



Fachhochschule Brandenburg  
Brandenburg University of Applied Sciences

# Qualitätsmanagement mit Continuous Integration

Untersuchung anhand einer Machbarkeitsstudie in der Praxis

Abschlußpräsentation zur Studienarbeit

**Lars Gohlke**

Diplom-Informatiker (FH)

- **Was ist *Continuous Integration*?**
- ***Continuous Integration* – wofür?**
- **Einordnung: Agile Softwareentwicklung**
- **Ziele meiner Arbeit**
- **Umsetzung**
- **Fazit**

- dt.: Kontinuierliche Integration

- Idee:

Das Problem der Integration *fortlaufend* in den  
*Entwicklungsprozess* einzubinden

## ■ Komplexität von Fehlern und deren Auswirkungen reduzieren

Unmodified  Added  Removed

```
trunk/services/BaseService/src/de/idema/session/ServiceBean.java
r4068 r4071
191 191         }
192 192         else{
193         e = (Exception) e_.getCause();
194         Throwable cause = e.getCause();
195         if (cause != null){
196             e = (Exception) cause;
197         }
198         else{
199             e.printStackTrace();
200         }
201     }
...
437 443     * @param message
438 444     */
445     @Deprecated
439 446     protected void setResponseFailure(final Map<String, Object> response, final String errorCode, final String message){
440 447
...
471 478     * @param e
472 479     */
473     protected void setResponseFailure(final Map<String, Object> response, final String errorCode, final Exception e){
480     protected void setResponseFailure(final Map<String, Object> response, final CODES errorCode, final Exception e){
474 481         setResponseFailure(response, errorCode, e.getMessage());
475 482         response.put(CONSTANTS.KEYS.EXCEPTIONS.toString(), e);
...
482 489     * @param errorCode
483 490     */
484     protected void setResponseFailure(final Map<String, Object> response, final String errorCode){
491     protected void setResponseFailure(final Map<String, Object> response, final CODES errorCode){
485 492         setResponseFailure(response, errorCode, (String) null);
486 493     }
```

## ■ Qualität erhöhen

RC: Suspicious comparison of Long references in  
`de.idema.entity.Credentials.Credential.isSame(Credential)`

This method compares two reference values using the `==` or `!=` operator, where the correct way to compare instances of this type is generally with the `equals()` method. It is possible to create distinct instances that are equal but do not compare as `==` since they are different objects. Examples of classes which should generally not be compared by reference are `java.lang.Integer`, `java.lang.Float`, etc.

