



ABAP Continuous Integration (CI) Plugin

Speeding up your (TDD) development process

Andreas Gautsch



andreas.gautsch@gmx.at



@andreas_gautsch



Parts of the presentation

1. **Why** a new plugin
2. **What** should be achieved
3. **How** looks it like

Continuous Integration (CI) Certification Test *)

Why?




1. Continuous commits [activations] to one repository
- 2a. Automated checks of Build and Unit Tests after each commit
- 2b. Automated static code checks [ATC]
3. Unit Test Failures [and ATC errors] are repaired within minutes

... and no CI Theatre

*) adapted, see original idea and definition by Martin Fowler at: <https://martinfowler.com/bliki/ContinuousIntegrationCertification.html>



Automated Continuous Integration saves time

Area	Without CI	Manual CI tools	With abapCI Plugin
source code change	Equal		
Minimal mouse clicks or shortcuts necessary*)	1 1	10 4	2 1
CI and quality components	None	AbapUnit ATC checks (Formatted code, Delete not used vars)	AbapUnit ATC checks (Formatted code Delete not used vars)
Project manager mood			

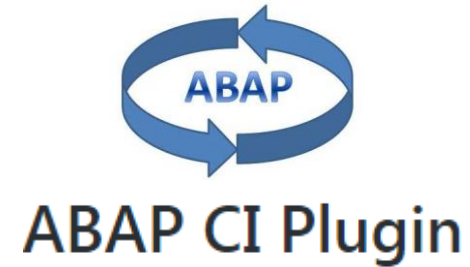
*) base for calculation: one changed ABAP class

The focus of the ABAP CI Plugin is on automation

What?

Target: Automatic execution of existing ABAP in Eclipse features

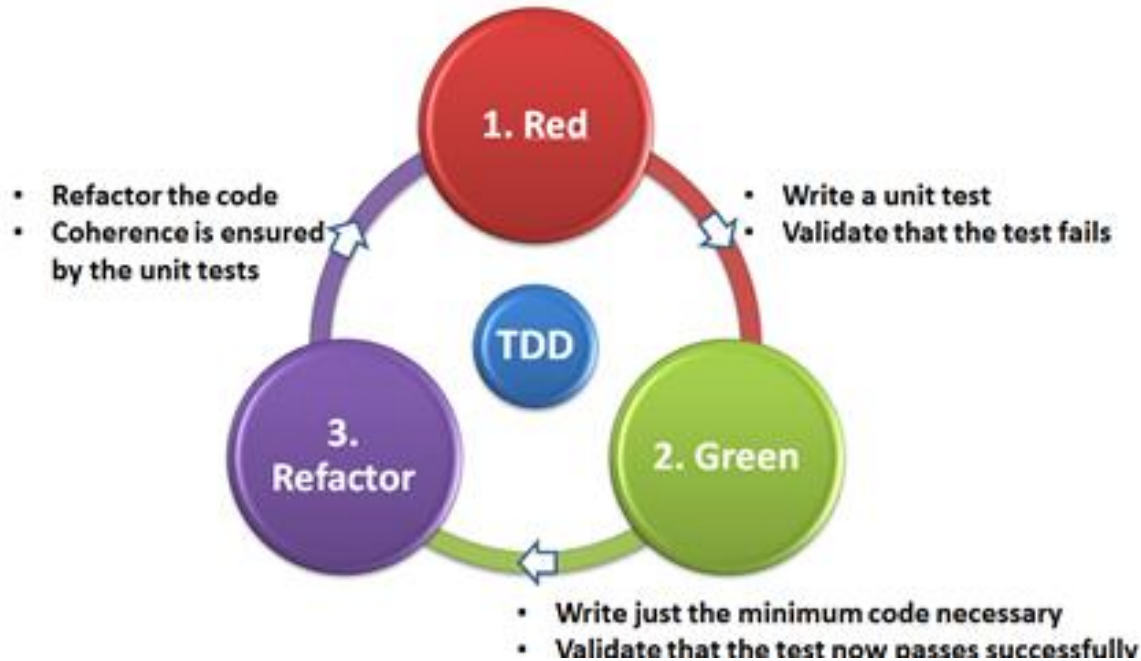
Purpose: Save development time and
Provide fast feedback about the code state



Target is primary the automatic visualisation of the actual state of the source code. For solving Unit test failures and cleaning up the particular ATC findings the standard ABAP in Eclipse views are much better and more powerful.



