

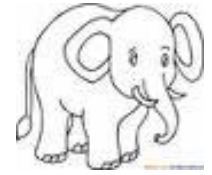


Von der Idee zum Erfolg.

10. ANECON Expertenfrühstück

Value Driven Testing oder Wie esse ich einen Elefanten?

Helmut Pichler | ANECON



Inhalt

- Orientierung - Was / Woher?
- Motivation / Ziele
- Voraussetzung/en
Eckpfeiler, Umfeld, Mitspieler
- Herangehensweise aus Sicht des
Test Managers
- Herausforderungen/Resumee

Orientierung - Was / Woher ?

- Value Driven Development → Value Driven Testing
 - Ist ein Ansatz, Softwareprojekte über den Nutzwert ihrer Ergebnisse zu steuern.
- Bei IT-Projekten zählt der Business Value
- Fixe Budgetrahmen – immer komplexere Anforderungen
- “Klassische” SW-Entwicklungsmethoden/Ansätze
- ↓
- Risiko orientierter Ansatz
- ↓
- Value Driven Ansatz

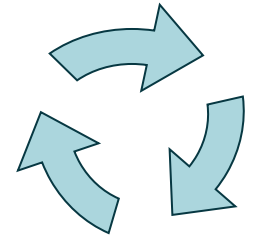
Inhalt



- Orientierung - Was / Woher?
- Motivation / Ziele
- Voraussetzung/en
Eckpfeiler, Umfeld, Mitspieler
- Herangehensweise aus Sicht des
Test Managers
- Herausforderungen/Resumee

Motivation / Ziele

- Rascher / Effizienterer Return of Invest (ROI)
- Störung der Wertschöpfung vermeiden
- Reduktion der Wartungskosten (WK)
 - WK mehr als 2/3 der Lebenszykluskosten (LZK)
 - 26-40 % der WK sind Reparaturkosten
 - **Fehlerbehebungskosten ca. 20 % der LZK**
- Kostenoptimierung & -transparenz bei Produkt- /Projektentwicklung
- Optimierter Einsatz von (Test-)Ressourcen
- Primär nur da investieren, wo Nutzen/Wert generiert wird



Motivation

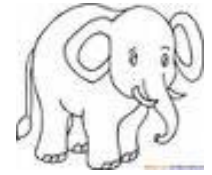
Wert

- Offensichtlich / direkt :



- Verborgen / indirekt:

Qualität



Inhalt

- Orientierung - Was / Woher?
- Motivation / Ziele
- Voraussetzung/en
Eckpfeiler, Umfeld, Mitspieler
- Herangehensweise aus Sicht des
Test Managers
- Herausforderungen/Resumee

Voraussetzungen

Eckpfeiler, Umfeld, Mitspieler (I)

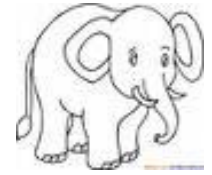
- Bewusste Entscheidung zum Value Driven/VD Ansatz:
 - Gesamtprojekt
 - alle Beteiligte
- Klare Vorgehensweise definiert:
 - Regelwerk
 - einheitliche Schätzmethode
- Eingespieltes Team
- Passendes Entwicklungs- und Test Framework



Voraussetzungen

Eckpfeiler, Umfeld, Mitspieler (II)

- Teilkomponenten/Stückliste liegen vor
 - Analyse, Design, Req-Engineering abgeschlossen
 - Klare, abgegrenzte/identifizierte Komponenten
 - Aufwandsschätzungen abgeschlossen und idealerweise Gewichtung durchgeführt
- Klare Vorstellung der künftigen Live-Einsatzumgebung
 - Umgebung
 - Features (meistgenutzte, businesskritische)
 - Kundenzufriedenheit



Inhalt

- Orientierung - Was / Woher?
- Motivation / Ziele
- Voraussetzung/en
Eckpfeiler, Umfeld, Mitspieler
- Herangehensweise aus Sicht des
Test Managers
- Herausforderungen/Resumee

Spannungsumfeld: Nutzen des Tests

Berechnung des Test ROI

Test ist eine Software Engineering Aktivität, die Kosten, aber auch Nutzen bringt.

