Technical Articles

Lars Hvam
February 20, 2020 5 minute read

Fully supporting CI in ABAP AS

Introduction

At the recent #DSAGTT20, SAP showed a few slides regarding in the area of continuous integration and ABAP:

https://twitter.com/PanzerDominik/status/1227274686750478338/photo/1

https://twitter.com/WachterSascha/status/1227254813542559745/photo/1

I did not attend the event, so the below might be out of context.

Continuous Integration

Continuous Integration(CI) is a word that can mean a lot of different things to different people, to me its something about providing feedback to the developer for each change, the feedback must be reliable and provided within a timely manner.

A change could be whenever the development is saved, or a commit, or something else, its really up to the development environment, which is a different topic. The developer needs the information to determine if its okay to push the change to a wider audience.

Also exactly how the development is deployed to the CI system is a deployment topic. The scenarios below can be set up using the classic transport system.

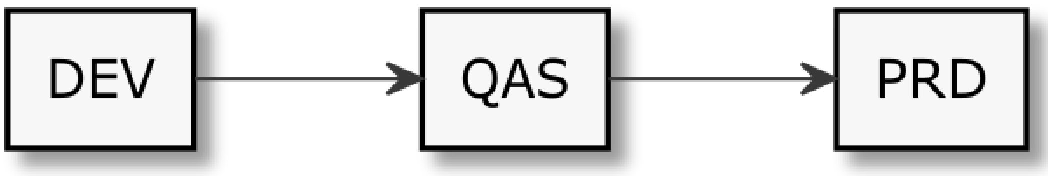
Elements of a CI pipeline might include:

- Code Inspector / ATC
- Coverage results
- Unit tests
- Integration tests
- eCATT
- UI tests
- Performance tests
- Metrics

Scenario 1 – Baseline

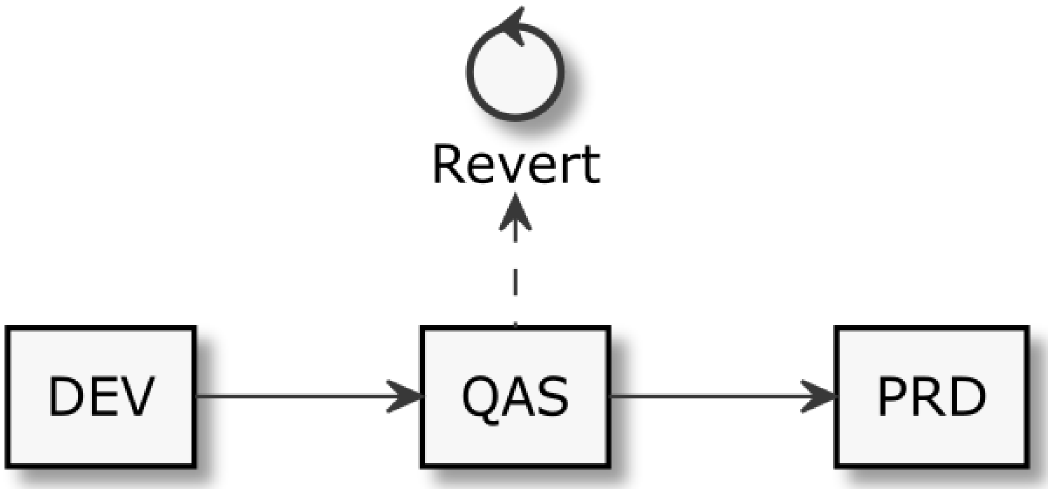
Let’s take a common setup with development, test and production. The developers do development in the development system, the test system is used for manual testing.

Assumption: All organizations have a requirement for doing manual exploratory testing, even if all testing is automated, the QAS system cannot be scrapped.



Scenario 2 – Use QAS for CI

For each change done by the developers, the change is deployed to QAS and CI is run. When the results have been reported the change is reverted.



