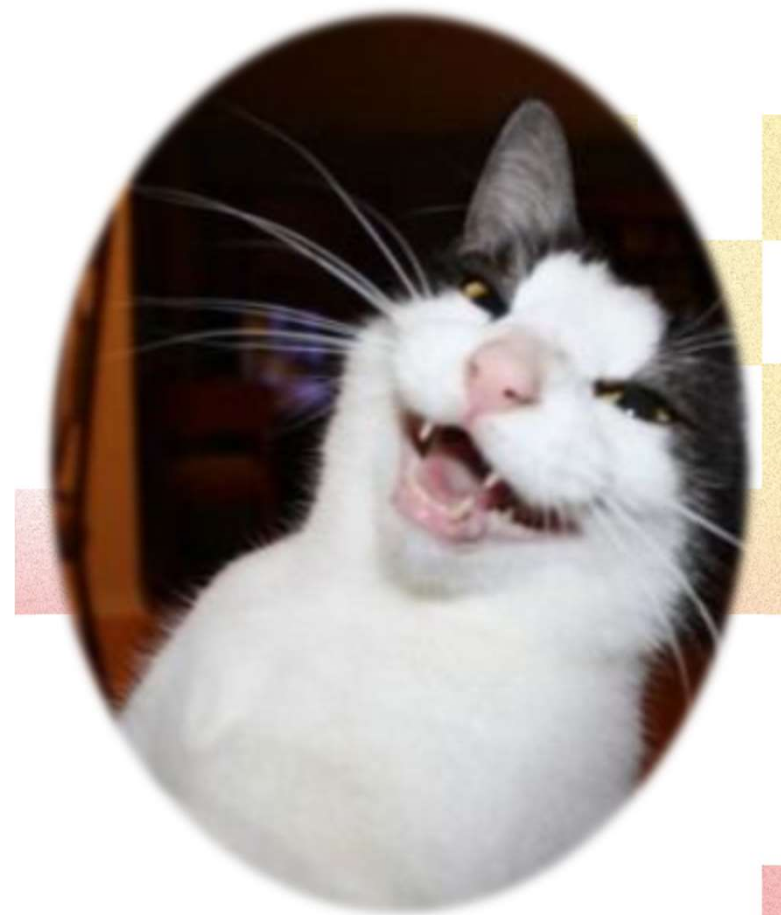


Do not be afraid, it will be quick....

YAML

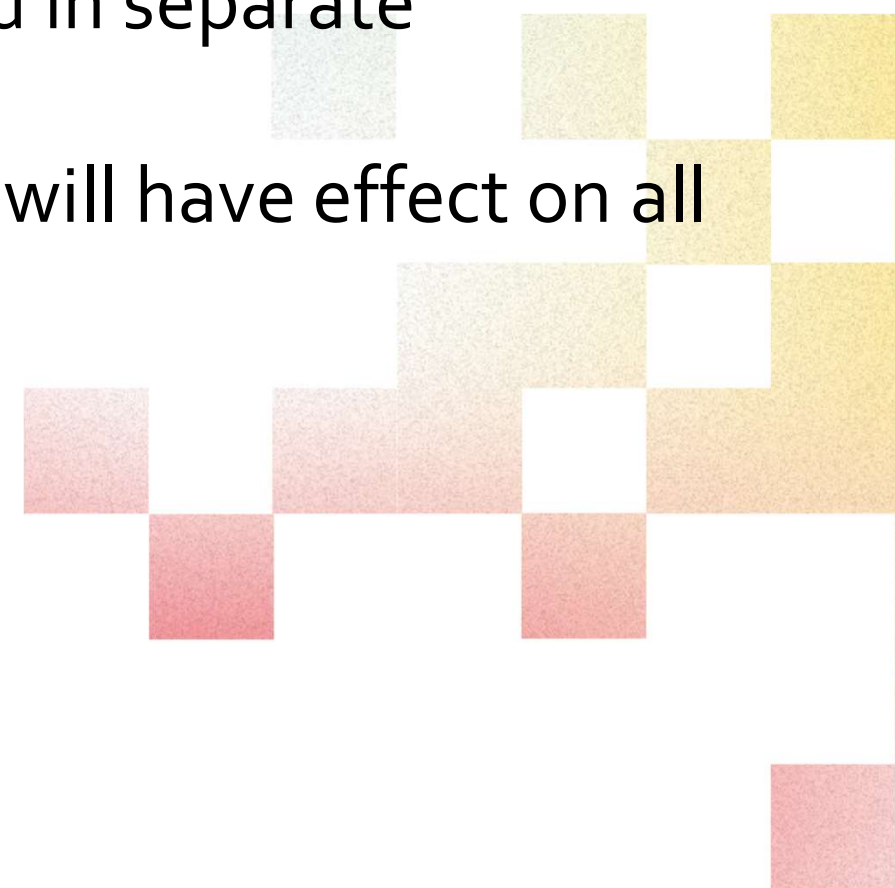
- „*YAML Ain't Markup Language*“
- „human-readable data serialization language“ (Wikipedia)
- Indentation is important

```
1 receipt:    Oz-Ware Purchase Invoice
2 date:      2012-08-06
3 customer:
4   first_name: Dorothy
5   family_name: Gale
6
7 items:
8   - part_no: A4786
9     descrip: Water Bucket (Filled)
10    price: 1.47
11    quantity: 4
12
13   - part_no: E1628
14     descrip: High Heeled "Ruby" Slippers
15     size: 8
16     price: 133.7
17     quantity: 1
18
19 specialDelivery: >
20   Follow the Yellow Brick
21   Road to the Emerald City.
22   Pay no attention to the
23   man behind the curtain.
24
```



