

# T-Systems Multimedia Solutions

... T ... Systems .....

# Testmanagement in Theorie und Praxis.





## Agenda

1. T-Systems Multimedia Solutions – Test and Integration Center.
2. Grundbegriffe des Testens.
3. Warum testen?
4. Application Lifecycle und damit einhergehende QS-Maßnahmen.
5. Der Testprozess und Aufgaben eines Testmanagers.
6. Auch so kann's gehen – Ein Beispiel aus der Praxis.
7. Überblick ausgewählter Testarten.
8. Data Privacy and Security.

# 1. T-Systems Multimedia Solutions.

# AUS GUTEM HAUSE.

- Die T-Systems Multimedia Solutions GmbH ist eine Tochter der T-Systems.
- Uns gibt es seit 1995.
- Wir sind Deutschlands Marktführer bei internetbasierten Lösungen.\*
- Mit über 980 Kollegen realisieren wir jährlich 1.400 komplexe IT-Projekte.
- Und wir wachsen weiter: wir stellen permanent neue Mitarbeiter ein.

\* New Media Service Ranking 2010 (BVDW)







E-Procurement  
Online-Marketing  
Certification  
Support Services  
E-Services  
**Testing Services & Support**  
Portale  
Business Processes  
Code Quality Management  
IT Strategy and Service  
Video-On-Demand  
Security  
Business Strategies  
Mobility  
Communities  
Project Management  
Content Management  
**E-Entertainment**  
**Consulting**  
Shopsysteme  
Pay Media  
Technology Consulting  
**E-Workplace**  
Application Integration  
Communications  
Event TV  
Web TV/IPTV  
Collaboration  
Mediathek

.. **T** .. **Systems** ..



# Test and Integration Center. Akkreditiertes Software- Prüflabor

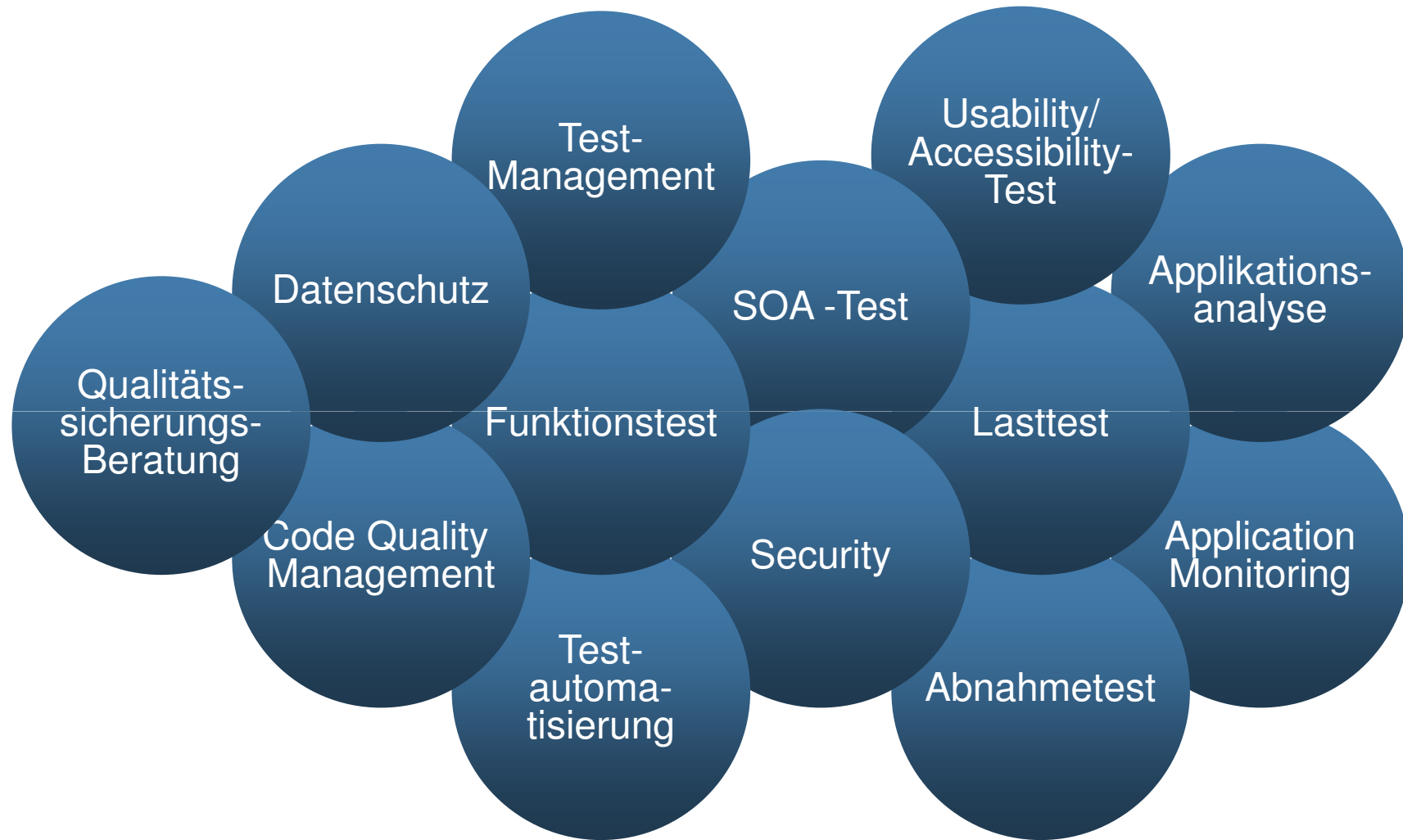
Das TIC der T-Systems MMS ist ein akkreditiertes Softwareprüflabor der Bundesrepublik in der Internet- und Multimediabranche. Die Akkreditierung erfolgte durch die Deutsche Akkreditierungsstelle Technik (DATech e.V.), die Mitglied im Deutschen Akkreditierungsrat (DAR) ist.