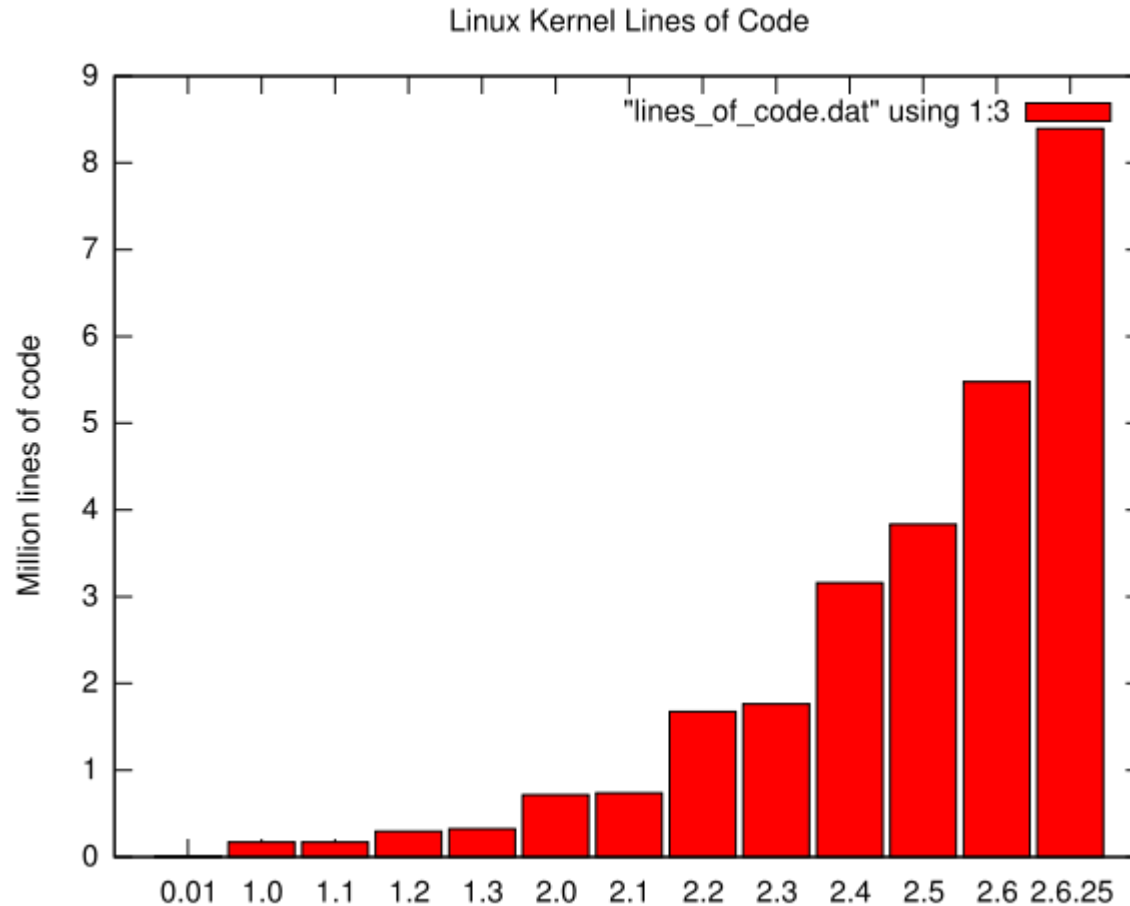
---

```
174  public boolean isSame(Credential credential)
175  {
176      return credential.getId() == id && equals(credential);
177  }
```