$$\text{Test-ROI} = (\text{Test Nutzen} - \text{Test Kosten}) / \text{Test Kosten}$$

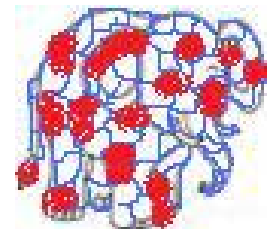
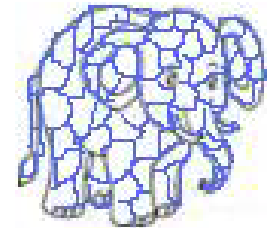
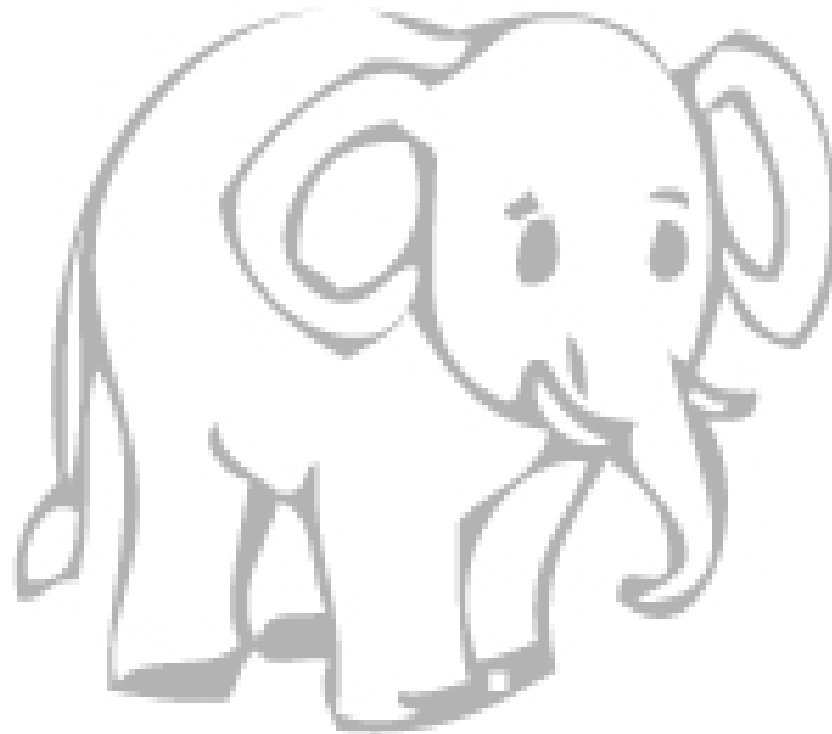
Test Kosten:

- Personal-, Software (Tools)-, Hardware Kosten

Test Nutzen:

- Kosten, die die beim Test gefundenen Fehler verursacht hätten
also: gesparte Korrekturkosten + gesparte Ausfallkosten

Herausforderung: Testkosten



Testmanager ToDo's am Beginn

- Anforderungsdokumente “brauchbar”, klar, vollständig?
- Testkonzept erstellen
 - Schwerpunkt Kostentransparenz & Test ROI
 - In Teststrategie VD Vorgehensweise definieren
 - Aufwands-Controlling (Reports, Steuerung)
 - Welche brauche ich, welche liefere ich
 - Wie kann ich Einsparungen transparent machen