Build pipeline as Code

- Imagine that you can create Build pipeline as any other code
- You can use SCM for it
- You do not need to leave VSCode to change it
- You can use pre-prepared building blocks
- When you fix bug in it, all Build pipelines will be updated
- Create ***azure-pipelines.yml***
- Put it into root of your repository
- VSCode have support for YAML format
- Templates are saved in separate repository
- Changing template will have effect on all pipelines using it



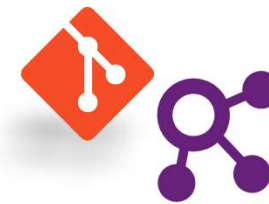
Build pipeline as Code



App repository

Azure-pipelines.yml

```
17 resources:
18   repositories:
19     - repository: MSDYN365BC_Yaml
20       type: github
21       name: kine/MSDYN365BC_Yaml
22       ref: 'refs/heads/master'
23       endpoint: GitHub
24     - repository: self
25       checkoutOptions:
26         submodules: true
27
28   You, 6 days ago | 1 author (You)
29   variables:
30     TestCodeunitId: 130402
31     UserPwd: $(USER.PASSWORD)
32     CertFile: ''
33     CertPwd: ''
34   steps:
35     - checkout: self
36       submodules: true
37     - template: Templates/SetAppBuildNo.yml@MSDYN365BC_Yaml
38       parameters:
39         UpdateDevOpsBuildNo: 1
40     - template: Templates/PrepareEnv.yml@MSDYN365BC_Yaml
41       parameters:
42         pwd: $(UserPwd)
43
```



Template repository

SetAppBuildNo.yml

```
2 parameters:
3   UpdateDevOpsBuildNo: 1
4
5 steps:
6   - powershell: |
7     Import-Module NVRAppDevOps -DisableNameChecking
8     $Config = Read-ALConfiguration -Path .\
9     $Config | Set-ALAppBuildNo -UpdateDevOpsBuildNo:([bool]$env:UpdateDevOpsBuildNo)
10
11     I
12   displayName: 'Set App Build and Revision No.'
13   env:
14     UpdateDevOpsBuildNo: ${ parameters.UpdateDevOpsBuildNo }
15
```

PrepareEnv.yml

```
3 parameters:
4   pwd: 'pass@word1'
5
6 steps:
7   - powershell: |
8     Import-Module NVRAppDevOps -DisableNameChecking
9
10     $Config = Read-ALConfiguration -Path .\ -Password $env:USER_PWD
11     $Config | Init-ALEnvironment -Build $True -Password $env:USER_PWD
12
13   failOnStderr: true
14   displayName: 'Prepare environment'
15   env:
16     USER_PWD: ${ parameters.pwd }
```