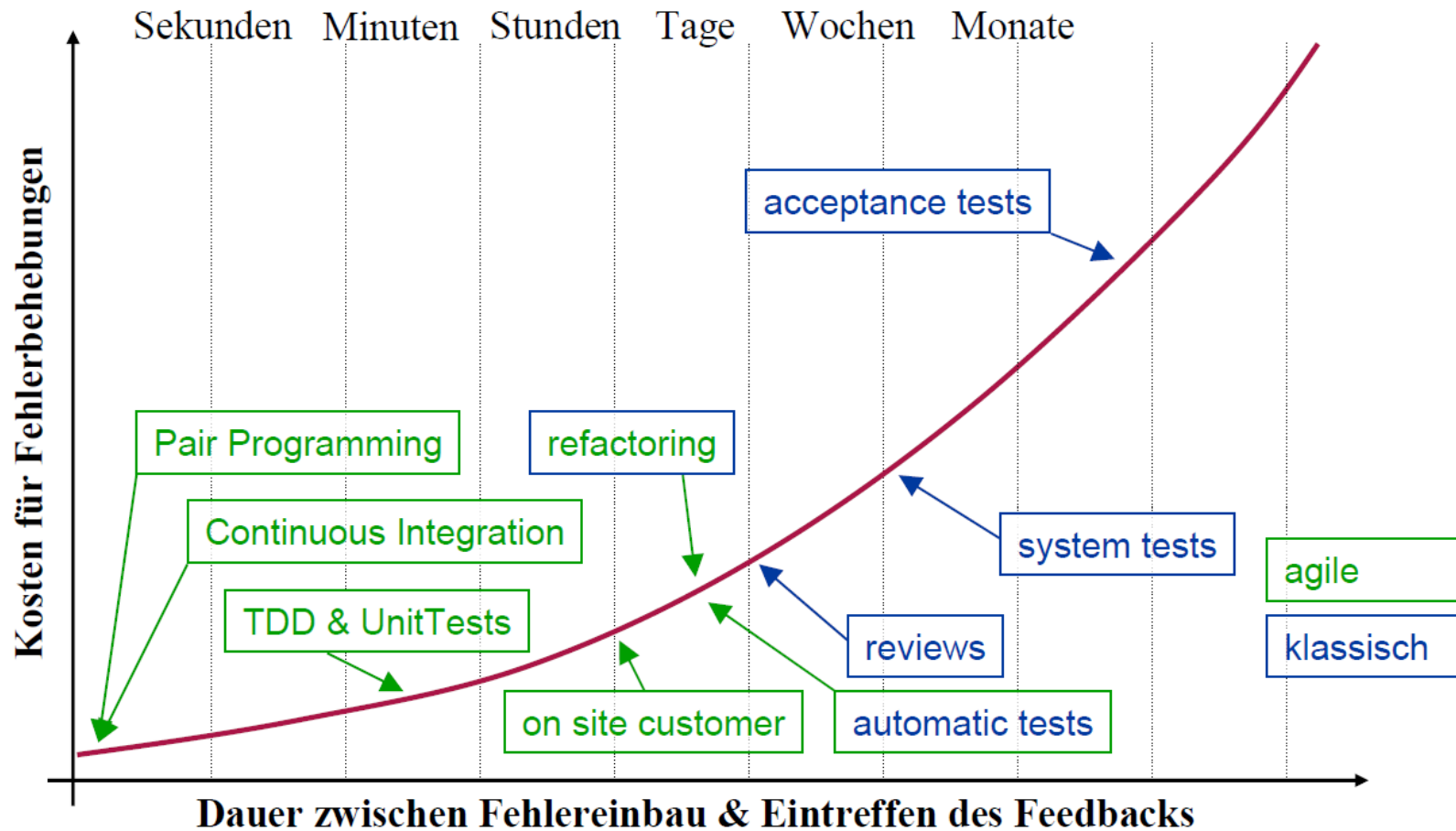
## ■ Qualität erhöhen

S	W	Job ↓	Last Success	Last Failure
		<a href="#">test-all</a>	12 hr ( <a href="#">#52</a> )	N/A
		<a href="#">AccountService</a>	12 hr ( <a href="#">#206</a> )	26 days ( <a href="#">#174</a> )
		<a href="#">AccountServiceEJB</a>	12 hr ( <a href="#">#183</a> )	10 days ( <a href="#">#177</a> )
		<a href="#">AccountServiceTest</a>	11 hr ( <a href="#">#277</a> )	27 days ( <a href="#">#238</a> )
		<a href="#">AuthenticationService</a>	12 hr ( <a href="#">#213</a> )	26 days ( <a href="#">#178</a> )
		<a href="#">AuthenticationServiceEJB</a>	12 hr ( <a href="#">#184</a> )	10 days ( <a href="#">#178</a> )
		<a href="#">AuthenticationServiceTest</a>	12 hr ( <a href="#">#257</a> )	16 hr ( <a href="#">#256</a> )
		<a href="#">BackendWrapper</a>	3 mo 8 days ( <a href="#">#9</a> )	10 days ( <a href="#">#54</a> )
		<a href="#">BackendWrapperClient</a>	3 mo 8 days ( <a href="#">#9</a> )	16 hr ( <a href="#">#68</a> )

