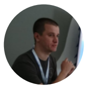
Technical Articles



Lars Hvam
December 28, 2020 2 minute read

So, what is CI?

Follow RSS feed Like

8 Likes 119 Views 0 Comments Edit

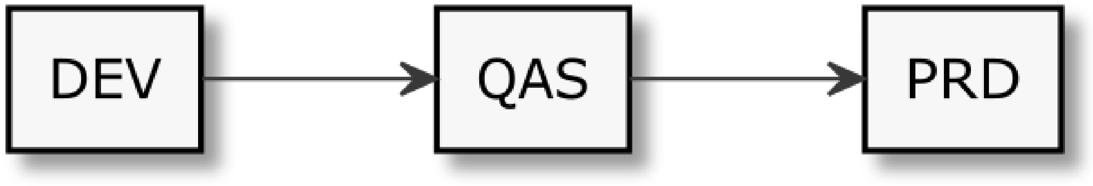
Continuous Integration keeps popping up every now and then, and there are already some great blogs on the subject. SAP also provides a great overview in What Are Continuous Integration and Continuous Delivery?, it describes CI as:

> *Continuous integration (CI) describes a software development process, in which various team members integrate their contributions frequently into a single main line. Before each integration, the changes are verified through builds and automated testing. Thereby, you can detect errors as quickly as possible and prevent integration problems before completing the development.*

Basically, CI helps the developer to verify the development *before* completing the development.

