

Acceptance Test-Driven Development

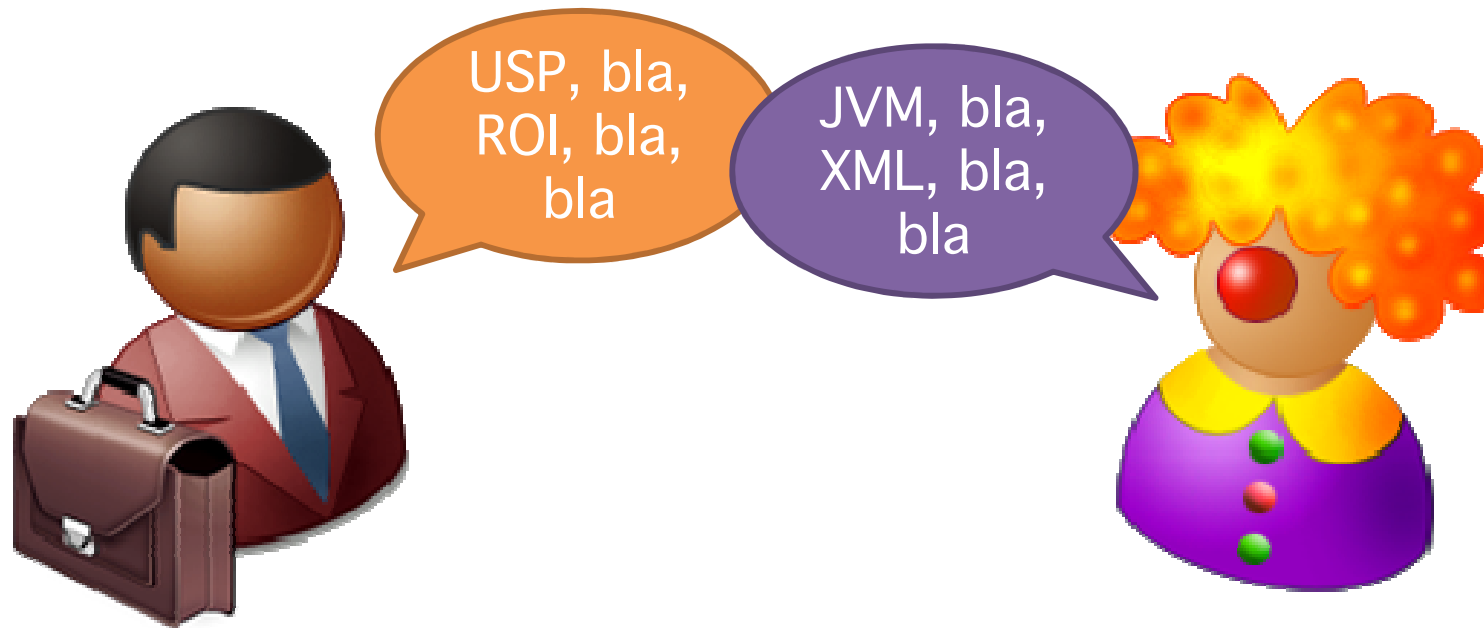
- Ausführbare Tests in natürlicher Sprache schreiben -