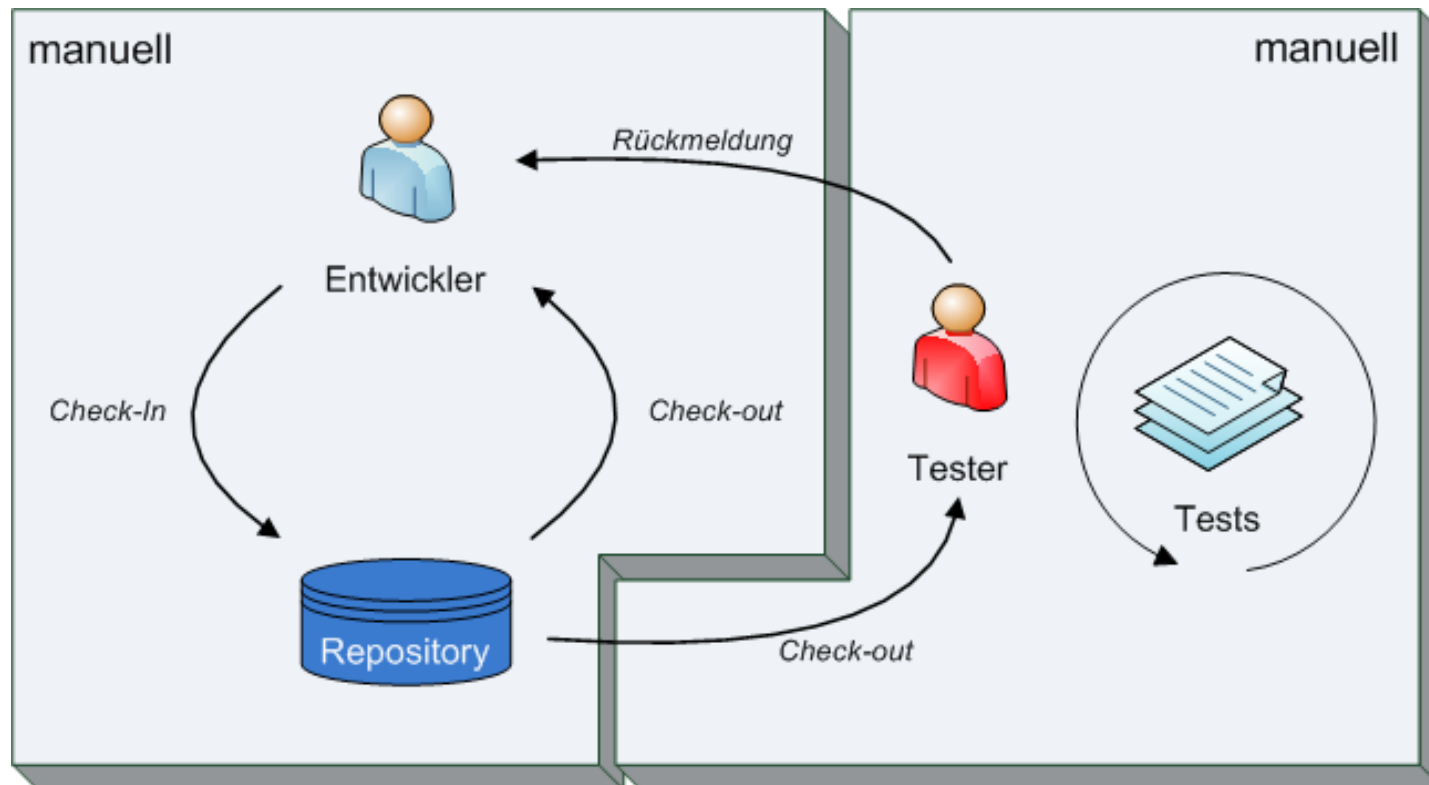
- Entwicklungsgeschwindigkeit erhöhen



## ■ Kosten senken







**Jede Änderung am System führt zu einem vollständigen Systemtest - JEDE!**

- Eine der Säulen im **agilen** Softwareentwicklungsprozess neben:
  - Automatisiertes Testen
  - User stories
  - ...

Prioritäten:

- funktionsfähige Software vor umfassender Dokumentation,
- Zusammenarbeit mit dem Kunden vor Vertragsverhandlungen,
- Flexibilität hinsichtlich veränderter Anforderungen vor Planerfüllung