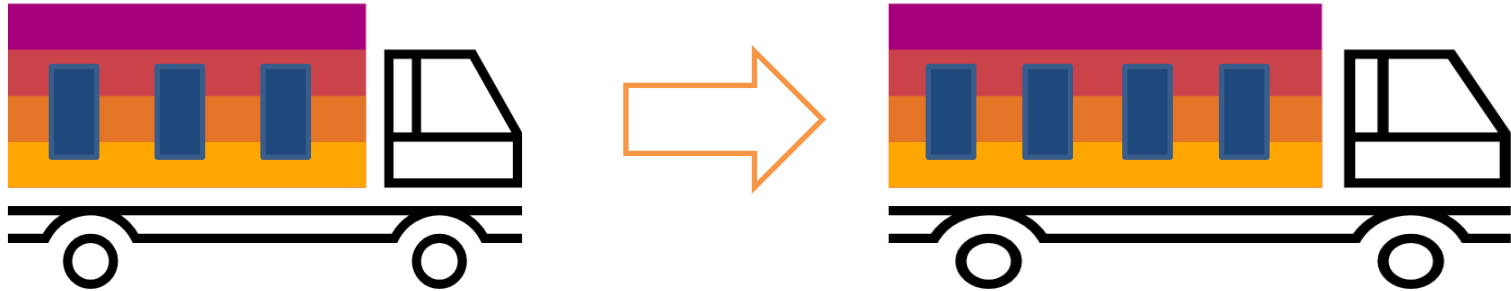
Test driven development



Coding example – Truck terminal

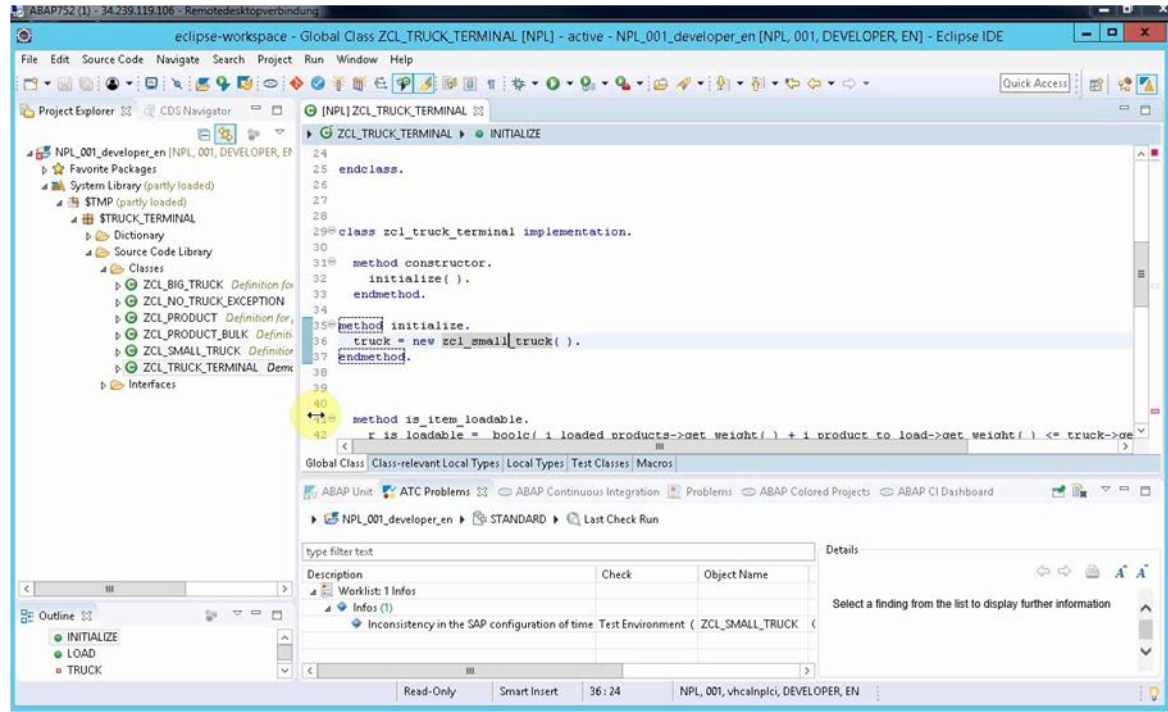
Use case: The maximal item **capacity can be increased from 3 to 4 items** as the new bought trucks have more loading space.

How?