Durch die Akkreditierung wird dem TIZ der T-Systems MMS bescheinigt:

- Das Testlabor arbeitet nach ausgereiften Methoden und Verfahren.
- Die Mitarbeiter verfügen über hervorragende Kompetenzen.
- Die Infrastruktur des Testlabors erfüllt die hohen technischen Anforderungen.
- Das Management und die Prozesse sind auf einem hohen Niveau.

# LEISTUNGSPORTFOLIO.



## 2. Grundbegriffe des Testens.

# WAS IST EIN FEHLER?

Fehler:

- Eine Abweichung zwischen dem gemessenen Wert / Zustand und dem entsprechenden spezifizierten Wert.
- Nichterfüllung einer Anforderung
- Differenz zwischen Soll- und Ist-Verhalten

Ähnliche Begriffe:

- Abweichung
- Mängel



# TEST: WAS IST DAS EIGENTLICH?

Was Test NICHT ist

Testen kann nicht garantieren, dass die Software fehlerfrei ist

Man kann nicht alles testen.

Man kann ein schlechtes Produkt nicht "gut"-testen.

Testen ist nicht Debugging / Testen identifiziert nicht die Fehlerursache

# TEST: WAS IST DAS EIGENTLICH?

Gesamter Prozess, ein Programm auf systematische Weise auszuführen, um die korrekte Umsetzung der Anforderungen nachzuweisen und Fehlerwirkungen aufzudecken.

Vergleich eines Zustandes des Testobjektes mit dem gewünschten bzw. verlangten Zustand.

Testen ist der Prozess, ein Programm mit der Absicht auszuführen, Fehler zu finden.



.. **T** .. Systems ..

# ZIEL DES TESTENS

Finden von Fehlern

Steigerung der Softwarequalität

Senkung der Fehlerrisiken

Erfüllung von Normen / Vorschriften / Auflagen

Warum?

