

# FrOSCon: Mockito für Einsteiger

20.08.16, P.Möbius

© Copyright 2016 anderScore GmbH

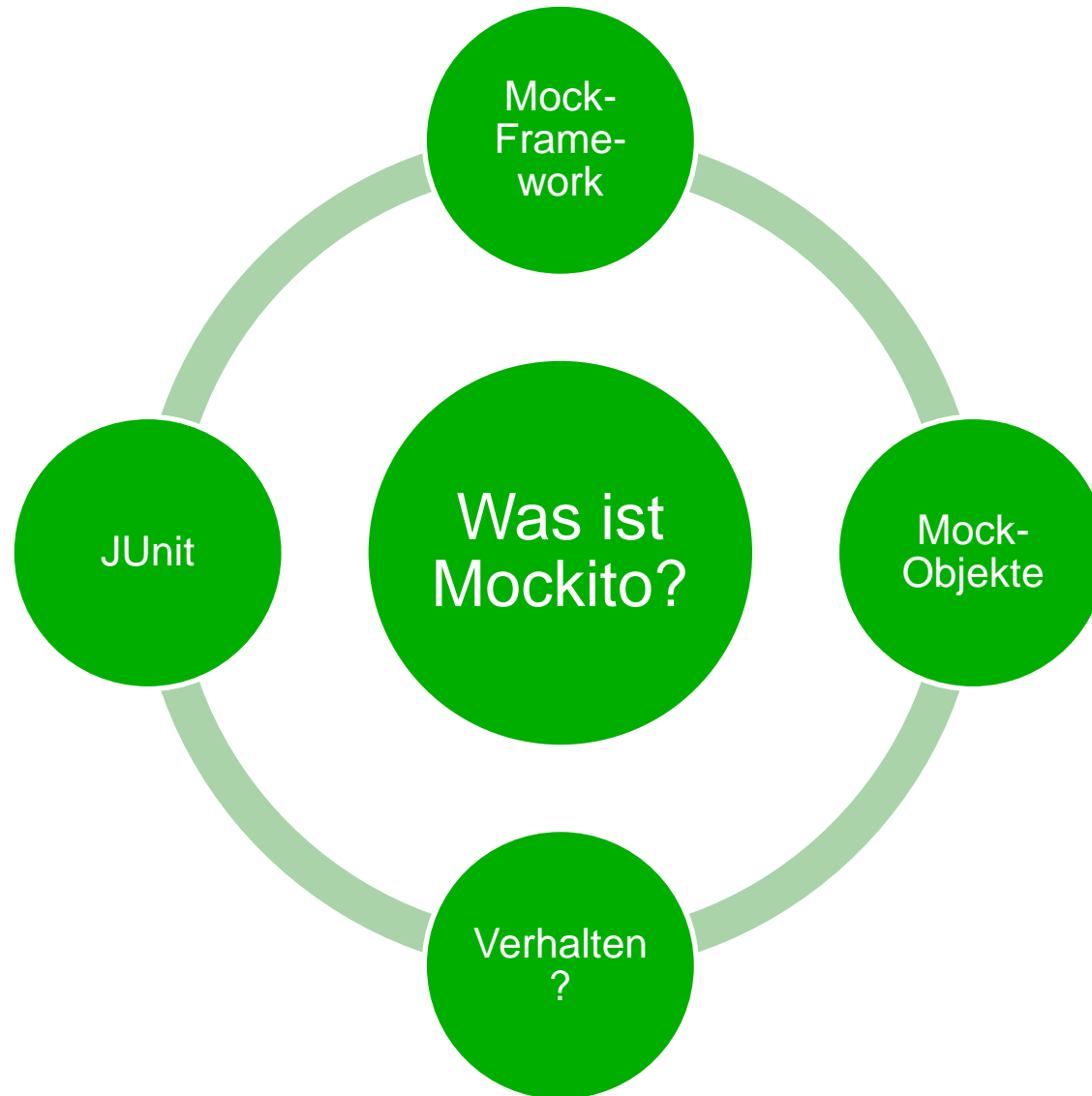


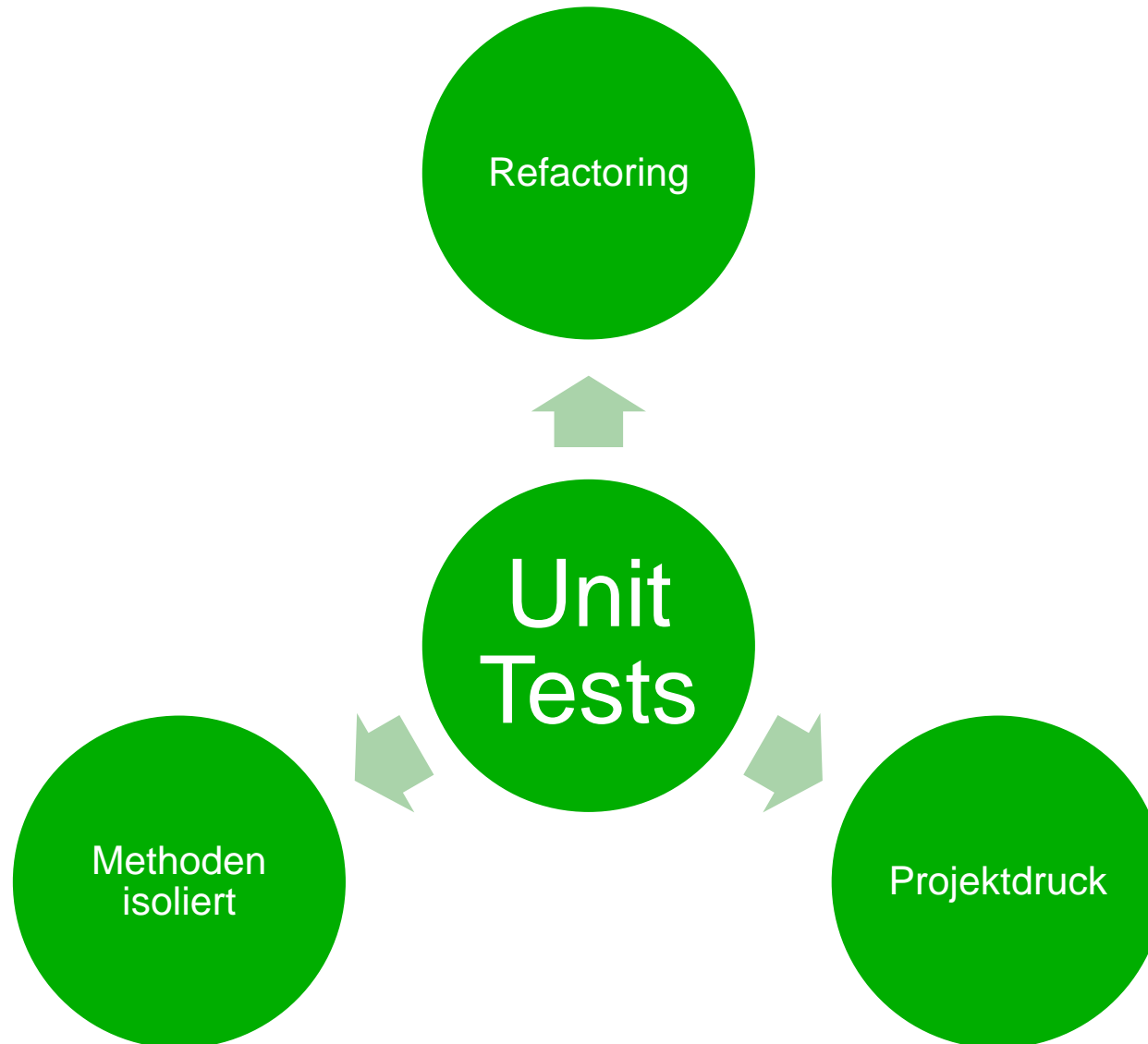
1. Was ist Mockito?	4
2. Warum Unit Tests?	5
3. Hilfsklassen & Mocking	6
4. Vorgehensweise & Don'ts	9
5. Features	11
6. Live Demo	12
7. Zusammenfassung	13

