



ABAP Continuous Integration (CI) Plugin

Speeding up your (TDD) development flow

Andreas Gautsch



andreas.gautsch@gmx.at



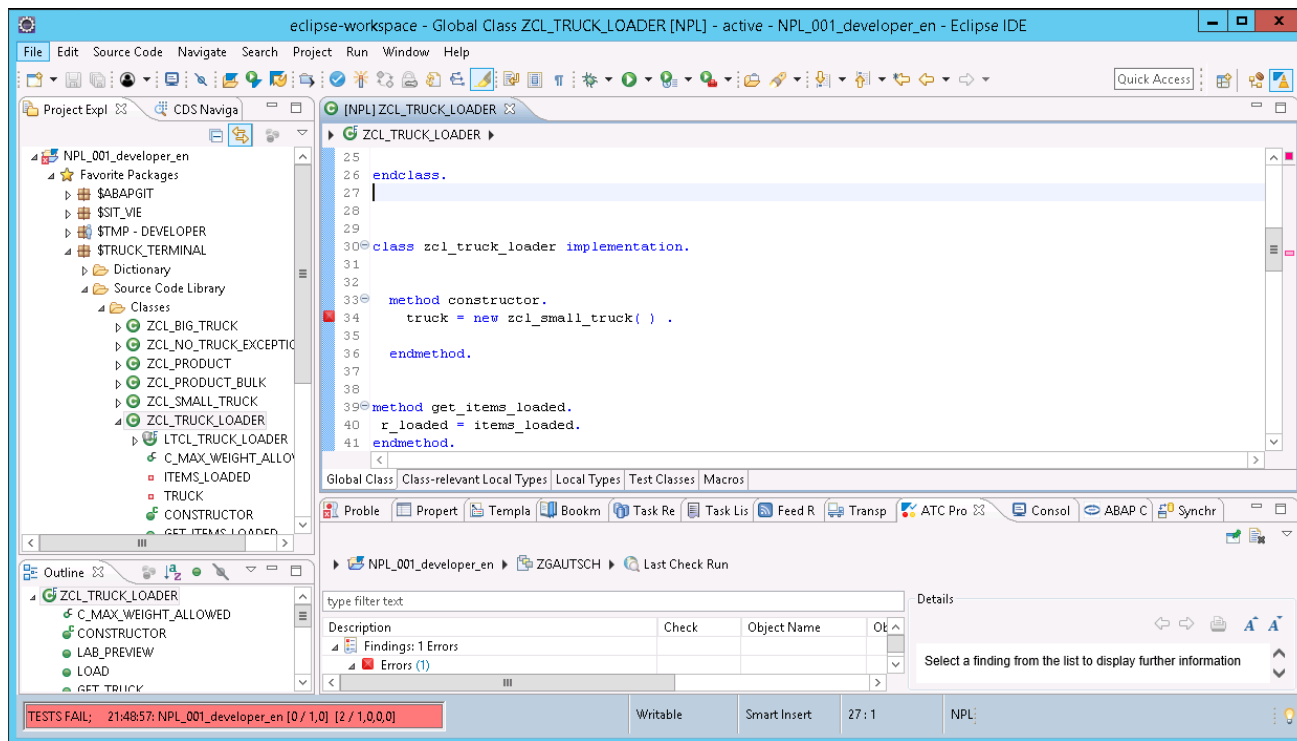
[@andreas_gautsch](https://twitter.com/andreas_gautsch)

Parts of the presentation

- 1. Why** an additional plugin
- 2. What** the plugin achieves
- 3. How** it works



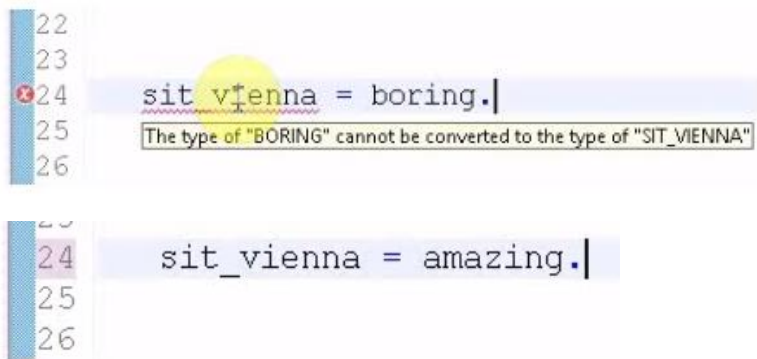
Screenshot ABAP Continuous Integration Plugin



Main benefits of an IDE like Eclipse

Why?

