



## **TDD mit ABAP Units**

Webinar, 23.09.2016, 10:00 Uhr



# Johann Föbleitner

Senior Consultant at Cadaxo

eMail: [johann.foessleitner@cadaxo.com](mailto:johann.foessleitner@cadaxo.com)

Twitter: [@foessleitnerj](https://twitter.com/foessleitnerj)

## Beratungsschwerpunkte

- Konzeption & Management v on SAP Entwicklungsprojekten
- Qualitymanagement & Performanceoptimierung
- Clean Code
- SAP Fiori/SAPUI5



## Domi Bigl

Senior Consultant at Cadaxo

eMail: [dominik.bigl@cadaxo.com](mailto:dominik.bigl@cadaxo.com)

Twitter: [@DomiBiglSAP](https://twitter.com/DomiBiglSAP)

### Beratungsschwerpunkte

- Konzeption & Management v on Entwicklungsprojekten
- Qualitymanagement & Performanceoptimierung
- ABAP Units
- SAP Fiori, SAP UI5

cadaxo

- **Tests**
- **Unit Tests**
- **ABAP Units + Demo**
- **TDD**
- **TDD mit ABAP Units – Demo**

# Tests

4. Juni 1996

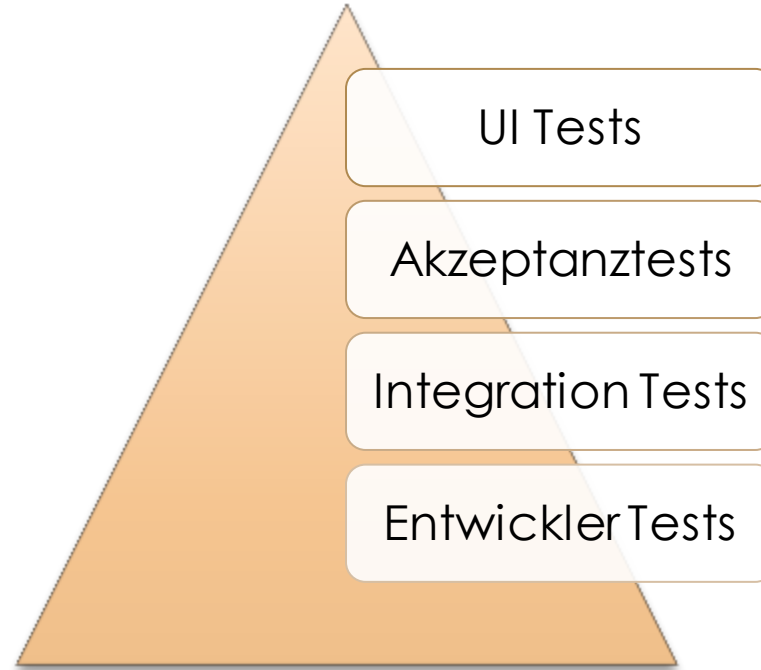
- **Explosion Ariane 5**
  - 36,7 Sekunden nach dem Start
  - Ursache
    - Arithmetischer Überlauf bei der Umwandlung einer 64-Bit-Gleitkomma-Zahl in eine 16-Bit-Ganzzahl – der Wert der horizontalen Geschwindigkeit.
    - Lenksystem brach zusammen und löste eine Selbstzerstörung aus
  - Hintergrund
    - Software stammt von Ariane 4. Ariane 5 flog jedoch schneller
    - Schaden ca. 500 Millionen Dollar für Rakete und 4 Satelliten
  - [https://www.youtube.com/watch?v=PK\\_yguLapgA](https://www.youtube.com/watch?v=PK_yguLapgA)

# Es geht ums Geld



Fehlerhafter/schlechter Code kostet!