Testmanager ToDo's

während des Testprojekts - Test Kosten

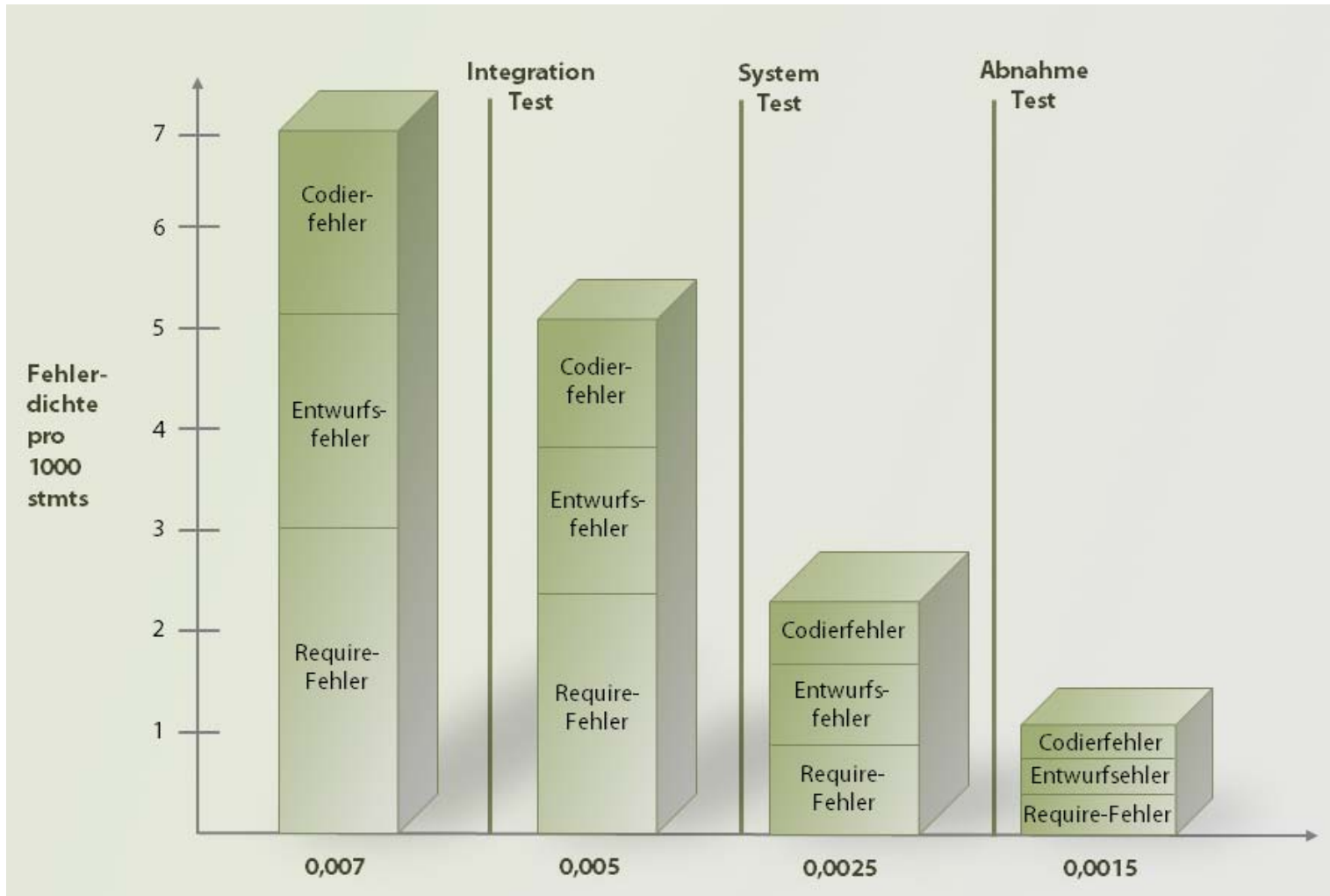
- Test Kosten/Aufwand pro identifizierter Anforderung ermitteln
 - Größe der Komponente/Modul [FP, LOC, ...]
 - Testtiefe und -Intensität festlegen
- Statistiken heranziehen:
 - Fehlerfindung (aus Erfahrungen früherer Projekte,...)
 - Fehler /Tag, Prio und Tester ggf. phasenbezogen
- Projektumfeld beachten (Domain, Größe)

Testmanager ToDo's

während des Testprojekts - Test Nutzen

- Statistische Grundlagen (allgemeine, aus Vorgängerprojekten)
- Fehlerauswirkung der Anforderung/Komponente identifizieren
 - (Risikoanalyse & Kritikalität)
 - Größe der Komponente/Modul
 - möglicher Schaden:
 - Kosten, wenn Komponente/System eingeschränkt läuft
 - Kosten bei Komplettausfall
 - Anzahl Benutzer
 - Einsatzumgebung
 - Time to market