- SW-Komplexität
- SW-Kritikalität (ohne Programme funktioniert kaum was noch)
- Fehler sind SEHR teuer

# WARUM GIBT ES ENTWICKLER UND TESTER?

Ein bisschen Psychologie

Unabhängiges Testen (Objektivität behalten)

Test soll von den Personen durchgeführt werden, die nicht an der Programmierung der Software-Module beteiligt waren!

Entwickler sind die Theoretiker im Projektteam. Sie entwickeln die Ideen und glauben gern, dass die ganze Welt so funktioniert, wie sie sich das vorstellen. Wenn Entwickler selbst testen, suchen sie nach einer Bestätigung der Gültigkeit ihrer theoretischen Überlegungen.

Tester sind die Praktiker. Sie wollen nachweisen, dass das Programm unter bestimmten Bedingungen versagt.

Tester bringen schlechte Nachrichten → Konfliktpotential



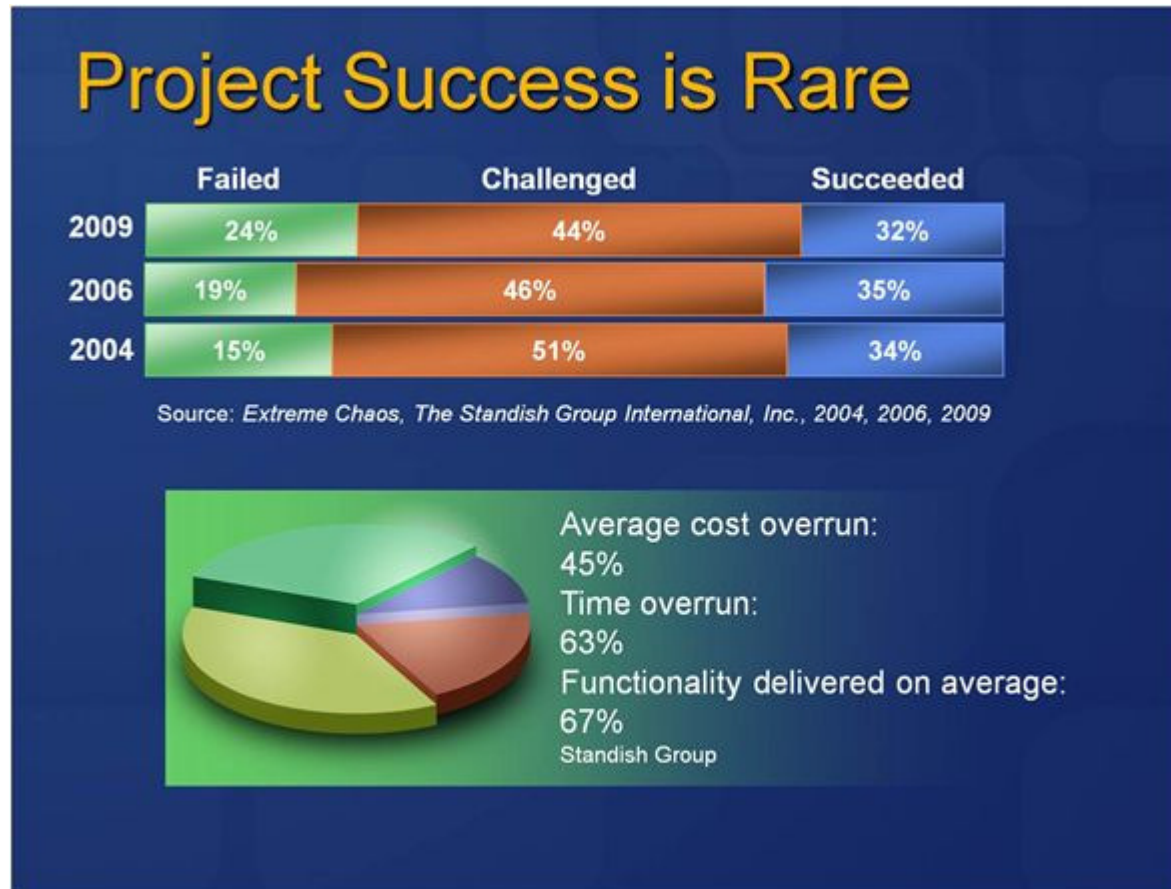
# 3. Warum testen?