- **Traditioneller Test eines ABAP Entwicklers**
  - /H, Debugging
  - SE80, SE24, SE37, SE38, Workbench ... und dann Testen
  - Testprogramme

- **Nachteile der traditionellen Vorgehensweise**
  - Zeitaufwendig
  - Nicht reproduzierbar
  - Unkoordiniert
  - Unprofessionell

- **Ein professioneller Entwickler**
  - Sorgt dafür, dass nur funktionierende bzw. ordentlich getestete Software an die Testteams übergeben wird

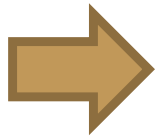
# Unit Tests

## Umfrage

- **Unit Tests bzw. xUnit Frameworks**
  - Mit Unit Tests werden einzelne Units (Methoden, Funktionbausteine) automatisiert getestet
  - Ziel der Unit Tests ist die Isolation aller Parts eines Programmes und die Prüfung ob alles ordnungsgemäß läuft

- **Ziel von Unit Tests**

- Probleme/Bugs im Developmentprozess so früh wie möglich entdecken
- Probleme/Bugs nach Upgrades etc. rasch entdecken
- Reduktion langer Debugging Sessions
- Reduziert die „Hemmschwelle“ schlechtes Coding zu optimieren -> Clean Code!



## **ABAP Unit**

- **ABAP Units**

- Sind erstmals mit 6.40 aufgetaucht
- 100% Integration in die ABAP Entwicklungsumgebungen
- Fortlaufende Weiterentwicklung durch SAP



# Live Demo

Wir erstellen einen ABAP Unit Test

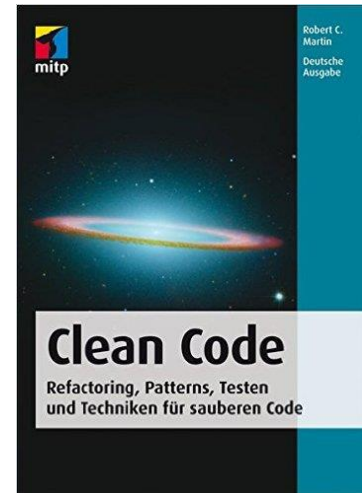
**Lesbar**

**Wartbar**

**Vertrauenswürdig**

- **Lesbare Tests**

- Unlesbare Tests sind (fast) völlig wertlos
- Tests werden von anderen Entwicklern gelesen
- Überlegte Namensgebung von Unit Tests, Variablen, ...
- Clean Code



ISBN 978-3-8266-5548-7

- **Wartbare Tests**

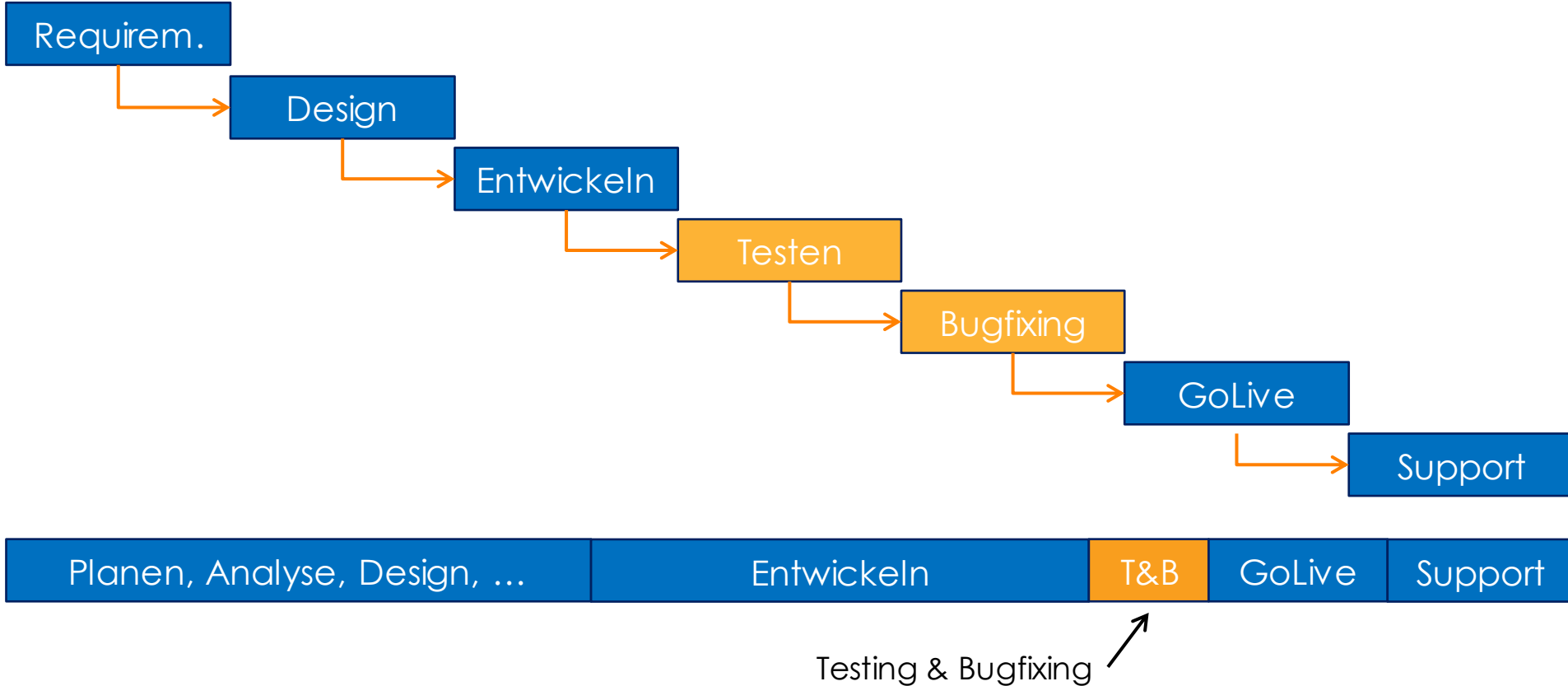
- Unwartbare Tests sind ein Albtraum, können Projekte negativ beeinflussen
- Entwickler werden Tests, deren Anpassungen zu lange dauern, einfach nicht mehr warten bzw. aktualisieren
- Keine Logik in Unit Tests aufnehmen
  - Ein IF oder CASE deutet schon auf Logik hin

