

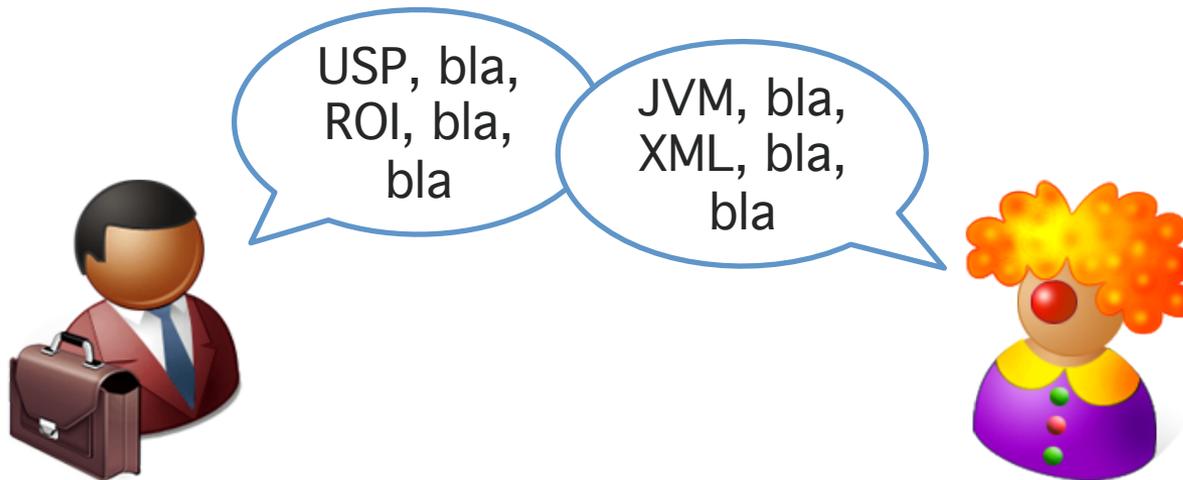
Lean Modeling - Software Systeme einfach und präzise mit natürlicher Sprache spezifizieren