## ■ Untersuchung der Machbarkeit

### *Wissenschaft trifft auf Realität*

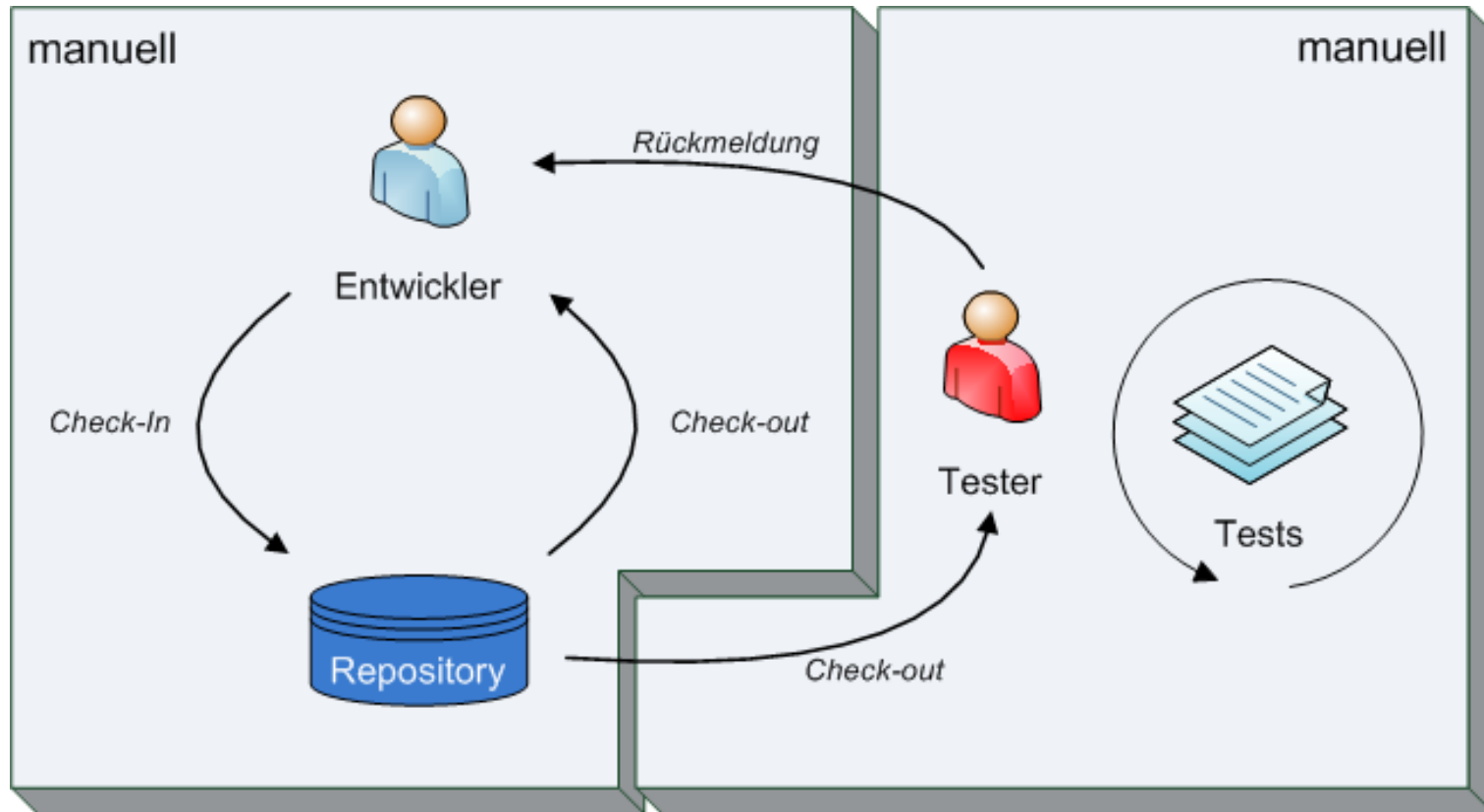
- Kleines Unternehmen – 5 Leute
- Ambitionierter Zeitplan mit konkreten Zielen