Mirko Seifert, Christian Wende, DevBoost GmbH

25. September 2013, Berlin Days of Software Engineering, Berlin

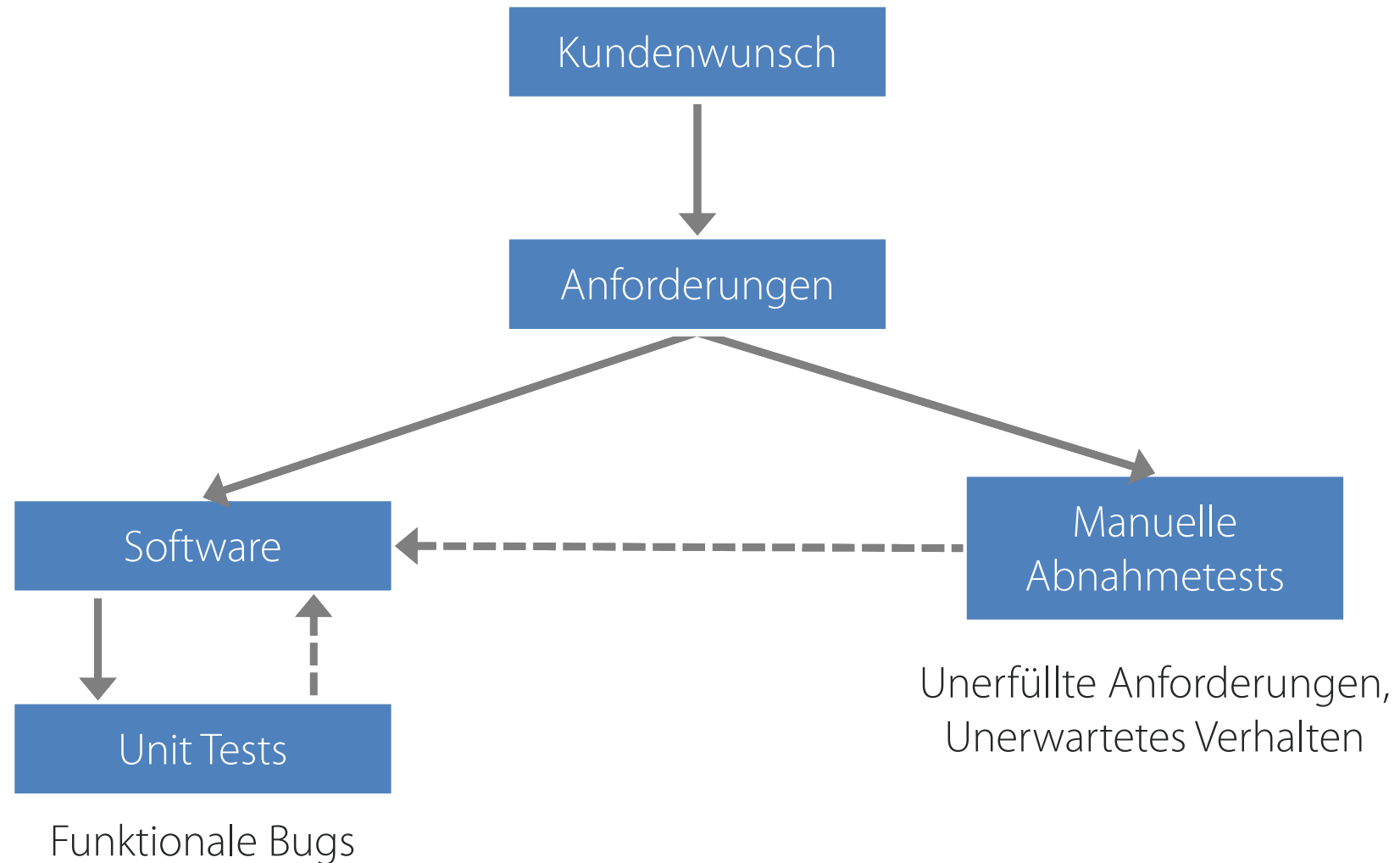
Ausgangssituation: Softwareentwicklung

- Softwareentwickler bauen Software, um die Anforderungen ihrer Kunden zu erfüllen (Job to be done)



- Kommunikationsproblem in der Anforderungsanalyse
- Anforderung als wichtiges Akzeptanzkriterium bei der Auslieferung