Dr. Christian Wende und Dr. Tobias Nestler, DevBoost GmbH

21. Mai 2014, Karlsruher Entwicklertag 2014, Dresden

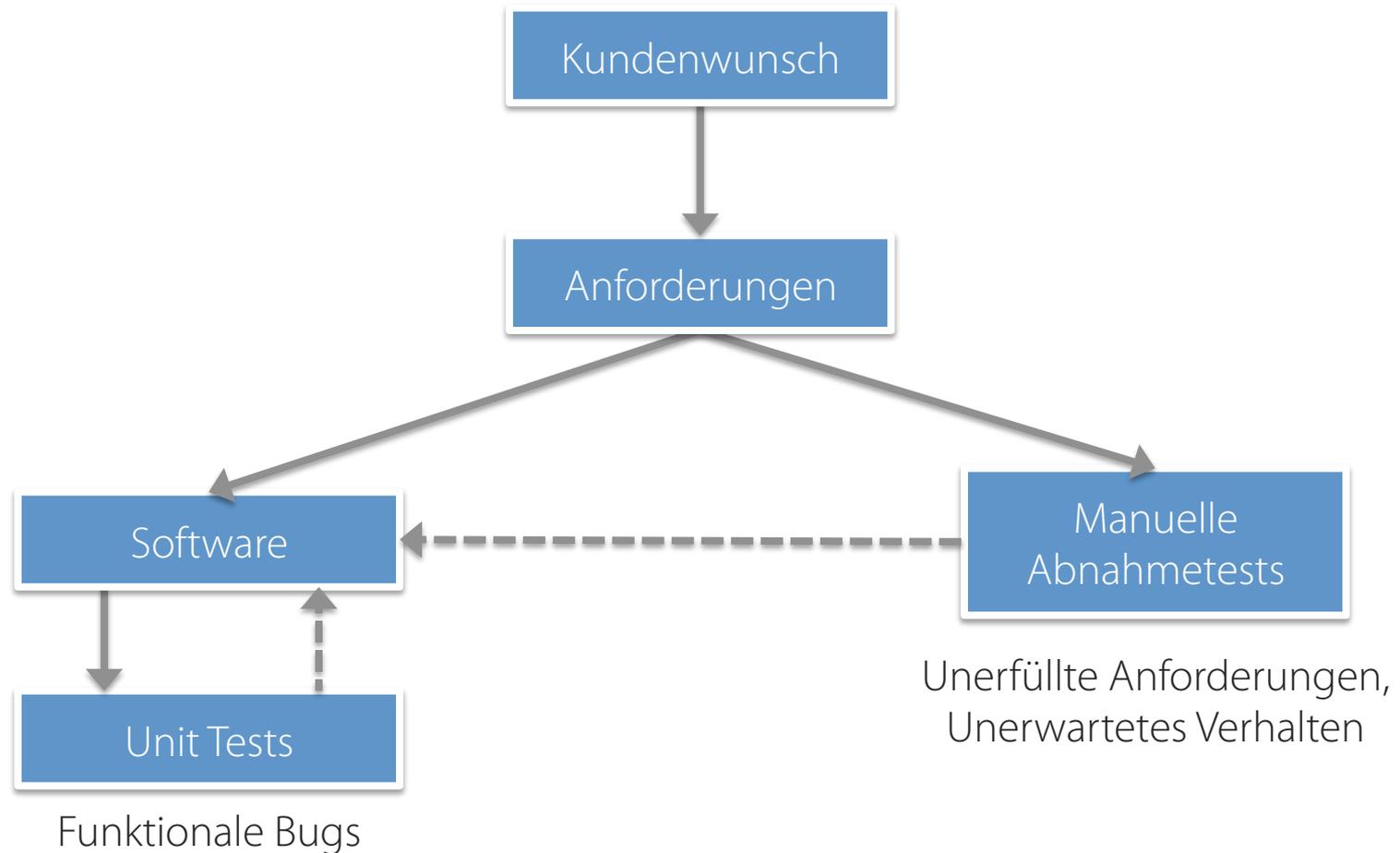
Ausgangssituation: Softwareentwicklung

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



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

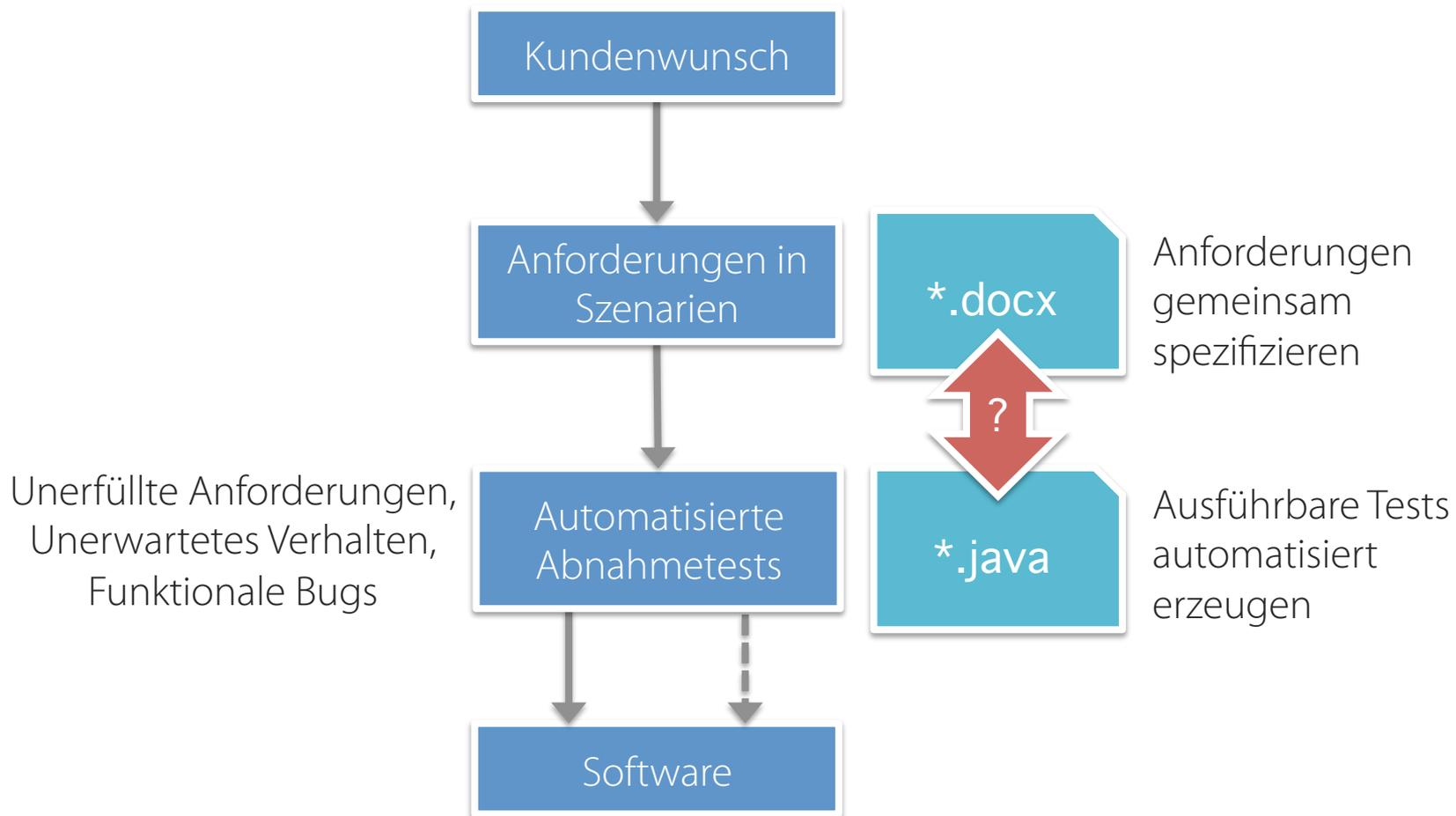
Klassisches Vorgehen



Probleme

- Verständnisprobleme werden zu spät erkannt
- Anforderungsspezifikation ist "totes" Artefakt
- Communication Gap zwischen
 - Anforderung und Test
 - Anforderung und Implementierung
- Automatisierte Unit Tests validieren aus Sicht des Entwicklers, nicht aus Sicht des Kunden
- Manuelle Abnahmetests sind aufwendig und fehleranfällig

Acceptance Test Driven Development (ATDD)



Anforderungen gemeinsam spezifizieren

Beispiel Flugbuchungssystem

- Klassische Anforderung [Lastenheft nach Balzert]
/FB 1030/ A Passenger receives a ticket after booking a flight
- Agile User Story [nach Jeffries]

As a passenger
I want to book a flight and
receive a valid ticket.

Confirmations:

- Ticket is only issued if
if enough seats
available

Anforderungen gemeinsam spezifizieren

Beispiel Flugbuchungssystem (Airplanes, Flights, Passengers)