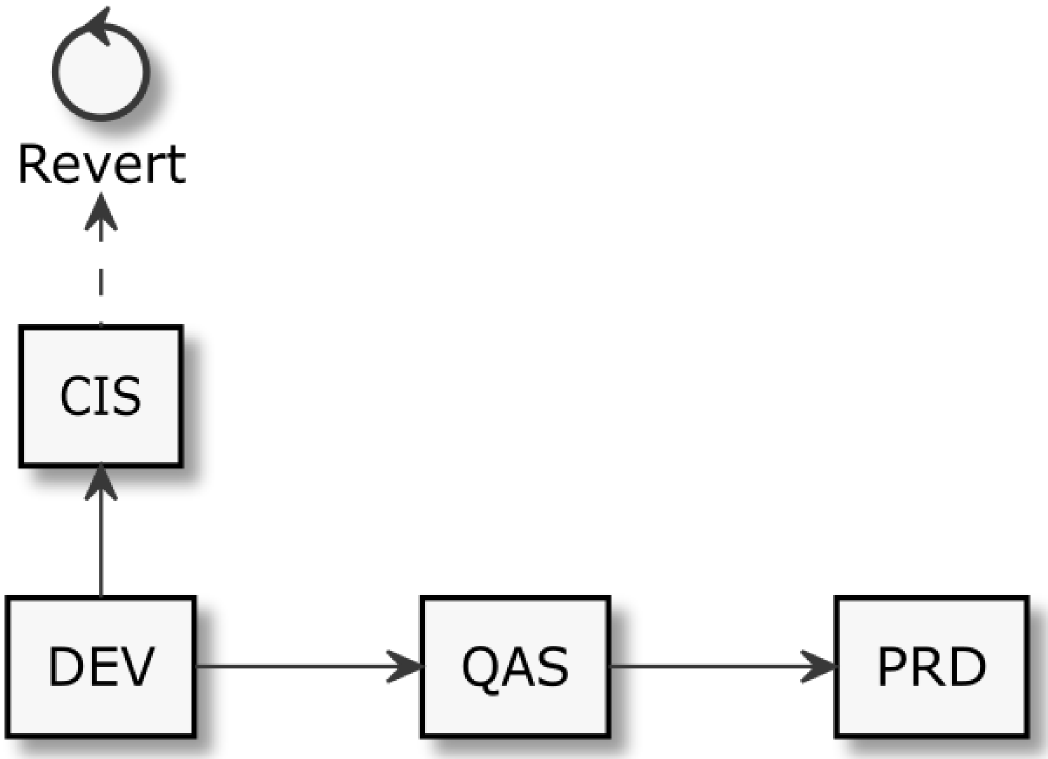
For ABAP systems its difficult to do a proper rollback of the objects. The unit and integration tests might change data in the system(they should not), so a full database rollback is required to have fully reliable results.

This means that the system cannot be used for manual testing at the same time, for each change the developers make the system is changed. Alternatively, specific time-slots are allocated for CI and some other for manual testing.

All-in-all this ends up with an automated process for breaking the QAS system, instead of avoiding errors via automation.

Scenario 3 – Add new CI system

So, to not disturb manual testing in the QAS system, the CI run can be moved to a different server. CI runs in CIS system, and manual testing is done in QAS,

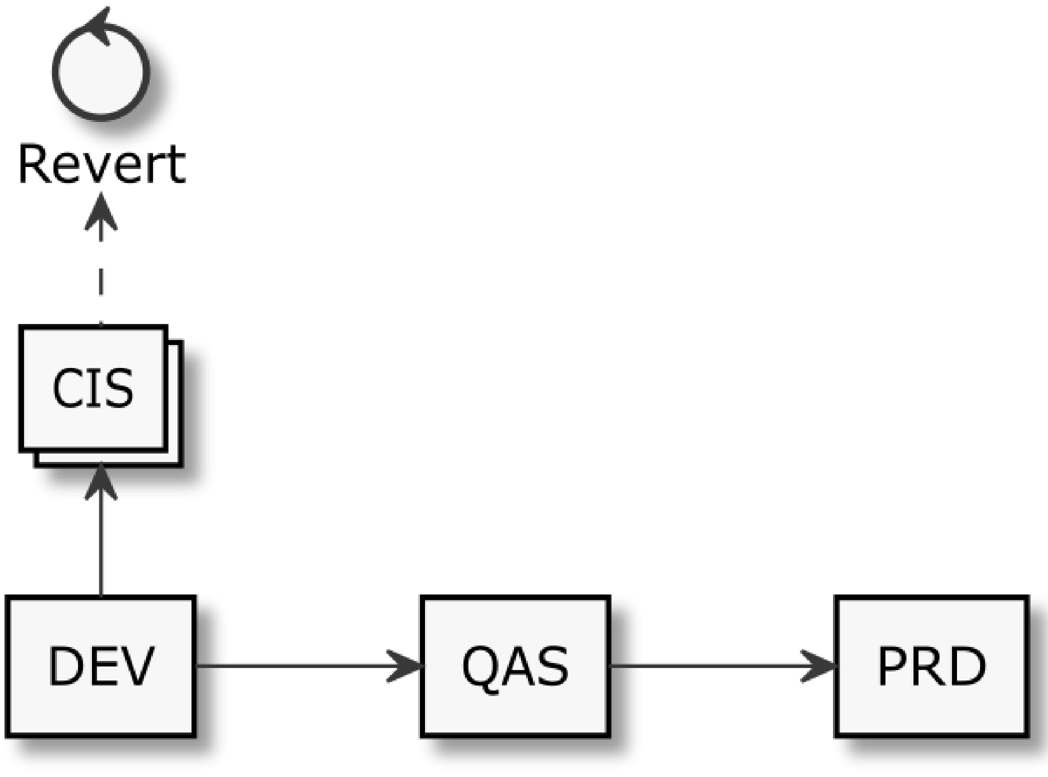


Scenario 4 – Scalable CI

In scenario 3, assuming CI is run for each change run one by one.