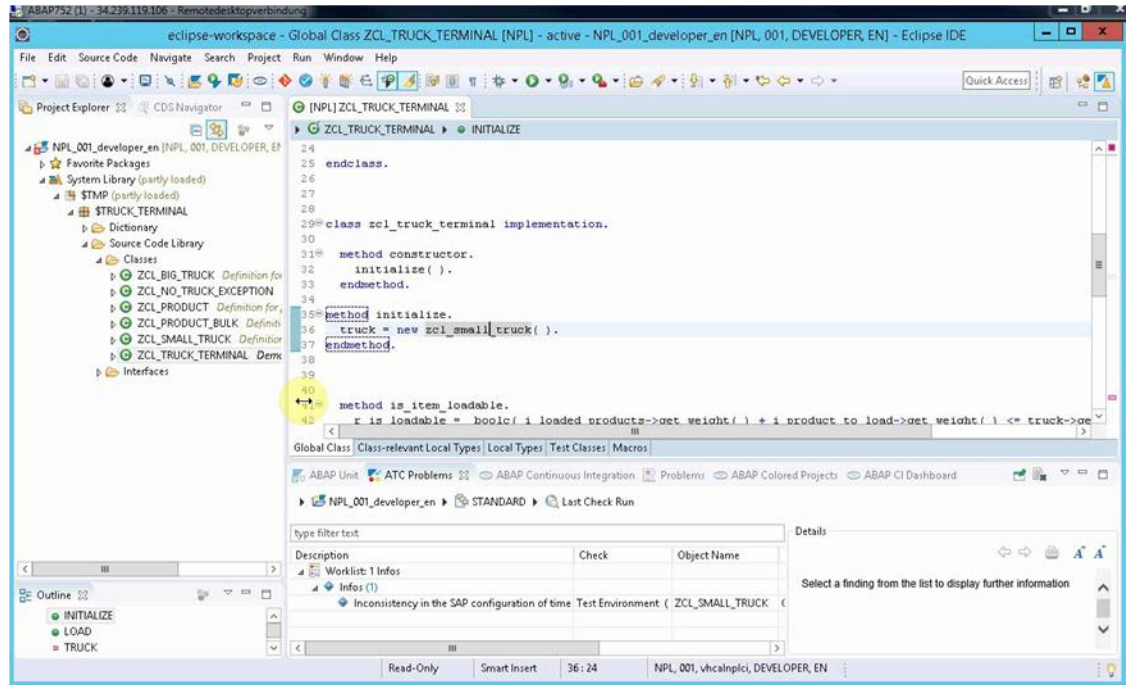


Images from <https://www.sap.com>

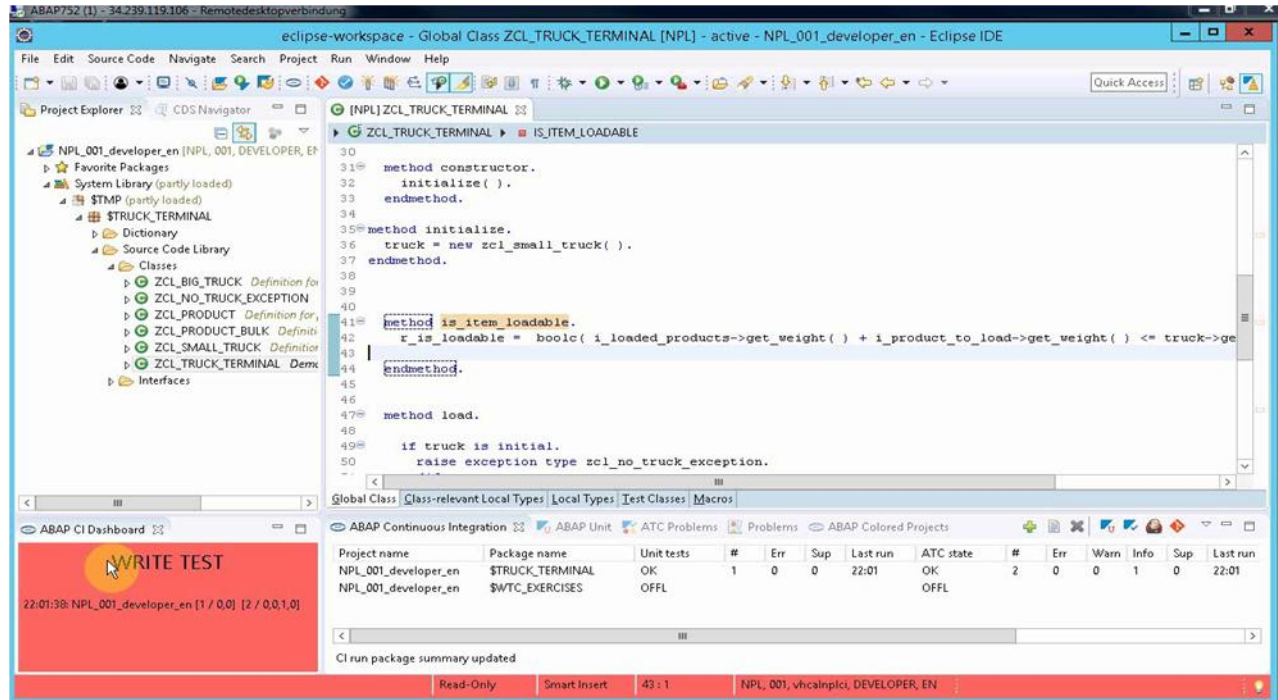
Skipping Continuous Integration is fast (... in the short time)