Immediate Feedback



Time savings

by bundling repetitive task blocks



Activation

There are a lot of perfectly automated features in ABAP in Eclipse,
but there is always something to improve ...




Immediate feedback with Continuous Integration

Continuous Integration (CI) Certification Test

1. Continuous activations [commits] to the central development system
- 2a. Every activation triggers a build check and **Automated Unit Tests**
- 2b. Every activation triggers an automatic check with ABAP Test Cockpit [ATC]
3. Unit Test Failures [and ATC errors] are repaired within a short period of time

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

Saving time by automating repetitive tasks

Area	Without CI	Manual CI tools	With abapCI Plugin
source code change	Equal		
Minimal mouse clicks or shortcuts necessary*)	1 1	10 4	1 1
CI and source code quality	None	AbapUnit ATC checks Source code formatting (Delete not used vars)	AbapUnit ATC checks Source code formatting (Delete not used vars)
Developer mood			

*) base for calculation: one changed ABAP class

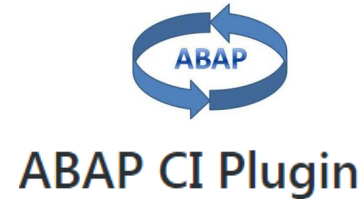
ABAP CI Plugin automates existing ADT features

Existing ADT features:

Unit test run
ABAP Testcockpit (ATC)
Pretty Printer



are called by

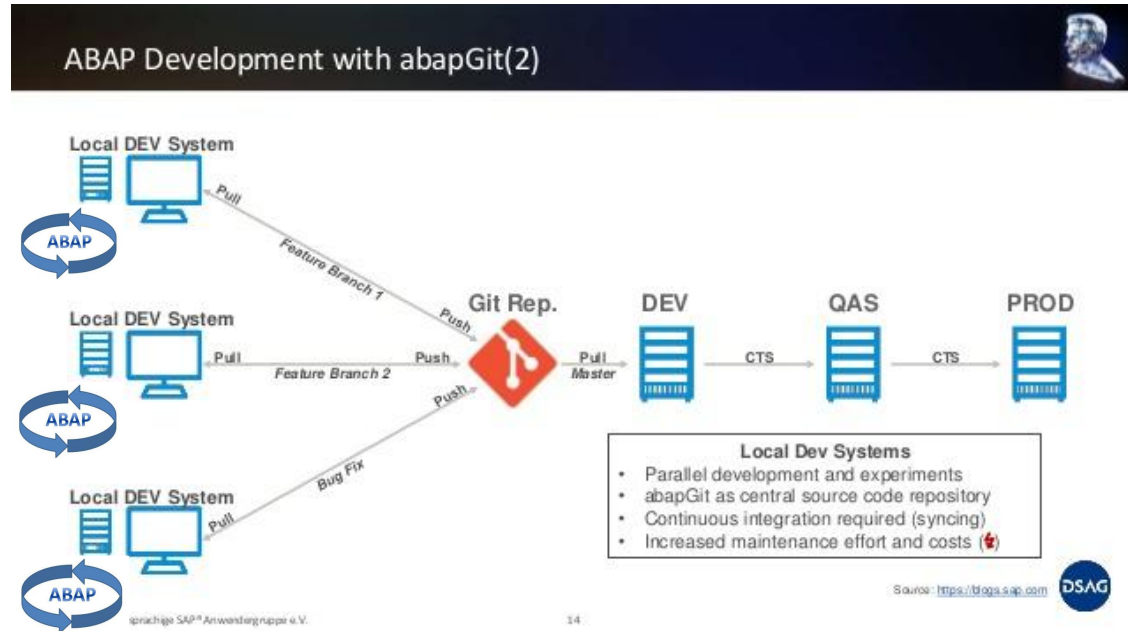


What?