And assuming each CI run takes 10 minutes, and each developer does 30 changes per day, with 2 developers = 60 changes = 600 minutes = 10 hours.

It should be possible to determine the feedback for the change while developers go get coffee, so more CIS systems are required. Typically hardware is cheaper than wages, so one CIS system per developer is an easy solution, but expensive?

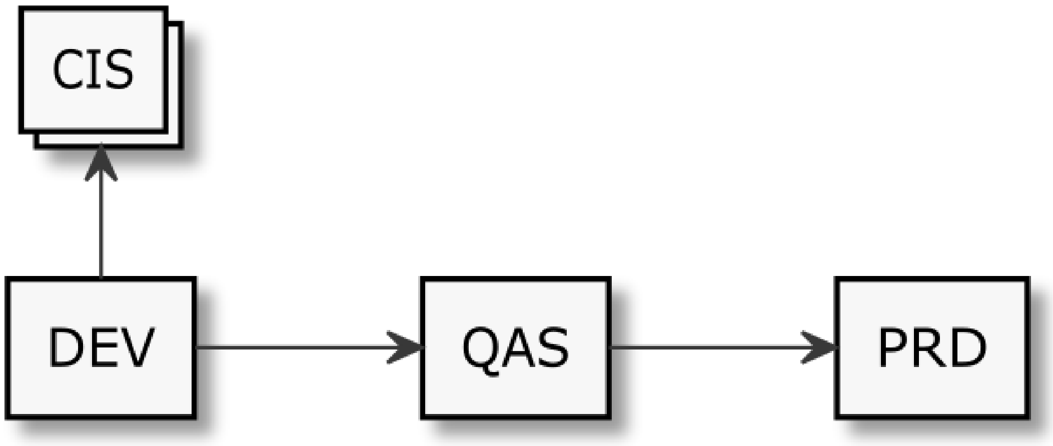


Scenario 5 – Containerization

Instead of having a lot of idle systems, spawn a new CIS system using containerization and deploy the changes, run CI.

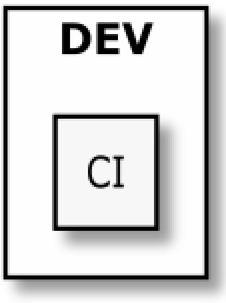
This way there will be a common baseline for everything, and no need to do any reverts, just spawning systems instead.

This can give reliable results in a timely manner, but I still have not seen anyone doing this setup yet, as its difficult to get everything running, and tough decisions have to be made to determine which baseline to use. If its a large container snapshot, hot standbys can be setup to respond fast.



Scenario 6 – Run inside DEV

Going in the other direction, setting up new infrastructure is time-consuming and expensive. How about running the CI inside the existing DEV system? This will sacrifice some reliability as the changes are not tested in isolation, but partner solutions exists that can help reduce this risk.

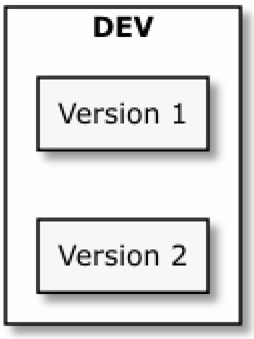


Scenario 7 – Multiple active versions

According to the slides, SAP has support for development and maintenance within the same system on the road-map.

This is possible today, but with multiple systems, so I assume the new innovation will be supporting this in a single system, and at the same time. So the ABAP kernel will have to support multiple active versions at a time. When(if?) this is implemented, the CI can run within the DEV system, just on the version with the changes done by the developer.

This, however, raises a lot of other questions regarding the maintenance of data, database isolation etc.



Guess this will never happen, instead what the slides refer to is probably a possibility to change between active versions. If there is more than one developer working in the area switched to development or maintenance on the same system, they will have to coordinate when to do one thing or another. I.e. it still makes sense to have separate development and maintenance systems.