Build pipeline as Code

DEMO



Links

- https://github.com/kine/MSDyn365BC_Yaml
 - Templates to use in your yaml
- https://github.com/kine/MSDyn365BC_AppTemplate
 - Template of App project including yaml pipeline definition
- <https://marketplace.visualstudio.com/items?itemName=Kine.navertical>
- NaverticAL – VSCode extension



Pull Request

Request to merge changes from a branch to the master branch

Linked to a work item

Changes can be reviewed

Comments can be made

Reviewers can vote on the changes

- Approve
- Approve with suggestions
- Wait for author
- Reject





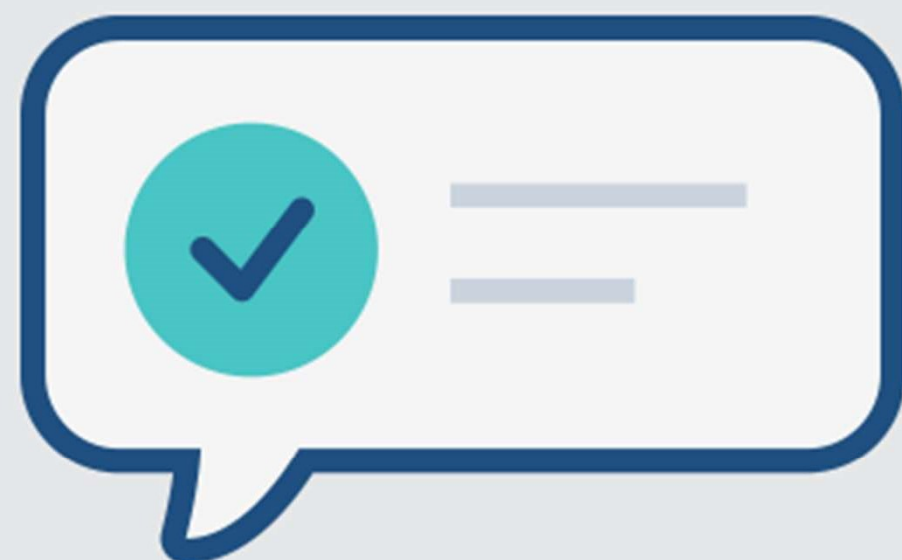
Create Branch



Create Pull Request



Merge Branch



Branch Policy

Require a pull request to merge changes into master branch

Enforces to

- Use a branch to develop code in
- Use a pull request to merge code

Branches can be linked to a work item



Require minimum number of reviewers

Require certain number of reviewers

They all need to approve the code change

☒ **Require a minimum number of reviewers**
Require approval from a specified number of reviewers on pull requests.

Minimum number of reviewers

☐ Allow users to approve their own changes.

☐ Allow completion even if some reviewers vote "Waiting" or "Reject".

☐ Reset code reviewer votes when there are new changes.



Check for linked work items

Require association between pull request and a work item

☒ **Check for linked work items**
Encourage traceability by checking for linked work items on pull requests.

Policy requirement

☒ **Required**
Block pull requests from being completed unless they have at least one linked work item.

☐ **Optional**
Warn if there are no linked work items, but allow pull requests to be completed.



Comment resolution

Require that all comments on a pull request have been resolved



Check for comment resolution

Check to see that all comments have been resolved on pull requests.

Policy requirement



Required

Block pull requests from being completed while any comments are active.



Optional

Warn if any comments are active, but allow pull requests to be completed.



Merge strategy

Maintain a consistent branch history

- **No-fast-forward** merge will merge the commit history of the source branch
- **Squash** merge creates a single commit in the target branch



Enforce a merge strategy

Require a specific type of merge when pull requests are completed.



No-fast-forward merge

The source branch and all its commits will appear as a second parent to the merge commit.



Squash merge

Condense all the changes in a pull request into one commit with one parent. [Learn more](#)



Add Build validation

A successful build with the merged code is required before the pull request can be completed

Build validation
Validate code by pre-merging and building pull request changes

+ Add build policy

Add build policy

Build pipeline *

Path filter (optional) ⓘ

No filter set

Trigger

☒ Automatic (whenever the source branch is updated)


☐ Manual


Policy requirement

☒ Required
Build must succeed in order to complete pull requests.

☐ Optional
Build failure will not block completion of pull requests.