# Warum testen?



..T..Systems.....

# Warum testen?



Quelle: Standish Group, CHAOS Report 2009: „Chaos History“ von IT-Projekten.

# Was kosten Fehler?

- Geschätzte Verluste durch Softwarefehler in Mittelstands- und Großunternehmen in Deutschland: ca. 84,4 Mrd. € p.a.
- 35,9% der IT-Budgets für Beseitigung von Programmfehlern entsprechen 14,4 Mrd. € p.a.
- Produktivitätsverluste durch Computerausfälle aufgrund fehlerhafter Software: ca. 2,6% des Umsatzes - 70 Mrd. € p.a.

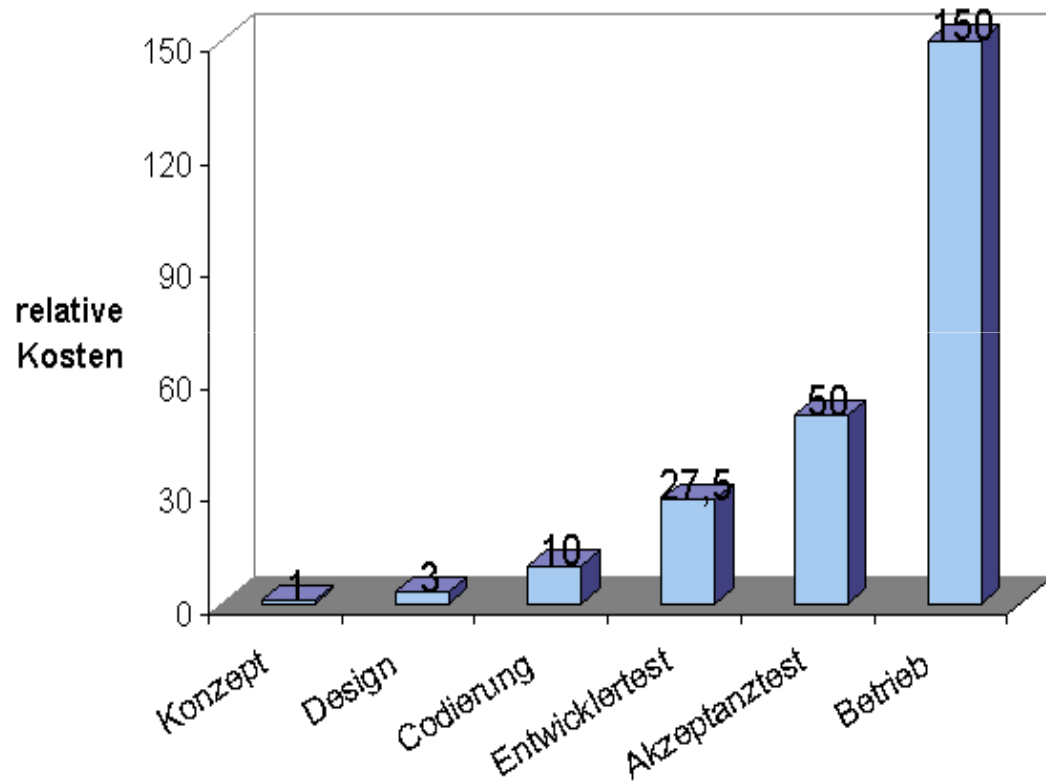
Quelle: Studie der LOT Consulting Karlsruhe, IT-Services 3/2001

- allein in Deutschland Ausfälle von jährlich 85 Milliarden Euro durch mangelnde oder unzureichende Testing - Maßnahmen.