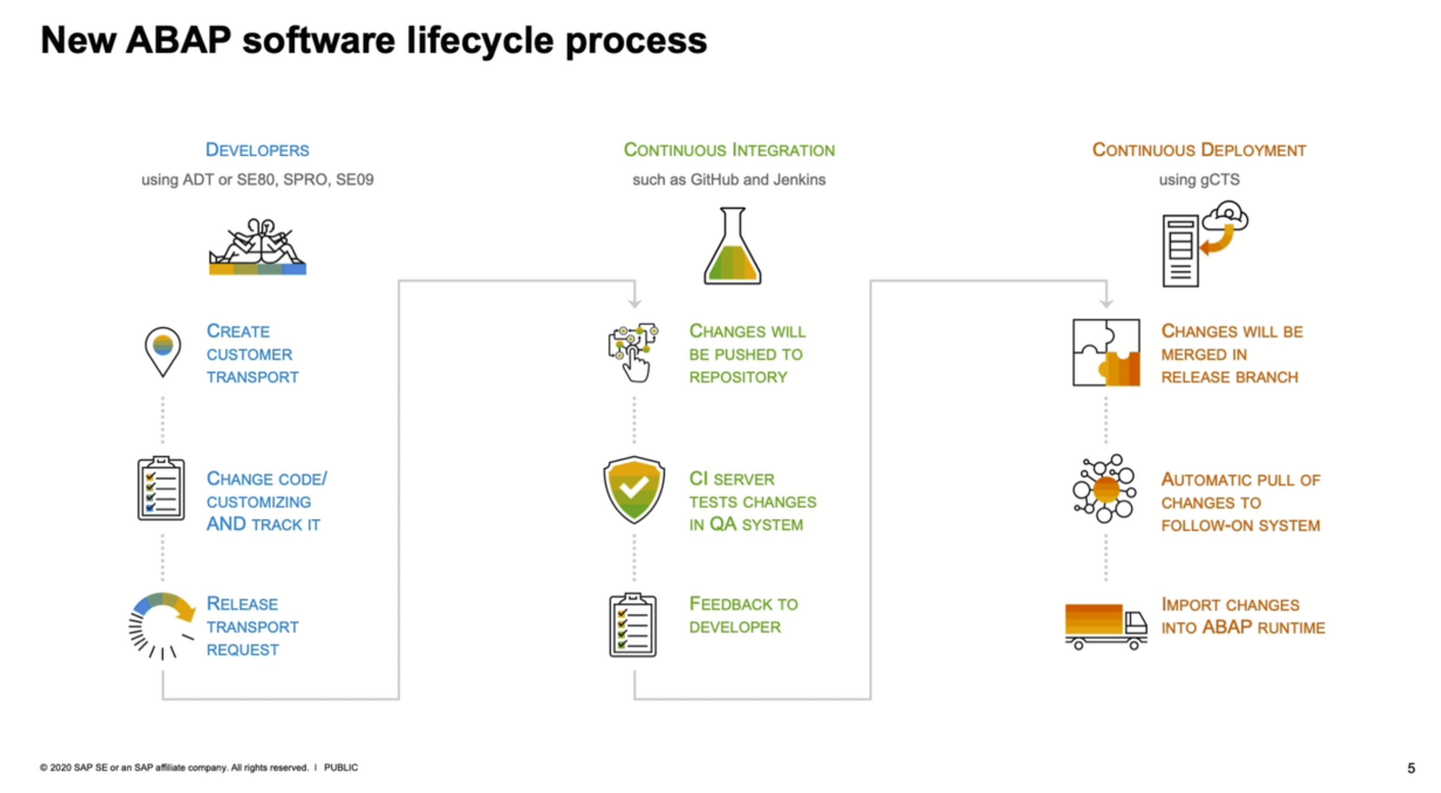
Landscape

In a typical SAP landscape, the developers perform changes in DEV, and *after* the development is done, the changes are moved to QAS for testing, and if successful the changes are moved to PRD.



Developers rely on having an operational DEV system for doing developments, and QAS must also work in order to be able to verify the development, which typically(?) involves manual steps.

At the recent TechEd, SAP recently suggested the following for a CI setup,



New ABAP software lifecycle process

DEVELOPERS
using ADT or SE80, SPRO, SE09

CREATE CUSTOMER TRANSPORT

CHANGE CODE/ CUSTOMIZING AND TRACK IT

RELEASE TRANSPORT REQUEST

CONTINUOUS INTEGRATION
such as GitHub and Jenkins

CHANGES WILL BE PUSHED TO REPOSITORY

CI SERVER TESTS CHANGES IN QA SYSTEM

FEEDBACK TO DEVELOPER

CONTINUOUS DEPLOYMENT
using gCTS

CHANGES WILL BE MERGED IN RELEASE BRANCH

AUTOMATIC PULL OF CHANGES TO FOLLOW-ON SYSTEM

IMPORT CHANGES INTO ABAP RUNTIME

© 2020 SAP SE or an SAP affiliate company. All rights reserved. ǀ PUBLIC

5

DEV208

Where the QA system is used to perform the CI tests, this corresponds to “Scenario 2” in Fully supporting CI in ABAP AS

Developers will Commit Early and Often, and each change will be built, triggering the CI, meaning:

1. The QA system will continuously be changed, and objects possibly rolled back. In ABAP rollbacks are not 100% reliable, e.g. a unit test might have modified the system state, thus leaving the QA system in an