- **Vertrauenswürdige Tests**
  - Ein Test ist dann vertrauenswürdig, wenn er uns das Gefühl gibt zu wissen was vorsich geht

- **Unit Tests** – FIRST Prinzip
  - **Fast** – Tests sollen schnell sein
  - **Independent** – Tests sollen nicht von einander abhängen
  - **Repeatable** – Jede Umgebung (Dev, QA, Prod, ... )
  - **Self-Validating** – der Test wird bestanden oder er scheitert
  - **Timely** – Tests müssen rechtzeitig geschrieben werden

# TDD

## Umfrage





- **Probleme des Wasserfallmodells**

- Auf neue bzw. geänderte Anforderungen kann nur schlechte oder gar nicht reagiert werden
- Phasen können oft nicht so klar getrennt werden
- Rückschritte zur vorigen Phase oft nicht vermeidbar
- Fehler werden unter Umständen erst spät erkannt und können nur durch großen Aufwand behoben werden

# Jetzt sind wir alle agil!



- **Eigenschaften von agiler Softwareentwicklung**
  - Anforderung sind unvollständig oder ändern sich
  - Iteratives Vorgehen, inkrementelle Entwicklung
  - Regelmäßige Reflexion
  - Frühzeitige Ergebnisse
  - Regelmäßiges Testen

Individuals and interactions  
over processes and tools

Working software  
over comprehensive documentation

Customer collaboration  
over contract negotiation

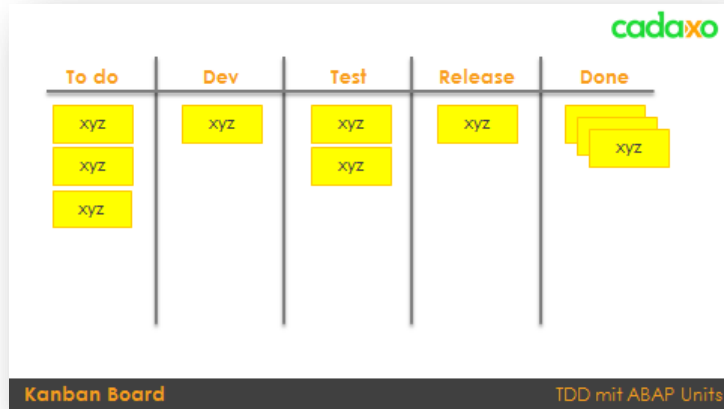
Responding to change  
over following a plan

Februar 2001: Robert C. Martin , Kent Beck, Ward Cunningham, Ron Jeffries, Martin Fowler ...