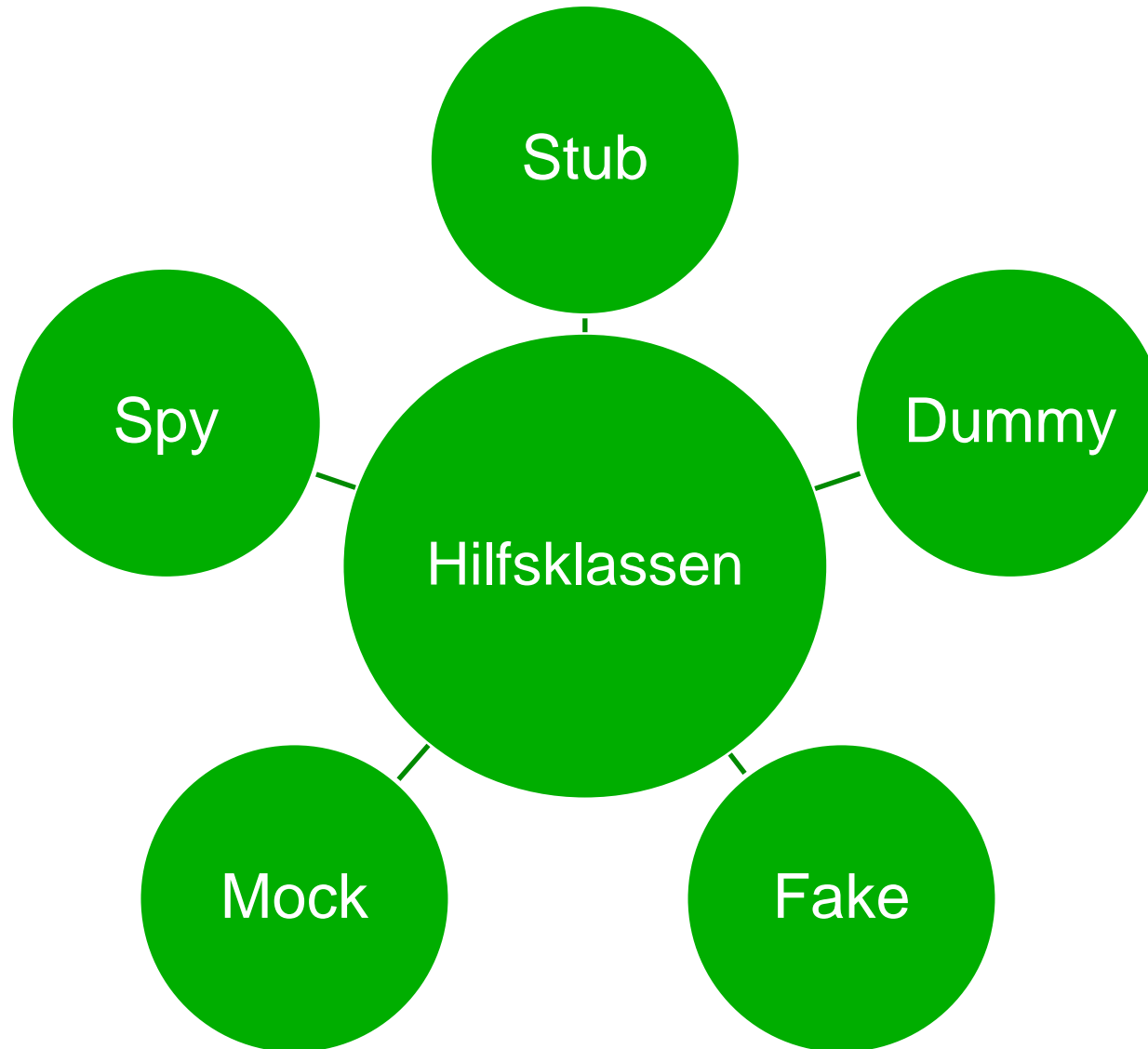
## Patrick Möbius (M.Sc. C.S.)