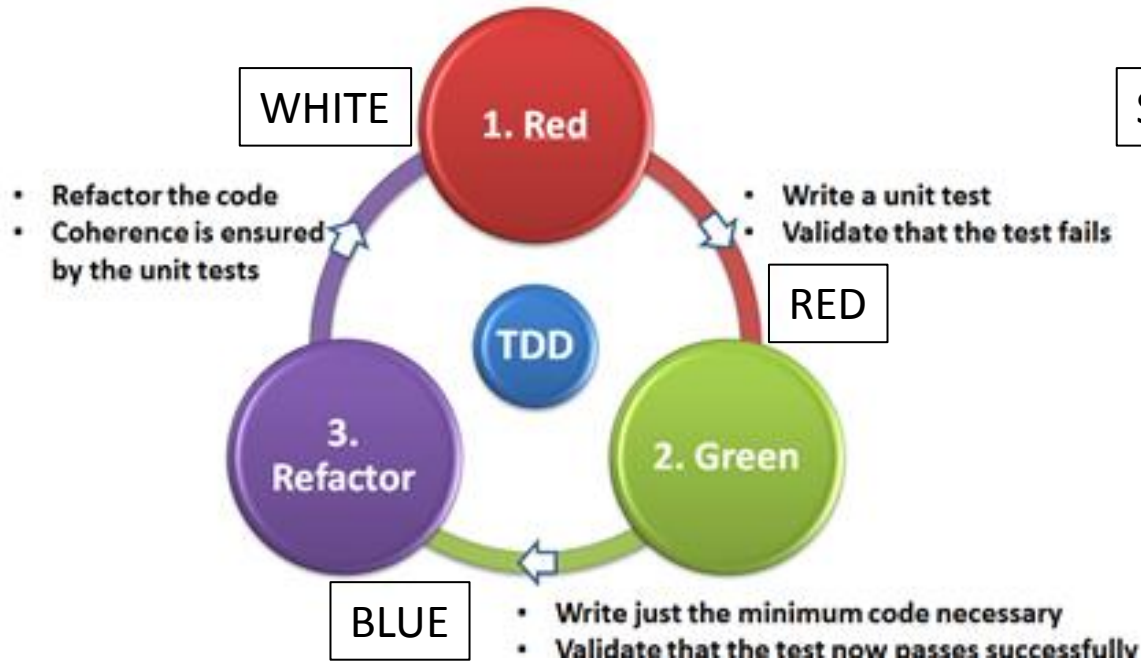
Target is only the automatic visualisation of the actual source code state after each activation by using existing ABAP in Eclipse features. For solving unexpected Unit test failures and cleaning up complicated ATC findings the powerful standard ABAP in Eclipse views are much better.