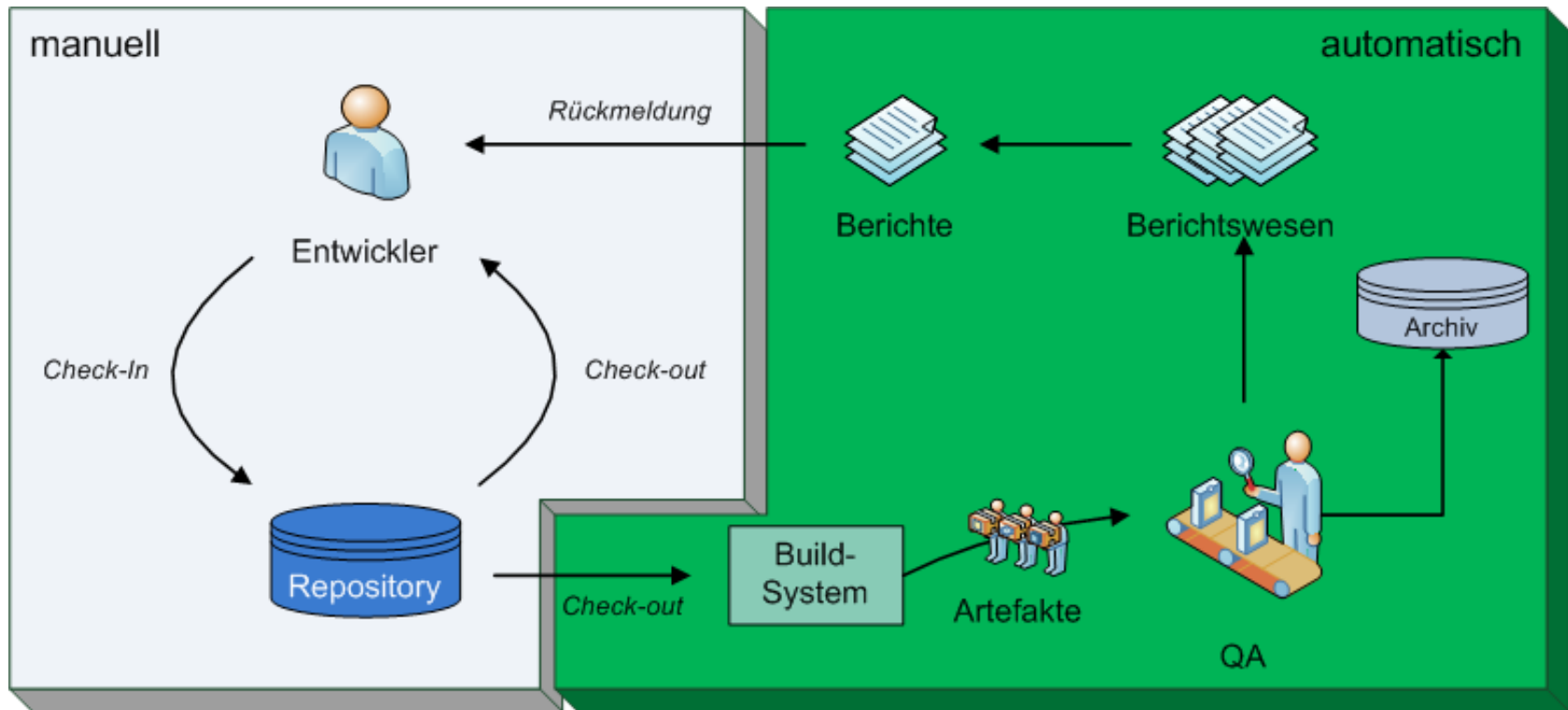
## ■ Rahmenbedingungen

- Wenig Zeit
- Wenig Leute
- Wenig Wissen
- Wenig Erfahrung
- ...

## ■ Rahmenbedingungen

- Wenig Zeit <- Zeit gewonnen
- Wenig Leute <- eine Person
- Wenig Wissen <- wächst ...
- Wenig Erfahrung <- ... exponentiell ;)
- ...







- Erfassen des IST-Zustandes,
- Formulieren des SOLL-Zustandes,
- Umsetzung des Konzeptes.

Arbeitsziele	IST	SOLL
Zentrale Quellcodeversionierung	✓	✓
Commit in die Hauptentwicklungslinie	✓	✓
Jeder soll Veränderungen wahrnehmen	✓	✓
"Keep the build fast!"	✳	✓
Test in einer geklonten Produktivumgebung	✳	✓
Automatisierter Build	✗	✓
Selbsttestender Build	✗	✓
Kompilierung jeder Version auf einem Integrationsrechner	✗	✓
Einfacher Zugang zur neuesten ausführbaren Version	✗	✓
Automatische Auslieferung	✗	✓

Legende : ✓ = ja/vollständig, ✗ = nein, ✳ = teilweise/unvollständig

### *Ausblick auf die Arbeit ...*

- Hudson
- Ant
- pmd/findbugs/checkstyle ...

- Versprechen von CI erfüllt sich (!)
- sehr gute Werkzeugunterstützung
- Hohe Lernkurve bei der Umsetzung
- -> Zeit einplanen

- Versprechen von CI erfüllt sich (!)
- sehr gute Werkzeugunterstützung
- Hohe Lernkurve bei der Umsetzung
- -> Zeit einplanen

*Es ist vorbei!*

**Danke für Ihre Aufmerksamkeit!**