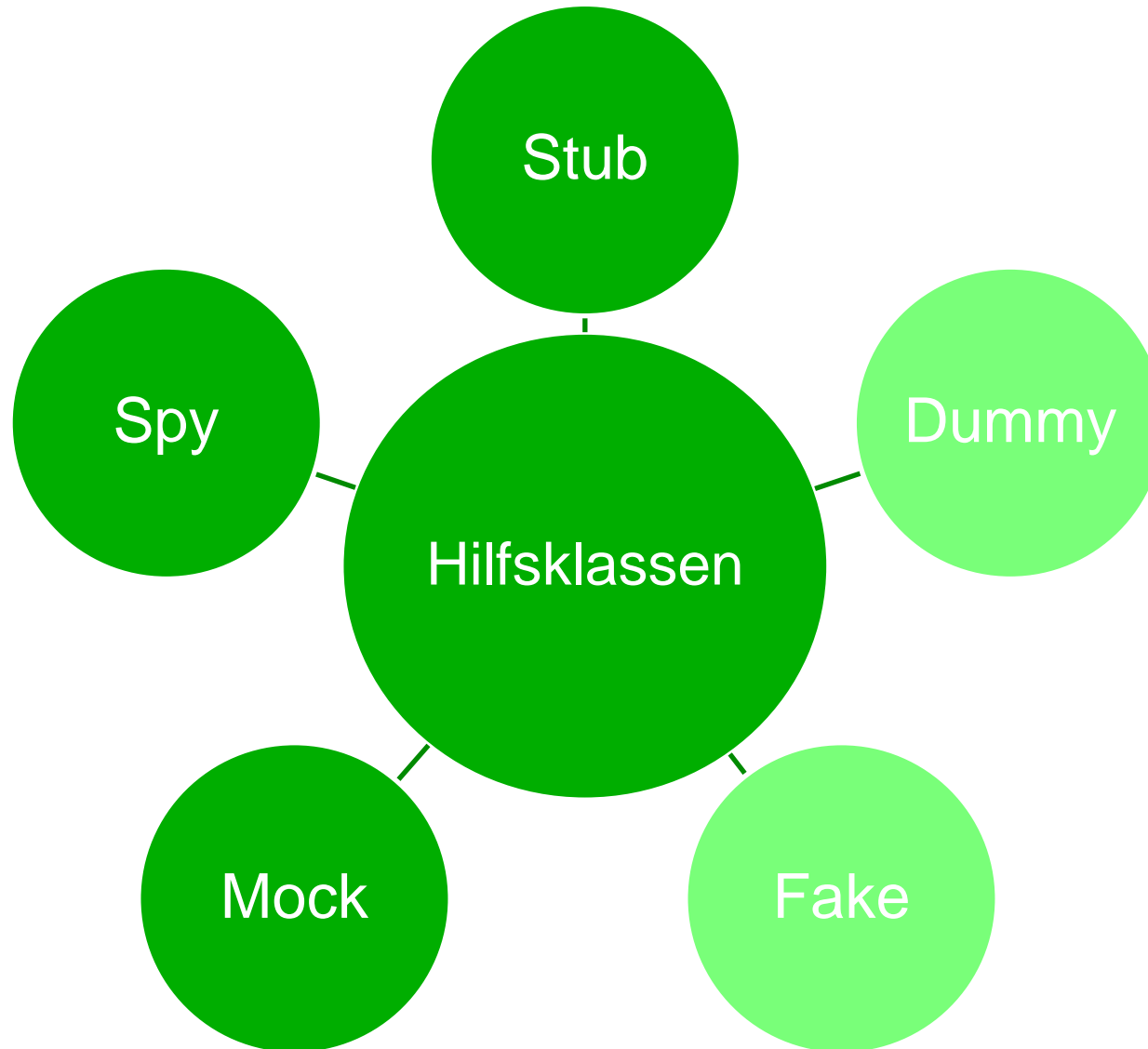
- Software Engineer
- Seit 2011 bei anderScore
- Schwerpunkte
  - Android mobile-Apps
  - Frontends für dynamische Webapplikationen
  - WS-Anbindung für Persistence Layer, Einsatz von speziellen Caching-Arten
  - BPM-Umgebungen; Prozessdarstellungen
- Java, Wicket, GWT/ GXT, JavaScript, ...











Warum  
Mocking?

Klassen  
sind nicht  
isoliert

Problem:  
Externe  
Services

Fokus:  
Unit  
Tests?

Mockito  
to the  
rescue!