In a full CI/CD strategy the plugin serves as local CI system

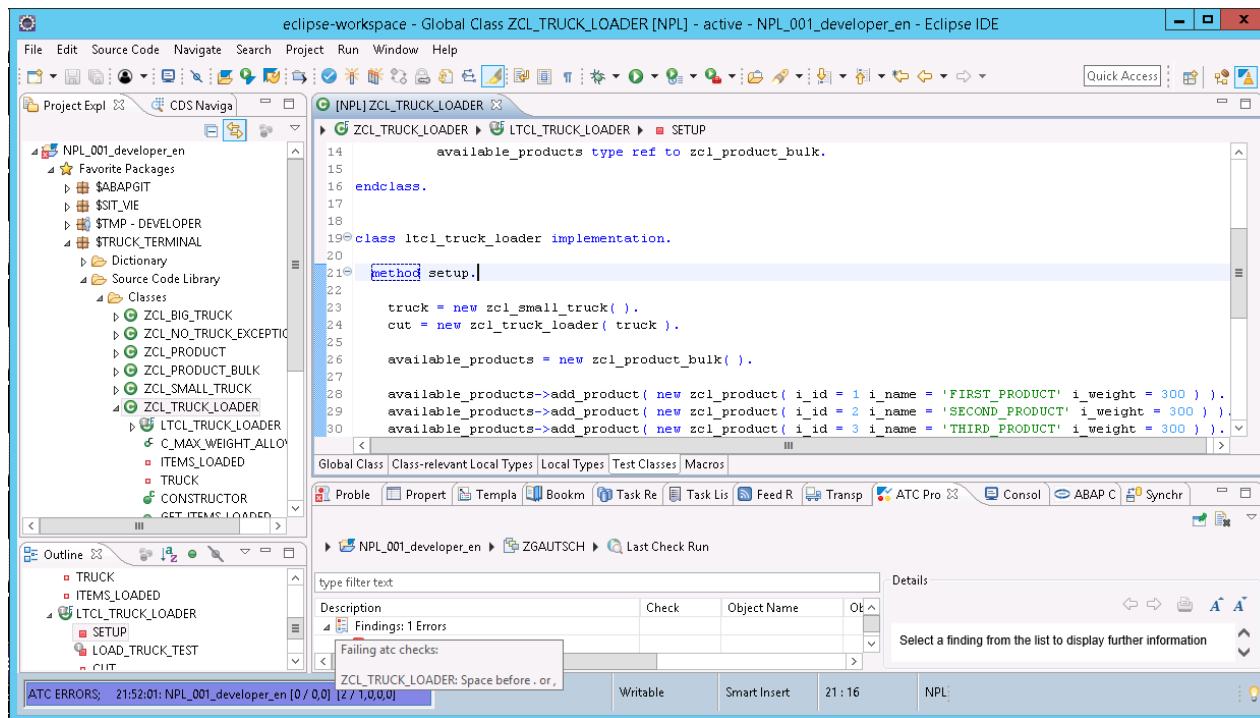


Immediate feedback with Test Driven Development



Visualisation of the
SOURCE CODE STATE
in this presentation

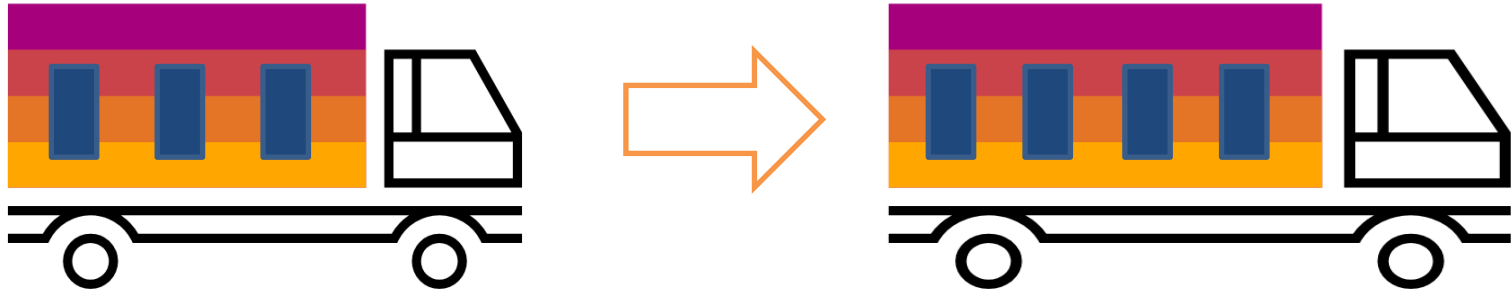
Displaying the actual state of the source code



Coding example – Truck loading 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?