Build expiration

☐ Immediately when  master is updated

☒ After hours if  master has been updated

☐ Never

Display name



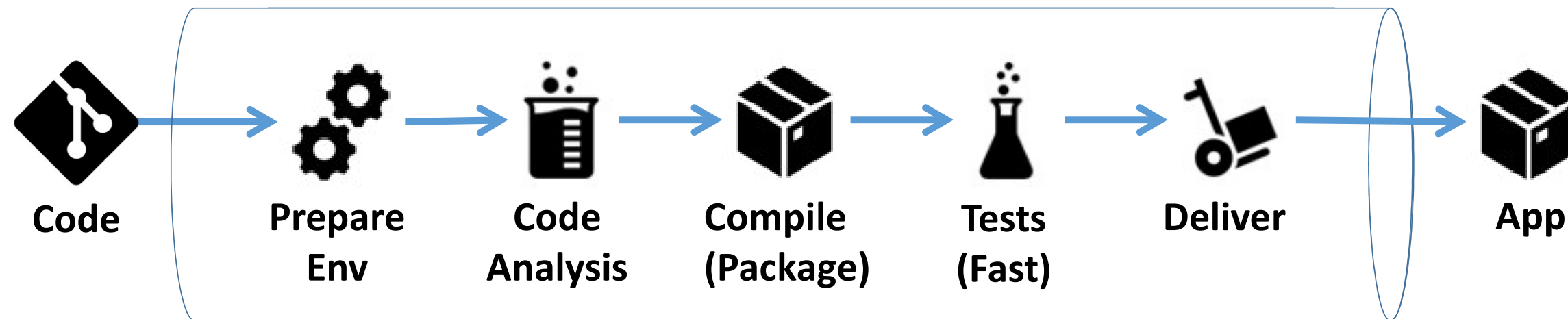
Demo branch policy

DEMO



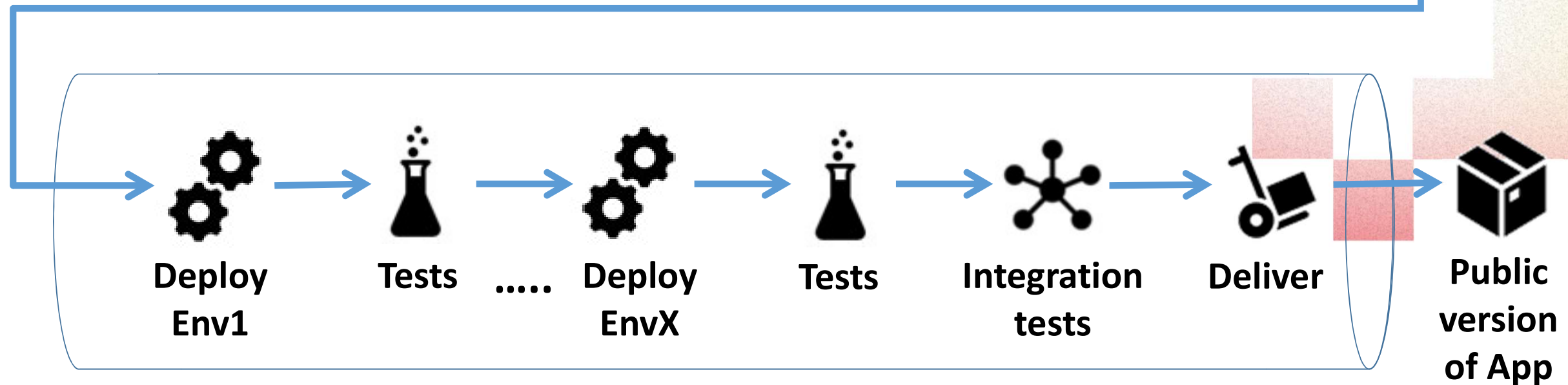
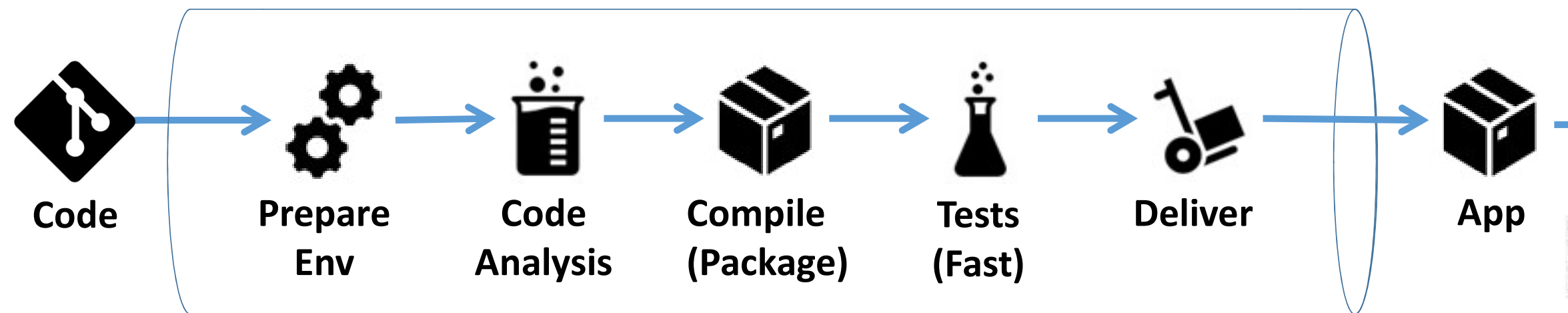
Build pipeline