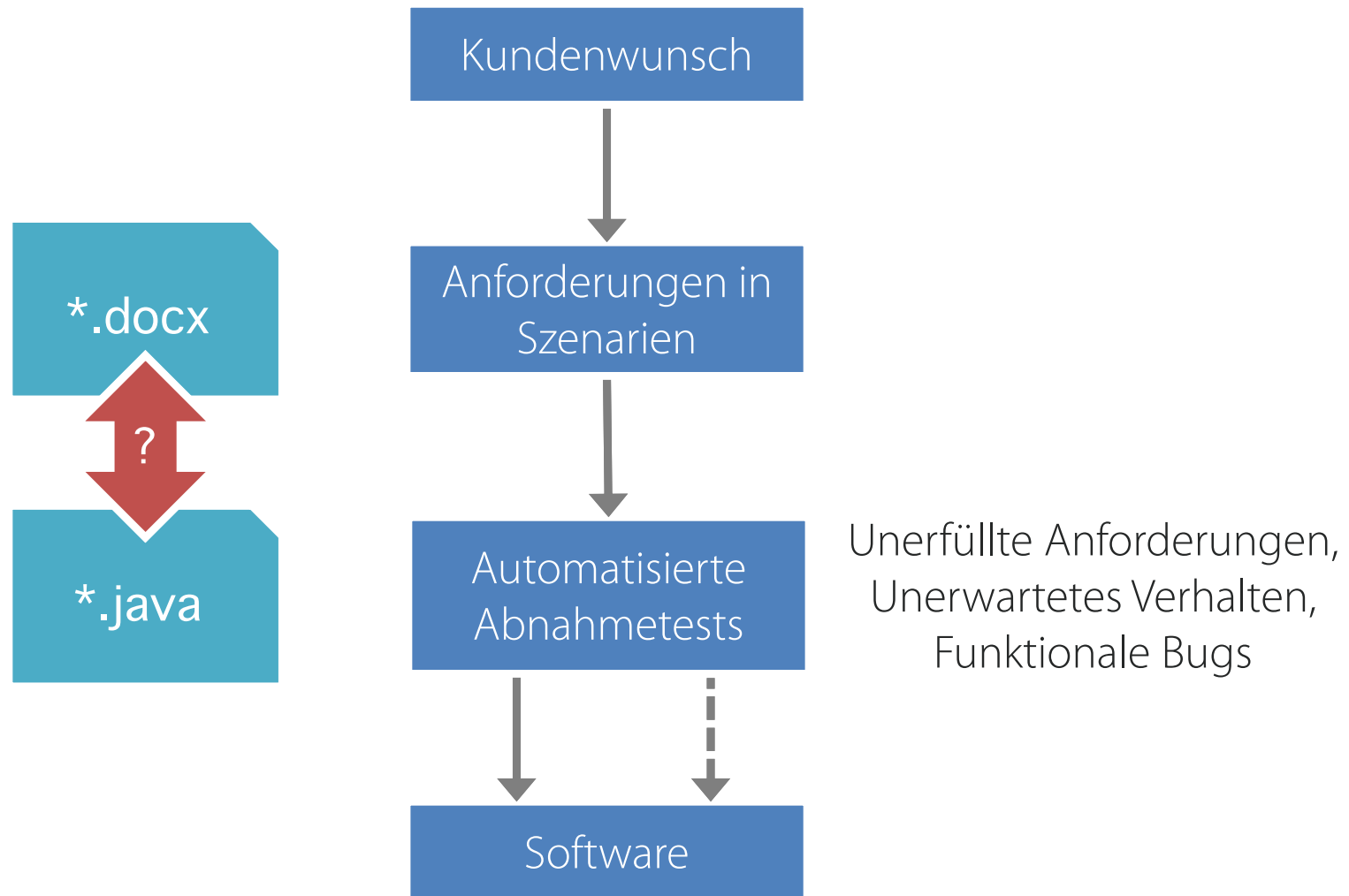
Klassisches Vorgehen



Probleme

- Verständnisprobleme werden zu spät erkannt
- Anforderungsspezifikation ist "totes" Artefakt
- Communication Gap zwischen
 - Anforderungsspezifikation und Softwareimplementierung
 - Anforderungsspezifikation und Abnahmetest
- Diskrepanz zwischen Implementierung und Akzeptanztest
- Unit Tests validieren aus Entwicklersicht, nicht aus Kundensicht

Acceptance Test Driven Development (ATDD)



Anforderungen gemeinsam spezifizieren

Beispiel Flugbuchungssystem (Airplanes, Flights, Passengers)

Story: Booking a Flight

Given a Passenger John Doe	
Given an Airplane Boing-787	
With 200 total seats	Eingabe
Given a flight LH-1234	
With 200 free seats	

Book seat for John Doe at LH-1234	Systemfunktion
-----------------------------------	----------------

Assume Success	Ergebniserwartung
----------------	-------------------

Tools, Tools, Tools


Cucumber

Jnario

 NatSpec

Vergleich

 *Cucumber*

Jnario

 NatSpec

Mapping von Text auf Step Definitions	Annotation von Sätzen mit Code	Mapping von Sätzen auf Test Support Methoden
Kommandozeilenorientiert	Eclipse Integration	Eclipse Integration
Interpretiert	Generiert Test Fälle	Generiert Code
Gherkin	Gherkin, Specs (Xtend)	Frei
ATDD	ATDD	ATDD + Lean Modeling
Community	Individual	Professional

Vorteile von ATDD

- **Verbesserte Kommunikation und Zusammenarbeit** im Team und mit Kunden
- **Fokussierung** auf die wesentlichen und kundenrelevanten Softwarefunktionen
- **Direkte Messung** des aktuellen Projektfortschrittes
- **Höhere Entwicklungseffizienz** durch klare, kontinuierliche Verständigung, weniger Missverständnisse, weniger Iterationen und weniger "tote" Spezifikationen
- **Höhere Produktqualität** durch frühestmögliche, kontinuierliche Überprüfung des Implementierungszustandes bzgl. der Akzeptanzkriterien

Die nächsten Schritte

Tools ausprobieren

- Cucumber: <http://cukes.info/>
- Jnario: <http://jnario.org/>
- NatSpec: www.nat-spec.com - Kostenlose 30-Tage Trial Version

Literatur

- Specification by Example: How Successful Teams Deliver the Right Software (Gojko Adzic - Manning)
- The Cucumber Book: Behaviour-Driven Development for Testers and Developers (Matt Wynne, Aslak Hellesoy - Pragmatic Programmers)
- ATDD in der Praxis: Eine praktische Einführung in die akzeptanztestgetriebene Softwareentwicklung (Markus Gärtner - dpunkt.verlag)
- ATDD by Example (Markus Gärtner - Pearson Education)

Danke! Fragen?

<http://www.devboost.de>

mirko.seifert@devboost.de – christian.wende@devboost.de