Manual usage of CI tools needs a lot of time






On purpose of ABAP CI plugin is to save time



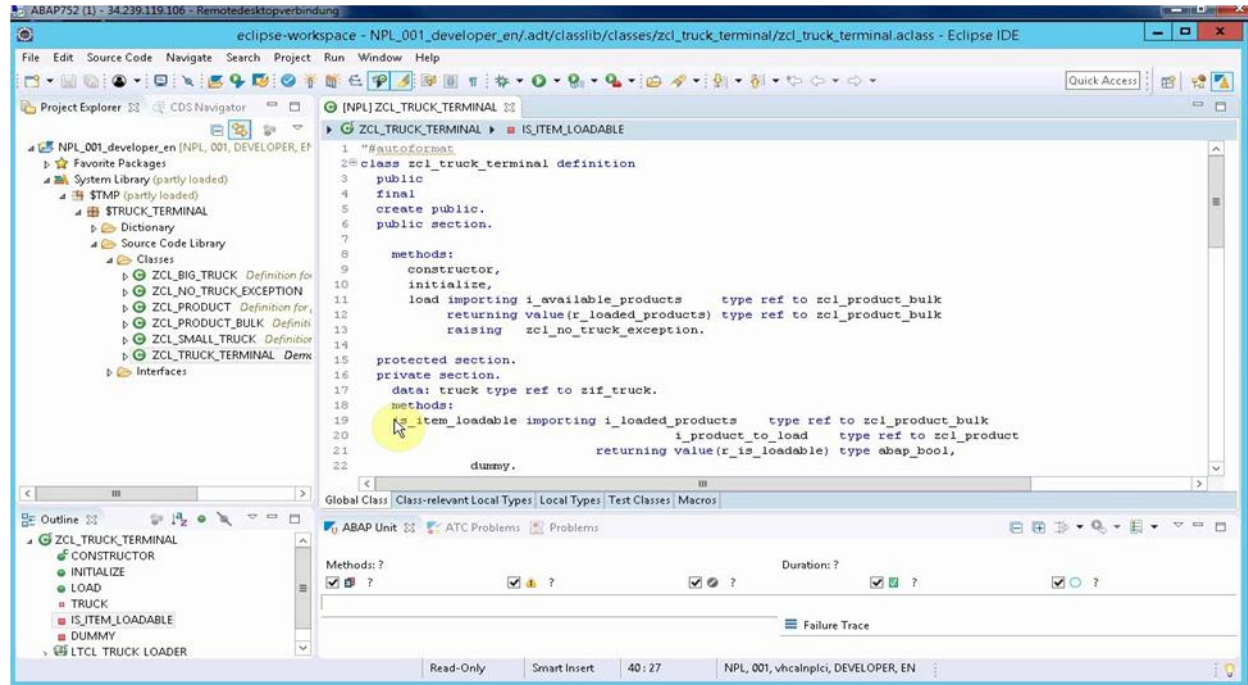
The screenshot shows the Eclipse IDE with the ABAP CI plugin. The main editor displays the source code of the ZCL_TRUCK_TERMINAL class. The ABAP CI Dashboard is visible at the bottom left, showing a 'WRITE TEST' button and a timestamp. The ABAP CI Unit Tests table is visible at the bottom right, showing test results for the NPL_001_developer_en project.