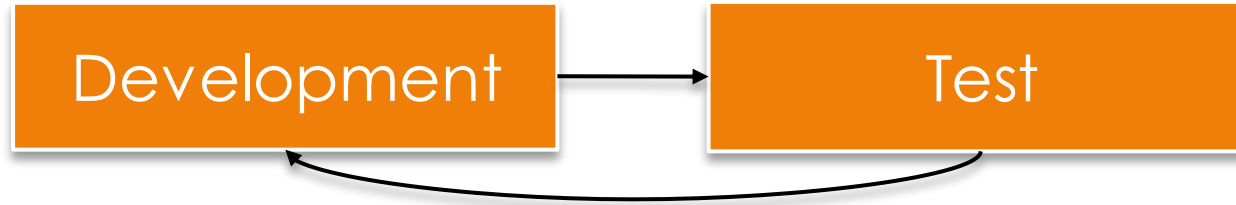
Kanban

Scrum

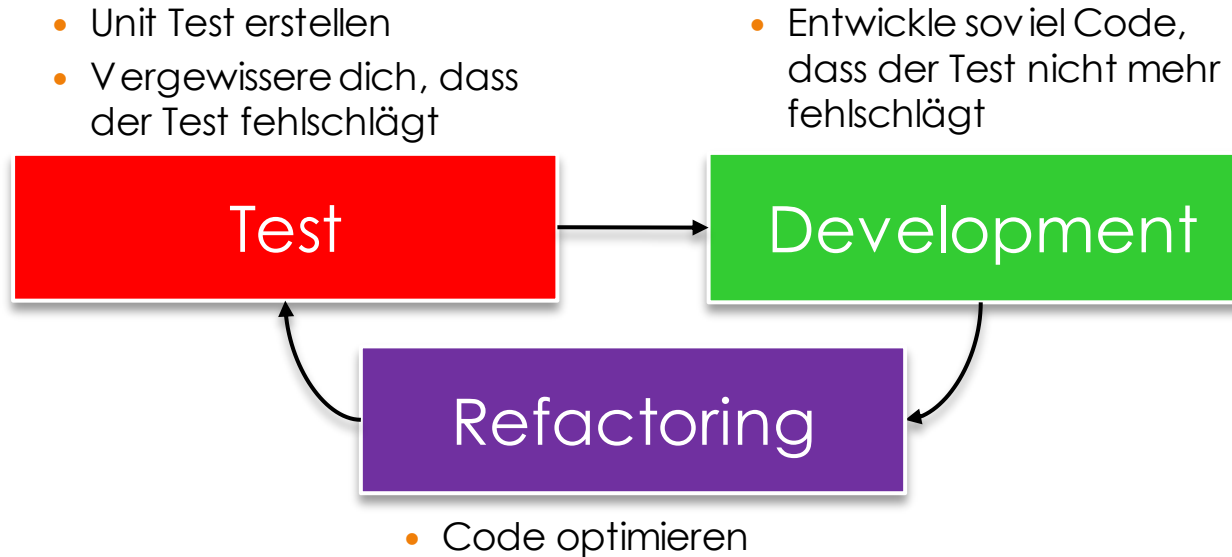
eXtreme Programming



**Was haben agile Projekte jetzt mit Unit Tests zu tun?**



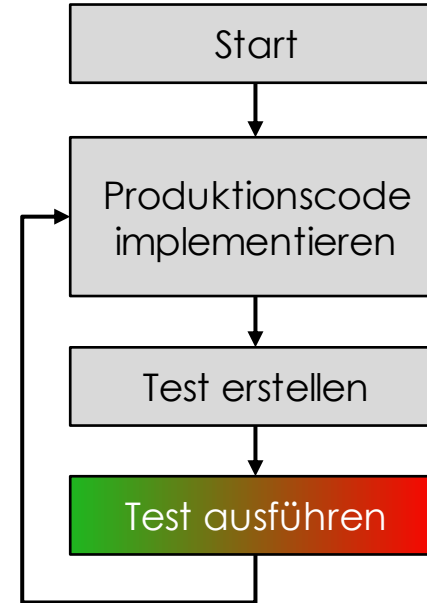




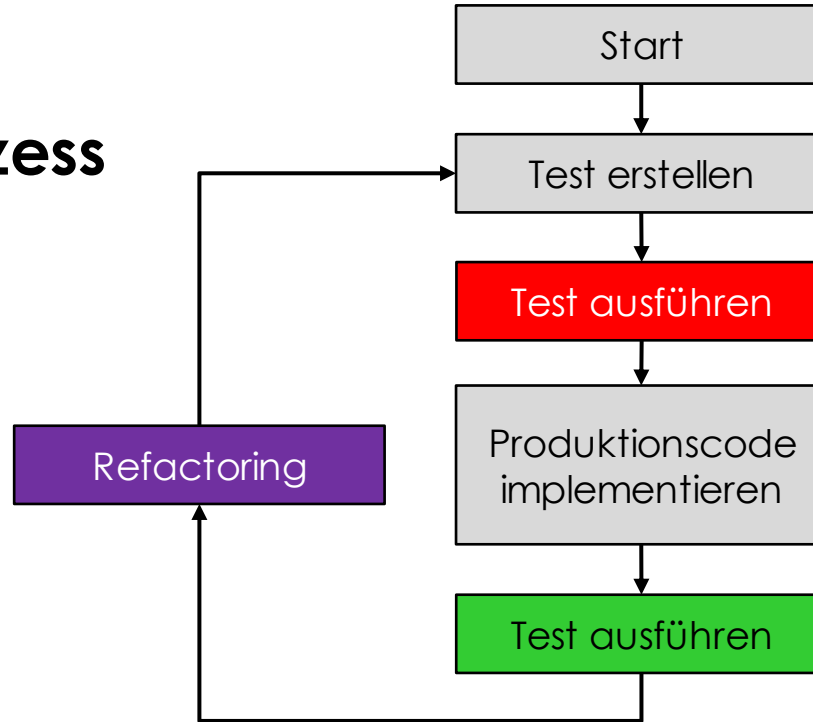
- **TDD – Test-driven development**

- Bei TDD geht es ganz einfach darum, dass automatisierte Tests vor der Code Implementierung erstellt werden.
- TDD ist ein wesentlicher Teil von Extreme Programming (XP)
- TDD setzt auf kurze Iterationszyklen zwischen Tests und Implementierung

- **Traditioneller Entwicklungsprozess**



- **Testgetriebener Entwicklungsprozess**



- **Refactoring**

- Optimieren des Codings ohne die Funktionalität zu ändern
- Coding lesbarer gestalten, Code-Dupletten bereinigen, Methode „verkleinern“, ...
- Robert C. Martin sagt: Wir „massieren“ unseren Code
- Clean Code

# Live Demo

Jetzt erleben wir TDD

- **Erhöht sich der Aufwand durch TDD/Unit Tests?**
  - Stimmt nicht
  - TDD bzw. Unit Tests verändern nur unsere Tests, die wir – hoffentlich – auch bisher schon alle machen

- **Unit Tests „verrotten“ mit der Zeit**
  - Stimmt nicht!
  - Unit Tests sind Bestandteil der Entwicklungen, es liegt ans diese aktuell zu halten