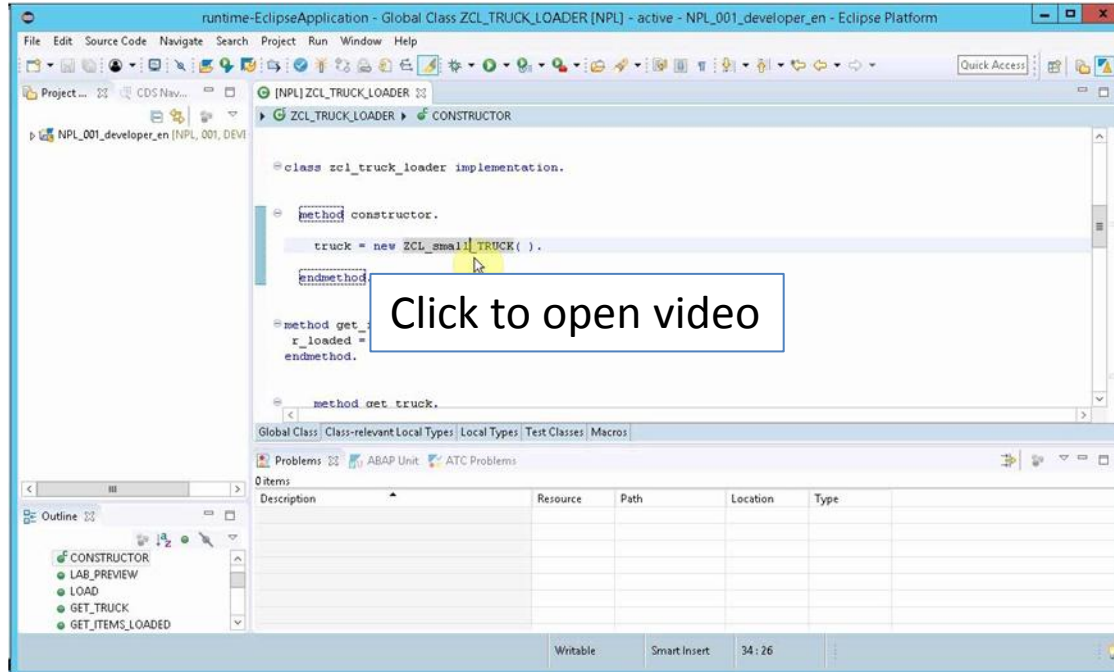
Images from <https://www.sap.com>

Automated Continuous Integration saves time

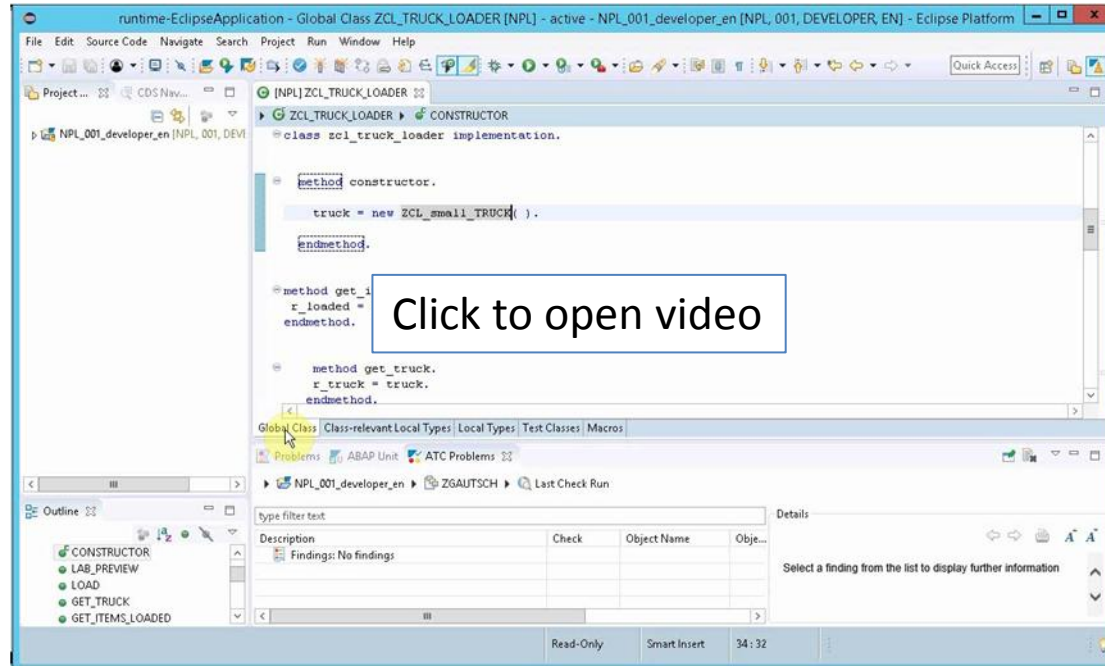
Area	Without CI	Manual CI tools	With abapCI Plugin
source code change	A	B	C
Minimal mouse clicks or shortcuts necessary*)	1 1	10 4	2 1
CI and source code quality	None	AbapUnit ATC checks Source code formatting (Delete not used vars)	AbapUnit ATC checks Source code formatting (Delete not used vars)
Developer mood			