ATDD Story: Booking a Flight

A flight LH-1234 is scheduled

It will be executed with a Boeing-787

There are 200 free seats

Eingabe

There is a passenger John Doe

Book seat for John Doe at LH-1234

Systemfunktion

Assume a valid ticket is issued

Ergebniserwartung

Ausführbare Tests automatisiert erzeugen

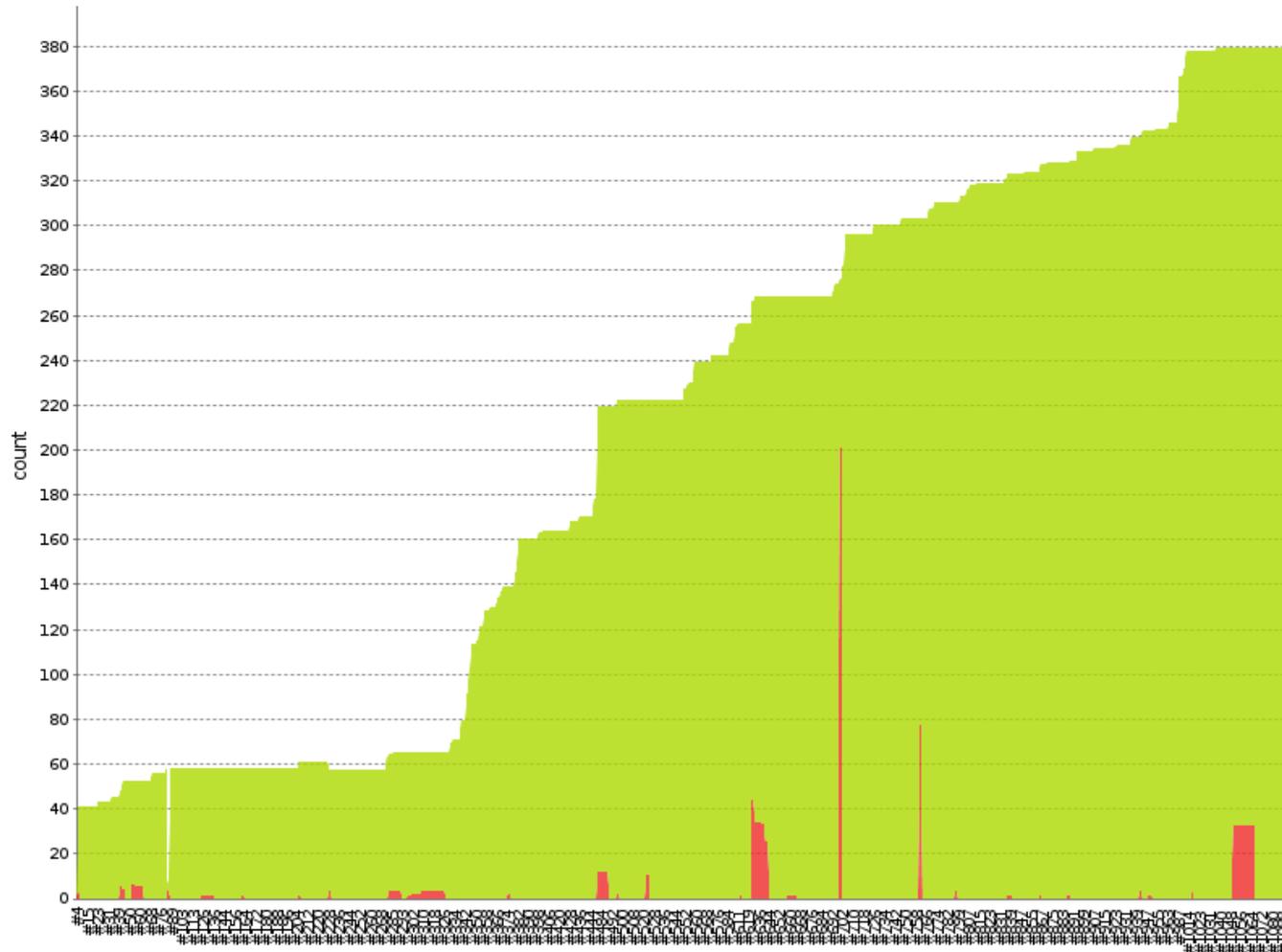
- Nur testbare (ausführbare) Anforderungen bleiben lebendig und sind wertvoll
- Testerzeugung muss direkt aus Anforderung erfolgen und darf diese nicht verändern
- Testerzeugung muss für Entwickler einfach und flexibel umsetzbar sein
- Testerzeugung muss auf bestehende Test-Frameworks aufsetzen (JUnit, TestNG, HtmlUnit, jMock, EasyMock, Arquillian, etc)

DEMO

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

Beispiel – Testentwicklung



Beispiel – Testentwicklung



Von ATDD zu Lean Modeling

- Konzeptuell analog zu ATDD, aber:
 - keine Beschränkung auf Testcode
 - keine syntaktischen Beschränkungen bzgl. der verwendeten Sätze
- Beispiele:
 - Businessregeln
 - Datenstrukturen
 - Oberflächen
 - Technische Dokumentationen

 - Schnittstellen
 - Workflows

DEMO

Vorteile von Lean Modeling

Klassisches "Fat" Modeling

Viele verschiedene Sprachen für spezielle Aufgaben

Formale Sprachen sind für Menschen schwerer zugänglich

Tracing zwischen Modell und Code ist aufwendig und technologisch schwierig

Konzepte und Werkzeuge sind komplex und erfordern Expertenwissen

Lean Modeling

Eine universelle Sprache mit einfacher Möglichkeit zur Spezialisierung

Natürliche Sprache ist für Menschen gemacht

Tracing zwischen Spezifikation und Code ist im Vorgehen "eingebaut"

Konzepte sind einfach und Ansatz integriert vertraute Werkzeuge

Nächste Schritte

Werkzeug ausprobieren

- NatSpec www.nat-spec.com - Kostenlose 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 Helleoy - 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.nat-spec.com>

tobias.nestler@devboost.de
christian.wende@devboost.de