- **Unit Tests sind in ABAP nicht sinnvoll einsetzbar**
  - Stimmt nicht!
  - Wenn wir nach modernen Standards entwickeln (Clean Code, OO, Entwurfsmuster, .. ), können Unit Tests wunderbar im ABAP Umfeld eingesetzt werden
  - DB Zugriffe, WebServices Calls, ... können beispielsweise über Interfaces abgebildet werden und damit durch Mockups ersetzt werden (trotzdem, DB Interaktionen sind natürlich ein Problem)

- **Unit Tests sind in ABAP nicht sinnvoll einsetzbar**
  - Stimmt nicht!
  - Wenn wir nach modernen Standards entwickeln (Clean Code, OO, Entwurfsmuster, .. ), können Unit Tests wunderbar im ABAP Umfeld eingesetzt werden

- **Bei User Interface, DB Zugriffen unbrauchbar**
  - Stimmt nicht!
  - DB Zugriffe, WebServices Calls, ... können beispielsweise über Interfaces abgebildet werden und damit durch Mockups ersetzt werden
    - trotzdem, DB Interaktionen sind natürlich ein Problem aber ....
  - ABAP 7.50 – TEST-SEAM & TEST-INJECTION

- **Kunden sind skeptisch (höhere Aufwände, ... )**
  - Unsere Kunden wollen sauberen, funktionierenden und wartbaren Code. - Unit Tests gewährleisten dies
  - Es liegt an uns, dem Fachbereich bzw. den Kunden die Vorteile aufzuzeigen und durch bessere Softwarequalität auch zu beweisen