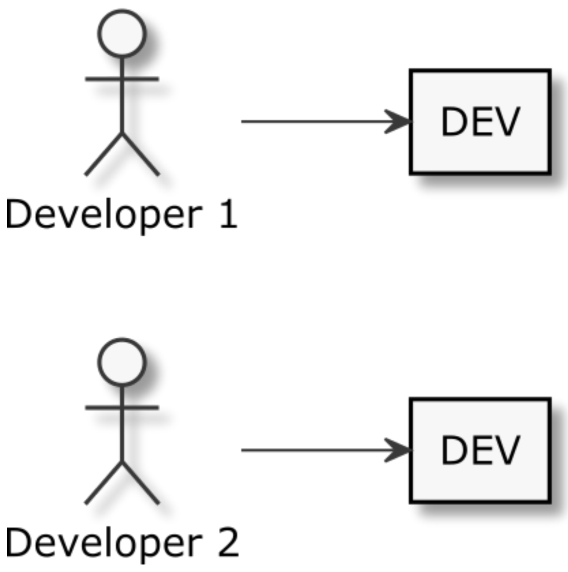
Scenario 8 – One system per developer

As suggested by Ethan Jewett back in 2016:

http://searchsap.techtarget.com/answer/How-can-I-use-Git-and-GitHub-for-SAP-software-development

http://searchsap.techtarget.com/tip/Implementing-modern-practices-in-an-ABAP-development-shop

Give one development system to each ABAP developer, this will help to do the development and unit tests in isolation. But still, CI should be run centrally on aligned infrastructure and configuration.



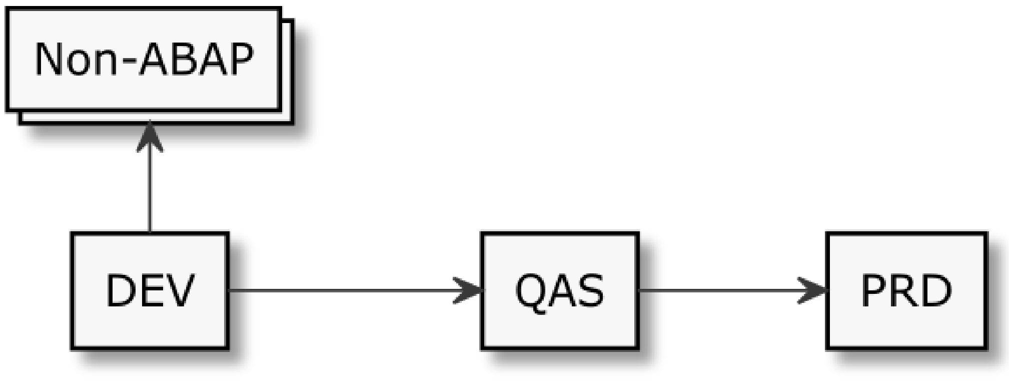
Scenario 9 – Front-load outside ABAP AS

Performing CI on ABAP application servers is required but might be an extensive setup.

For tasks like checking whitespace, a complete ABAP AS is not really required, multiple steps can be added in the CI pipeline, and once initial checks have been made, the ABAP CI run can be started.

abaplint is one free option that exists, it can run on Linux, so it hooks easily into GitHub Actions / GitLab Pipelines / whatever. If doing development in other languages than ABAP this is a typical setup for development, so it might be able to run on already existing infrastructure. Also note that this does not require any commercial licenses to run, so it can be used for open source development.

Results can be unreliable, but the setup is easy and cheap.



Conclusions

CI in ABAP is difficult, choose wisely among cost, reliability, speed, complexity etc. they have large consequences on the development process and infrastructure.

Fully supporting continuous integration is not just getting some results to the pipeline, like getting code into git is not devops.

ABAP containerization is required to fully support CI, without it it’s not possible to fully support continuous integration.

Alert Moderator

Assigned tags

ABAP Development | ABAP Testing and Analysis | abaplint |

Related Blog Posts

ABAP Continuous Integration Plugin now with Plug and Play
By **Andreas Gautsch** , Aug 20, 2018

ABAP Continuous Integration with Eclipse
By **Andreas Gautsch** , Sep 10, 2017

Eclipse beginners tour completed
By **Former Member** , May 25, 2014

Related Questions

Unable to get ABAP Continuous Integration Option Under Windows preference under ADT
By **saumohan mohanty** , Aug 05, 2019

Can anyone give me ABAP scenarios?
By **Former Member** , Apr 18, 2008

Please send me some screen shots on ABAP debugging on 4.6 c
By **Former Member** , Jan 07, 2008

1 Comment

Lars Hvam | Post author

February 20, 2020 at 1:36 pm

Also make sure to check out https://blogs.sap.com/2020/02/20/weve-finished-our-renovations-our-new-ci-cd-best-practices/

Some snippets from the help pages:

> *Thereby, you can detect errors as quickly as possible and prevent integration problems before completing the development.*
>
> *Continuously committing even smaller code changes into the main line and beginning to do so at an early stage of your development process is the key principle of continuous integration.*
>
> *Execute scenario tests on machines that are comparable to the final production landscape.*
>
> *Include fast-running tests directly into your CI builds and run more time-consuming ones later.*

Like (0) Reply Edit Alert Moderator Trash

Add Comment

Find us on