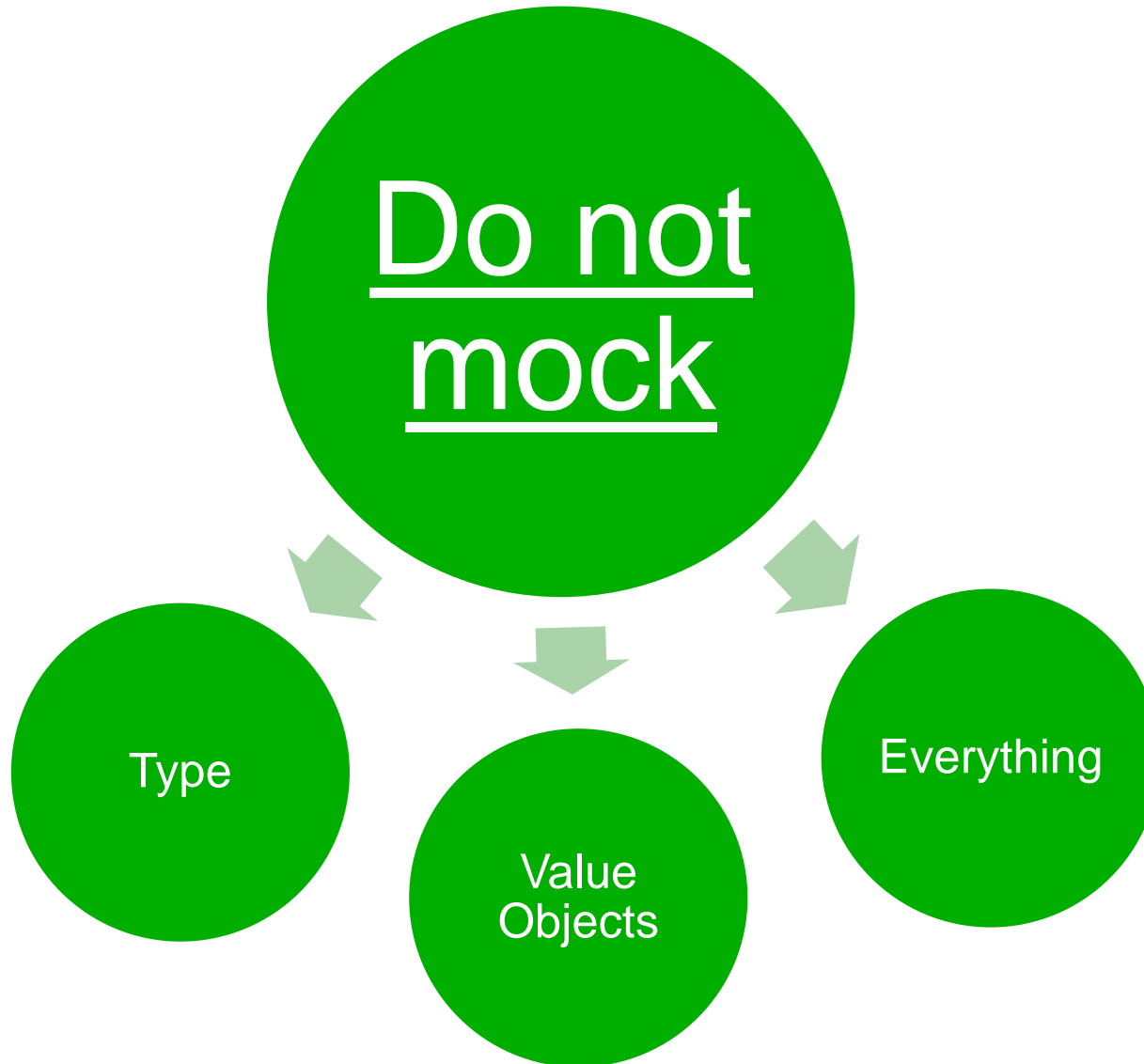
“when-method-call-then-  
return”-syntax