*) base for calculation: one changed ABAP class

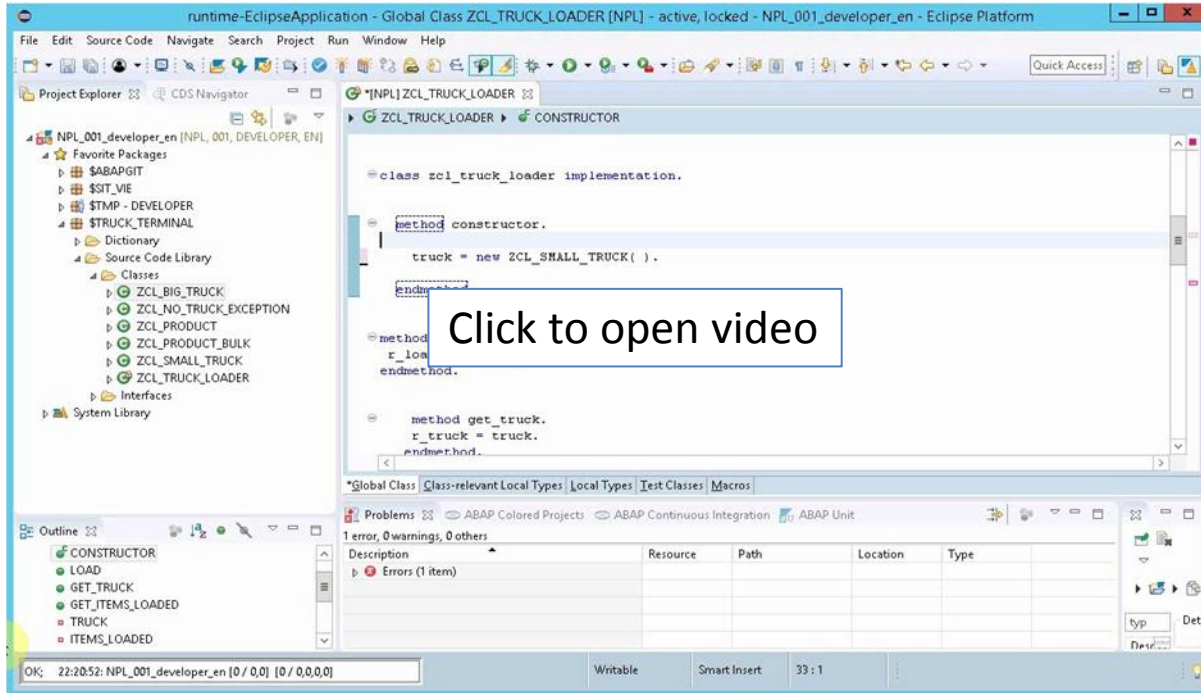
A - Skipping Continuous Integration is fast (... in the short time)






B - Manual usage of CI tools needs a lot of time



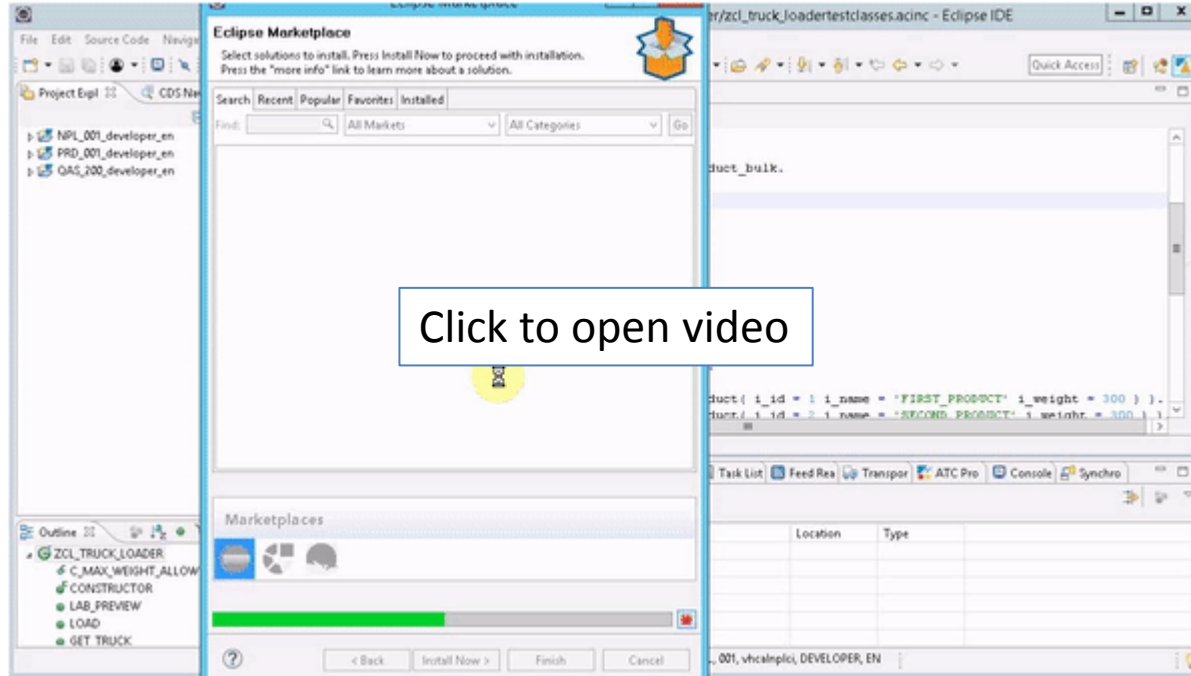
C – With ABAP CI plugin the CI tools are faster