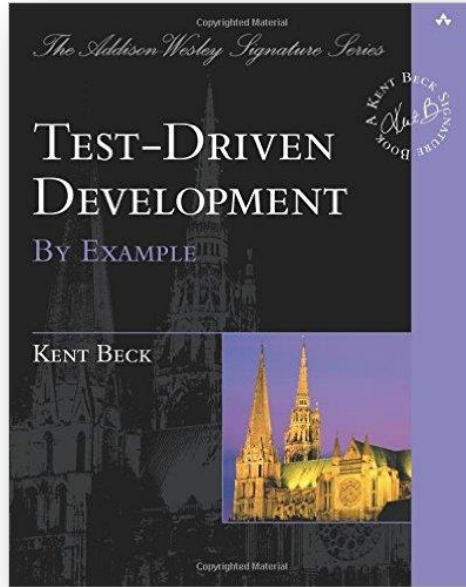
# TDD üben mit Waza, Randori

- **Paarweise üben**

- Entwickler A schreibt einen ABAP Unit Test
- Entwickler B implementiert die Methode, bis der Test bestanden wird
- Entwickler A kann für den Test z.B. auch min. Geschwindigkeit oder den max. Speicherverbrauch vorgeben.

- **Übungen für's Team**

- Editor wird an die Wand projiziert
- Entwickler A schreibt einen ABAP Unit Test
- Entwickler B implementiert die Methode, bis der Test bestanden wird
- Entwickler B schreibt einen ABAP Unit Test
- Entwickler C implementiert die Methode, bis der Test bestanden wird
- ...



ISBN 978-0321146533



ISBN 978-3826697128



- **Weitere Informationsquellen**

- <https://sites.google.com/site/unclebobconsultingllc>
- [https://de.wikipedia.org/wiki/Extreme\\_Programming](https://de.wikipedia.org/wiki/Extreme_Programming)
- <https://www.youtube.com/watch?v=QHnLmvDxGTY>
- <http://clean-code-developer.de/>



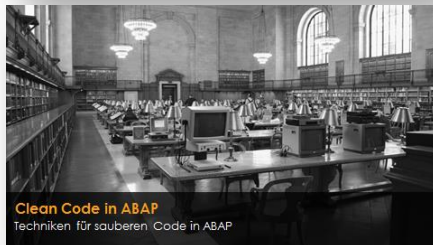
## ABAP 7.50 Releaseabhängige Änderungen

Webinar, 25.11.2016

<https://www.eventbrite.com/e/abap-750-releaseabhangige-anderungen-tickets-28011248398>

- **Intensiv Workshop**
  - **Clean Code**
  - **ABAP Units/TDD**
  - Ca. 3 - 4 Stunden
  - Wien
  - An einem Freitag
  - Dezember 2016 oder Jänner 2017





<http://www.cadaxo.com/blog/>

# See you again!

***Thank you for participating!***



<https://twitter.com/foessleitnerj>



<https://www.linkedin.com/in/johann-fößleitner-a9851b2a>



[https://www.xing.com/profile/johann\\_foessleitner](https://www.xing.com/profile/johann_foessleitner)



[johann.foessleitner@cadaxo.com](mailto:johann.foessleitner@cadaxo.com)



<https://twitter.com/domibiglsap>



<https://www.linkedin.com/in/dominik-bigl-9b98b68b>



[https://www.xing.com/profile/dominik\\_bigl](https://www.xing.com/profile/dominik_bigl)



[dominik.bigl@cadaxo.com](mailto:dominik.bigl@cadaxo.com)

**If you want to stay in touch ...**



**cadaxo**

**Cadaxo GmbH**  
Stubenring 18/5a | 1010 Vienna, Austria

[office@cadaxo.com](mailto:office@cadaxo.com)  
[www.cadaxo.com](http://www.cadaxo.com)



<https://www.linkedin.com/company/cadaxo-gmbh>



<https://www.xing.com/companies/cadaxogmbh>



<https://www.facebook.com/CadaxoGmbH>



<https://twitter.com/cadaxo>



<http://com.slideshare.net/cadaxogmbh>



<http://www.youtube.com/CadaxoGmbH>