Build Pipeline



Build and Release pipeline

Build Pipeline



Release Pipeline

Continuous Delivery

- You can deliver to:
 - File system
 - Azure DevOps Artifacts
 - Another package server
- Release pipeline by code (YAML)
 - Planned for Q4 2018
 - <https://docs.microsoft.com/en-us/azure/devops/release-notes/>



Delivery pipeline

DEMO



Hybrid Environment

- C/AL Core
- C/AL ISV
- C/AL Solutions
- AL ISV
- AL Solutions
- AL Customizations

Development

Allowed Extension Target Level:

Internal

HttpClient AL Function Maximum Tim...

00:05:00

Enable Developer Service Endpoint:

☐

HttpClient AL Function Response Size:

15

Enable loading application symbol references at serv...

☒

Port:

7149

Enable SSL:

☐

Enable Developer Service Endpoint

Specifies whether the Developer service endpoint will be enabled.

Hybrid Development

- Use Build Server to build AL symbol packages
 - Include all C/AL used in Hybrid Environment
 - Include full test toolkit
- mcr.microsoft.com/businesscentral/onprem for OnPrem Docker Container

▼ AL Packages

IS365 2018 AL Packages

✓ #76

4 hours ago

IS365 BC AL Packages

✓ #28

4 hours ago

LS PUB 2018 AL Packages

✓ #70

5 hours ago

PUB 2018 AL Packages

✓ #74

5 hours ago

Hybrid Continuous Deployment

- Import new FOB
- Sync all tenants
- Execute Data Upgrade
- Create symbol references
- Try not to remove table fields, use field Obsolete settings.
- Uninstall and unpublish old APP
- Publish new APP
- Sync all tenants where the old APP was installed
- Execute Data Upgrade for new App



C/AL to AL in Hybrid Environment

- Remove C/AL objects, data moved to temporary UPG tables
- Read from UPG tables in AL app install Codeunit
- Remove UPG tables

```
codeunit 10035990 "04N Konto Installation"
{
    Subtype = Install;
    trigger OnInstallAppPerCompany()
    begin
        GetUPGDataFromCALTable(10035985, Database::"Konto Service Setup");
        GetUPGDataFromCALTable(10035986, Database::"Konto Integration Record");
    end;

    2 references
    local procedure GetUPGDataFromCALTable(SourceTableId: Integer; DestinationTableId: Integer)
    var
        Source: RecordRef;
        Destination: RecordRef;
    begin
        Destination.Open(DestinationTableId);
        if TableExists(SourceTableId) then begin
            Source.Open(SourceTableId);
            if Source.FindSet() then begin
                repeat
                    Destination.Init();
                    CopyRecRef2RecRef(Source, Destination);
                    Destination.Insert();
                until Source.Next() = 0;
                Source.DeleteAll();
            end;
        end;
    end;

    1 reference
    local procedure CopyRecRef2RecRef(Source: RecordRef; var Destination: RecordRef)
    var
        DataTypeMgt: Codeunit "Data Type Management";
        SourceField: FieldRef;
        DestinationField: FieldRef;
        FieldIndex: Integer;
    begin
```

Any Questions?

Thank
THANK YOU
you