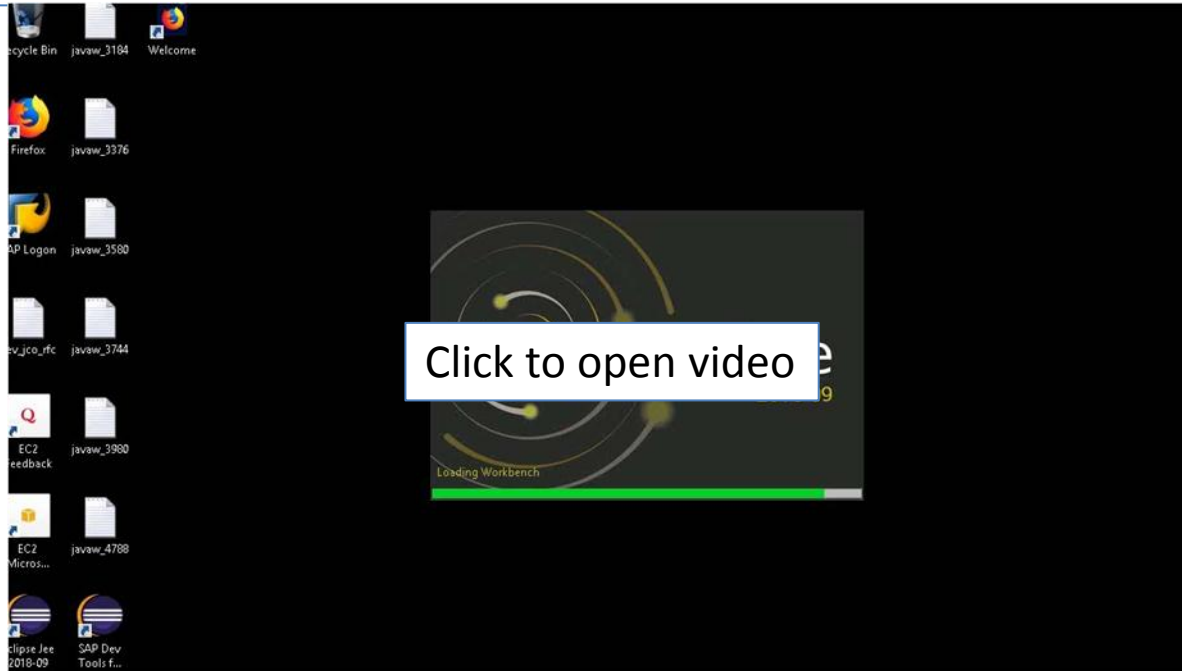
Continuous Integration tools avoid coding errors