Quelle: Forrester Research, 2003



# Was kostet die Fehlerbeseitigung?



Giga Information Group:

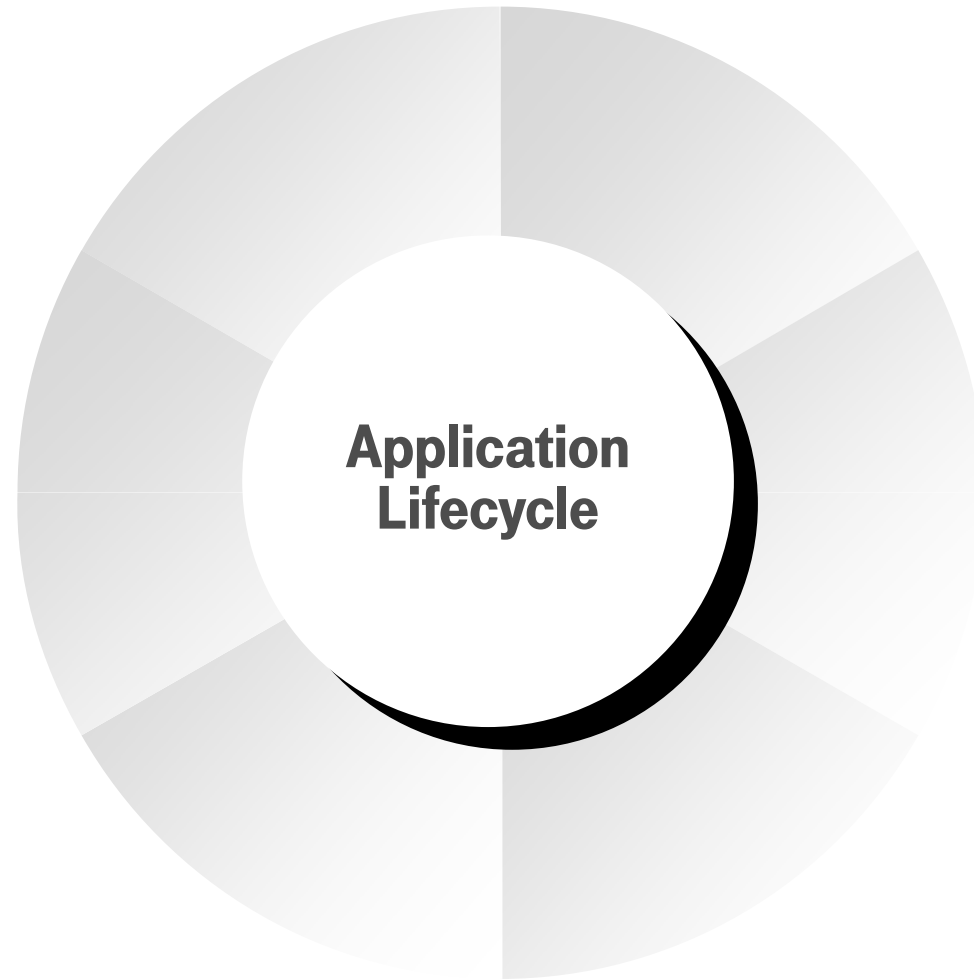
Softwarefehler, die erst in der Post-Production-Phase entdeckt werden, verursachen 50 Prozent höhere Kosten als wenn sie bereits in der Entwicklung behoben worden wären.

.. T .. Systems ..