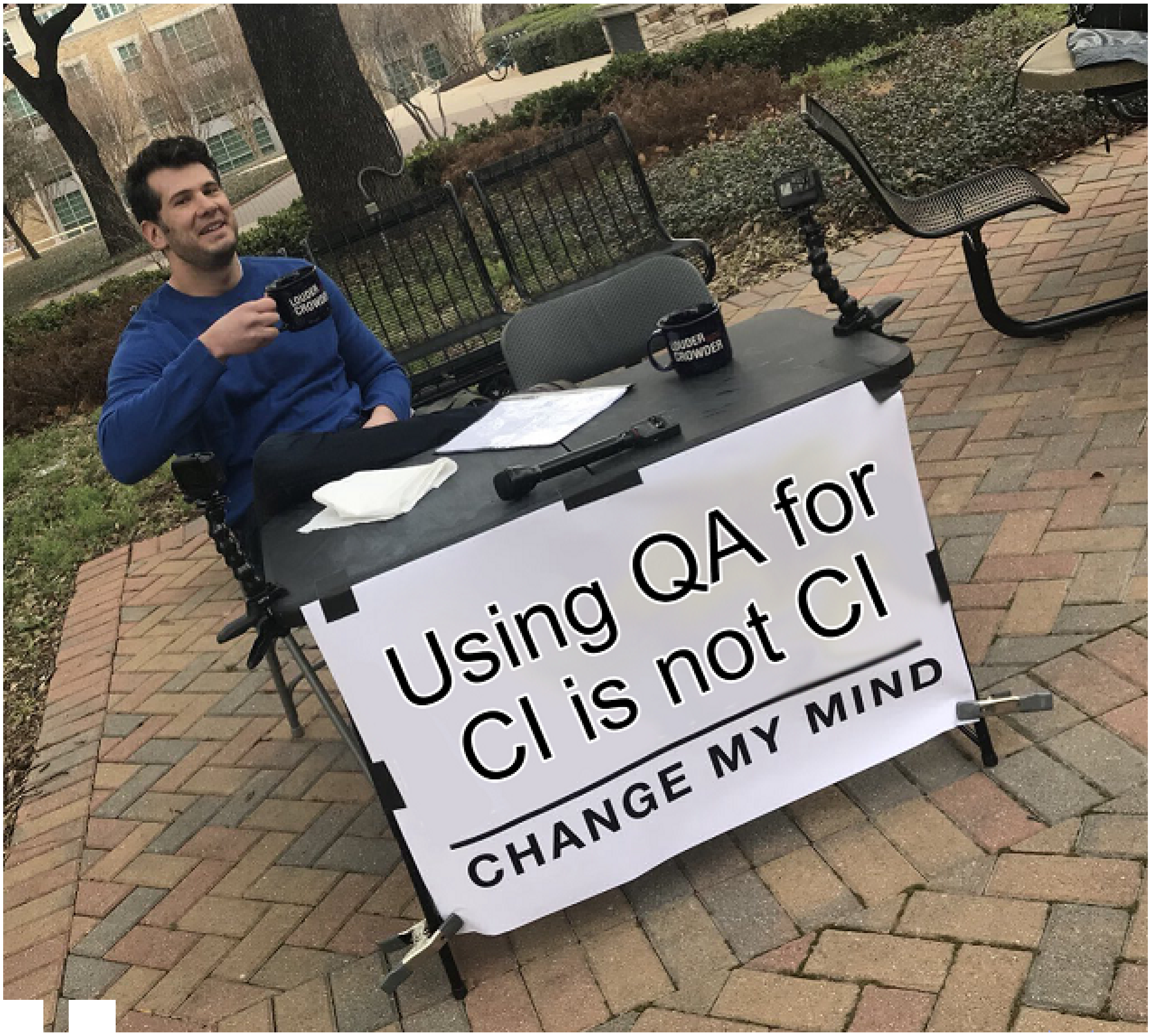
unknown state.

2. CI is triggered *before* the developments are done, possibly moving faulty code to QA, which has not been through any quality control steps.
3. Manual testing in QA will be void, as the system state continuously changes. However, this could work in a waterfall process, if all testing is done after all development, but it goes in the opposite direction of Continuous Deployment

All-in-all this ends up with a *harmful* automated process for breaking the QAS system, instead of avoiding errors via automation and helping developers.

Having void test results might be the right compromise for some organizations, but there are also multiple alternatives, like adding an extra system to handle CI.

Personally, I am moving towards a setup(Scenario 9), where the static analysis and unit tests are run outside of the ABAP stack, with the compromise of possibly giving false positives to the developer.



Alert Moderator

Assigned tags

ABAP Development | SAP TechEd | #ABAP |

Related Blog Posts

Fully supporting CI in ABAP AS

By **Lars Hvam** , Feb 20, 2020

What is RICEF?

By **Tuncay Karaca** , Jul 03, 2006

What is Maintenance View!

By **Tolga POLAT** , Oct 24, 2013

Related Questions

Enhancing the 0001 infotype second time

By **Former Member** , Dec 21, 2011

how many fields we can add in a CI include structure

By **Former Member** , Jan 07, 2010

Enhance Standard Infotype 1005 in HR

By **Subhash Karidhal** , Aug 31, 2011

Be the first to leave a comment

Add Comment

Find us on

Privacy	Terms of Use

Legal Disclosure	Copyright
Trademark	Cookie Preferences
Newsletter	Support