Project name	Package name	Unit tests	#	Err	Sup	Last run	ATC state	#	Err	Warn	Info	Sup	Last run
NPL_001_developer_en	\$TRUCK_TERMINAL	OK	1	0	0	22:01	OK	2	0	0	1	0	22:01
NPL_001_developer_en	\$WTC_EXERCISES	OFFL					OFFL						

Using Continuous Integration prevents errors

Area	No code quality	Manual quality tools	With abapCi Plugin
source code change	Equal		
Minimal mouse clicks or shortcuts necessary*)	2 1	10 4	2 1
CI and quality components	None	AbapUnit ATC checks (Formatted code, Delete not used vars)	AbapUnit ATC checks (Formatted code Delete not used vars)
Project Manager mood for 5 change requests			

Installing and configuring needs only 2 minutes



There are a lot of customization options

ABAP CI

General settings for ABAP Continuous Integration

1. Automatic Unit test runs

☒ Run Unit tests after an ABAP object was activated

☒ Run Unit tests for the activated ABAP objects, not for whole package

2. Automatic ATC runs

This feature writes data into the table SATC_AC_RESULTVT, reorganize the table with program SATC_AC_CLEANUP continuously!

☒ Run ATC check for activated ABAP objects

Run ATC with variant:

3. Visualisation of SourceCode State on UI

☒ Show TDD Labels on ABAP CI Dashboard

Backgroundcolor for OK Unit Test State used in ABAP CI dashboard

Backgroundcolor for FAIL Unit Test State used in ABAP CI dashboard

Backgroundcolor for FAIL ATC State used in ABAP CI dashboard

☒ Use test state color as background color of the Eclipse statusbar

☐ Change Theme layout on failed tests (do not use)

☒ Show a dialog when a new package for the CI Run is activated

4. Different coloring for each ABAP project

☐ Change color of tab header for colored projects

☒ Change color of left ruler for colored projects

☒ Change color of right ruler for colored projects

☐ Change color of status bar for colored projects

ABAP CI

General settings for ABAP Continuous Integration

1. Automatic Unit test runs

☒ Run Unit tests after an ABAP object was activated

☒ Run Unit tests for the activated ABAP objects, not for whole package

2. Automatic ATC runs

This feature writes data into the table SATC_AC_RESULTVT, reorganize the table with program SATC_AC_CLEANUP continuously!

☒ Run ATC check for activated ABAP objects

Run ATC with variant:

3. Visualisation of SourceCode State on UI

☒ Show TDD Labels on ABAP CI Dashboard

Backgroundcolor for OK Unit Test State used in ABAP CI dashboard

Backgroundcolor for FAIL Unit Test State used in ABAP CI dashboard

Backgroundcolor for FAIL ATC State used in ABAP CI dashboard

☒ Use test state color as background color of the Eclipse statusbar

#autoformat

backend)

<jenkins_host>:<jenkins_port>

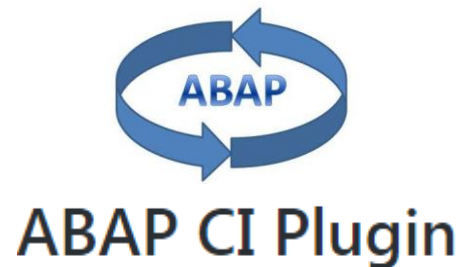
jenkins_username

jenkins_password

PUT_YOUR_OWN_TOKEN_FOR_ABAP_

Thanks for your attention!

**Purpose: Save development time and
Give fast feedback**



Installation with:



Open Source
and documentation:



Question, Issues, Feature or Pull Requests welcome!



Appendix I

- Open Source Github repository for ABAP CI Plugin
<https://github.com/andau/abapCI>
- Listing in Eclipse Marketplace
<https://marketplace.eclipse.org/content/abap-continuous-integration>
- Continuous Integration by Martin Fowler
<https://martinfowler.com/articles/continuousIntegration.html>
- Article about CI theatre
<https://www.gocd.org/2017/05/16/its-not-CI-its-CI-theatre/>
-
- ATC – Blogs by , for example:
<https://blogs.sap.com/2017/02/27/remote-code-analysis-in-atc-for-developers/>



Appendix II

- ABAP Test Cockpit (ATC) - Introduction Video on the ABAP channel on YouTube :
<https://youtu.be/SmSJ8Tkl8w>
- ABAP Open Checks by Lars Hvam
<https://github.com/larshp/abapOpenChecks>