Fehlerrate



Quelle:
SW Institut,
2004

Kosten der Fehler Korrektur

(aus der Test Literatur)

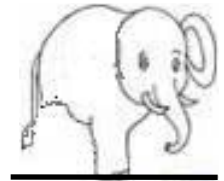
- **Fagan, IBM Federal Systems Division, 1977**
Coding = 1, Test = 12, Production = **28 Hours**
- **Boehm, B.: in Software Engineering Economics, 1981**
Coding = 1, Test = 16, Production = **60 Hours**
- **Remus, IBM Santa Theresa Labor, 1983**
Coding = 1, Test = 20, Production = **82 Hours**
- **Stephan Kan, IBM AS400 Laboratory, 1995**
Coding = 1, Test = 13, Production = **92 Hours**
- **Maria-Theresa Teichmann, QS-Managerin in GEOS Projekt, 2007**
Coding = 1, Test = 8, Produktion = **24 Hours**

Anm: Korrekturaufwand in Maintenancephase ähnlich wie in der Produktionsphase

Testmanager ToDo's

während des Testprojekts - Test Nutzen

- Statistische Grundlagen (allgemeine, aus Vorgängerprojekten)
- Fehlerauswirkung der Anforderung/Komponente identifizieren
 - (Risikoanalyse & Kritikalität)
 - Größe der Komponente/Modul
 - möglicher Schaden:
 - Kosten, wenn Komponente/System eingeschränkt läuft
 - Kosten bei Komplettausfall
 - Anzahl Benutzer
 - Einsatzumgebung
 - Time to market



Testmanager ToDo's

während des Testprojekts - Entscheiden/Reporten

Testkosten



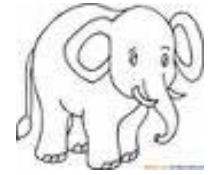
Ausfall-, und Fehlerbehebungs-kosten im Produktivbetrieb

→ Test der Komponente durch-/weiterführen oder einstellen

- Controlling (regelmäßige Reports, Steuerung)
 - Alle aufgedeckten Fehler (Folgekosten bzw. Testnutzen)
 - Testkosten (dem Testnutzen gegenübergestellt)
 - Basis für Steuerung der weiteren Testdurchführung

→ Einsparungen/Vorgehen transparent darstellen

Inhalt



- Orientierung - Was / Woher?
- Motivation / Ziele
- Voraussetzung/en
Eckpfeiler, Umfeld, Mitspieler
- Herangehensweise aus Sicht des
Test Managers
- Herausforderungen/Resumee

Herausforderungen

- (Prozess-)Reife der Stakeholder nötig
 - Erfahrungswerte (Fehlerdichte, Findungsrate, etc.)
- Detailliertes Wissen erforderlich
 - des zu entwickelnden Produktes
 - der Einsatzbedingungen im Livebetrieb
(# Endnutzer, deren Verhalten & Bedürfnisse / Umfeld)
- Kosteneinsparung; der Test zielt auf jene Fehler, die die größten Risiken in sich bergen und die höchsten Reparaturkosten verursachen
- Controlling-, Steuer- und Reportingaufwand
 - Transparenz braucht Konsequenz

Resumee

Value Driven Testing ist ein erfolgsversprechender Ansatz, wenn:

- granular & eindeutig beschriebene Anforderungen vorhanden,
- Gesamtprojekt auf Value Driven ausgerichtet,
- Kosten / Nutzen ermittelt und dem Kunden transparent gemacht werden kann,
- erfahrener/s Test Manager/Team vorhanden,
- Testeffizienz ggf. durch Testautomatisierung steigerbar,
- Mut, den Test wirklich abubrechen, wenn der Nutzen nicht mehr gegeben ist.



Software Test mit optimalem Kosten/Nutzen-Faktor

ANECON Software Design und Beratung G.m.b.H.

Alser Straße 4 / Hof 1 | A-1090 Wien | www.anecon.com

office@anecon.com | Tel.: +43 1 409 58 90 - 0 | Fax: -998