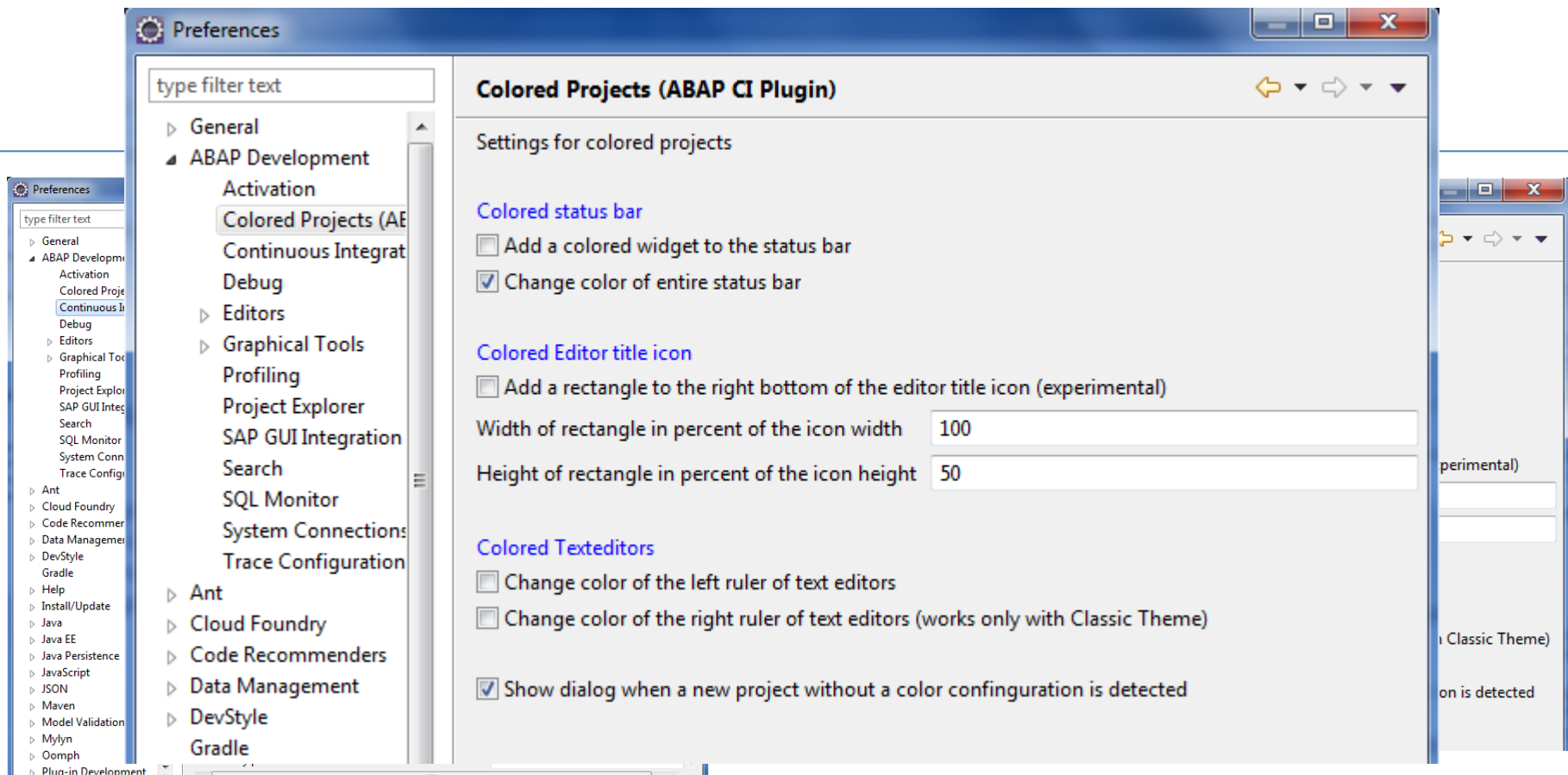
Area	No code quality	Manual quality tools	With abapCi Plugin
source code change	Equal		
Minimal mouse clicks or shortcuts necessary*)	1 1	10 4	1 1
CI and source code quality	None	AbapUnit ATC checks Source code formatting (Delete not used vars)	AbapUnit ATC checks Source code formatting (Delete not used vars)
Developer mood for a couple change requests			

Installing and configuring needs only 2 minutes



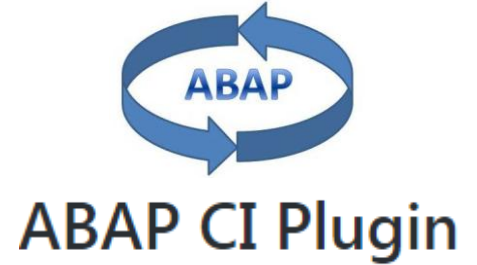
Installing and configuring needs only 2 minutes





Thanks for your attention!

Purpose: Save development time and giving immediate feedback





Installation with:



Open Source
and documentation:



 andreas.gautsch@gmx.at
 @andreas_gautsch

Questions, 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>