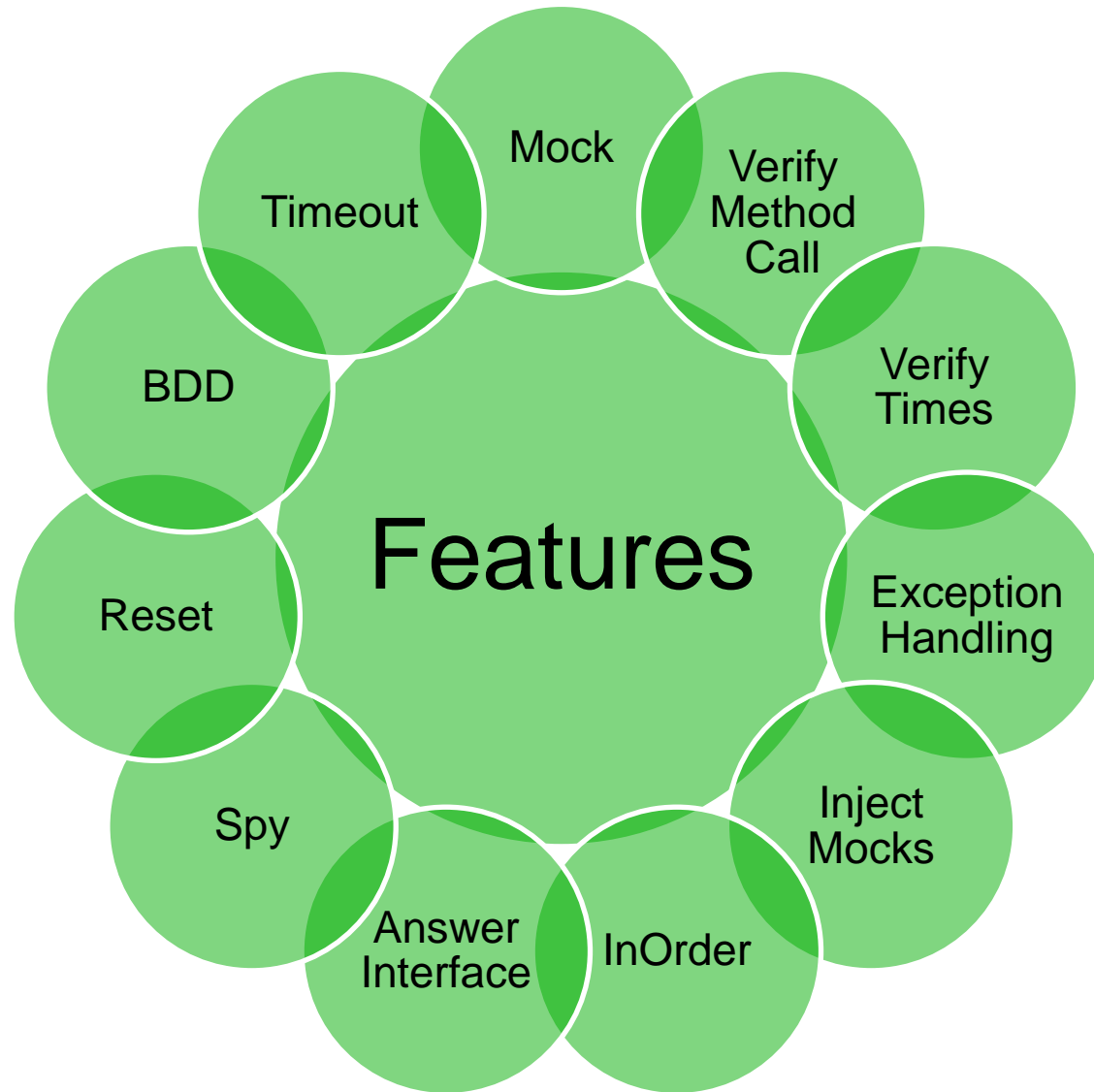


AssertEquals(expected,actual)



Verify().itHappend





## Zeit für eine Live Demo!



- **mock()/ @Mock** erzeuge Mock
  - Optional Verhalten spezifizieren via **Answer/ ReturnValues(/ MockSettings**
  - **when()(/ given())** Verhalten eines Mocks definieren
  - **Answer** Interface für eigene Antworten
- **spy()/ @Spy** partielles Mocking
  - Reale Methoden + verify
  - Stubbed Methoden + verify
- **@InjectMocks** auto. Inject von **@Spy** & **@Mock** Feldern
- **verify()** Methodenaufruf mit vorgegebenen Argumenten

# Feedback? Besten Dank!

## Lust auf mehr?

Wir bieten:

- Anstellungen
- Praktika
- Studentenjobs
- Abschlussarbeiten

✉ info@anderScore.com

☎ 0221-3558-3530

Folien, Beispielcode:

<https://github.com/anderscore-gmbh>