## Was kostet die Fehlerbeseitigung?

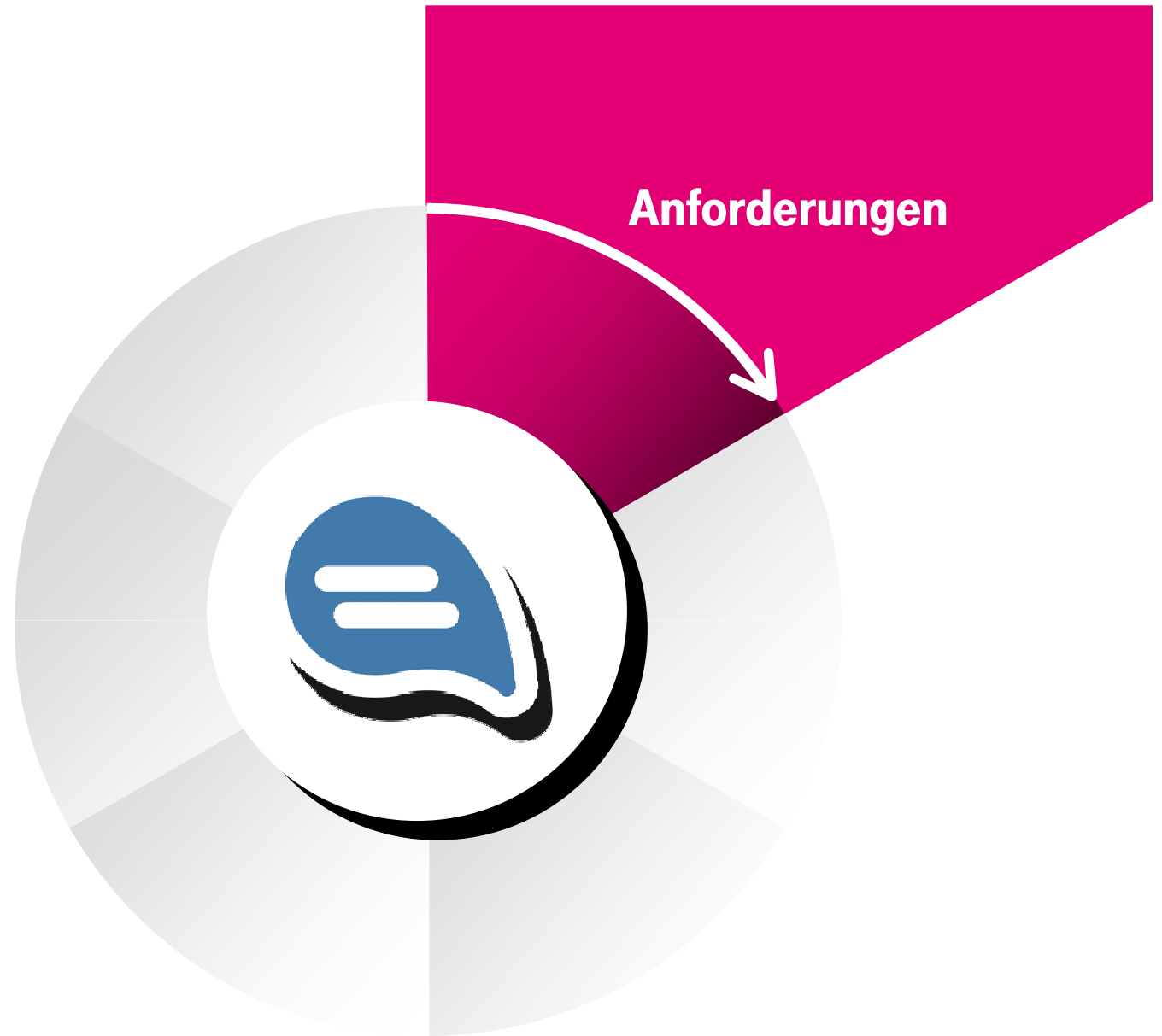
- |                                |          |  |
|--------------------------------|----------|--|
| ■ Istanalyse / Grobkonzept     | 1.000 €  | für Überarbeitung<br>Geschäftsprozess, Review,<br>Diskussion |
| ■ Feinkonzept / DV-<br>Konzept | 4.000 €  | für Überarbeitung<br>Geschäftsprozess, Review,<br>Diskussion |
| ■ Realisierung                 | 12.000 € | für zusätzliche<br>Entwicklungskosten                        |
| ■ Abnahme / Einführung         | 48.000 € | für zusätzliche<br>Abnahmekosten,<br>wiederholtes Roll out   |
| ■ Betrieb                      | 90.000 € | Umsatzverlust, Imageschaden                                  |

## 4. Application Lifecycle und damit einhergehende QS-Maßnahmen.

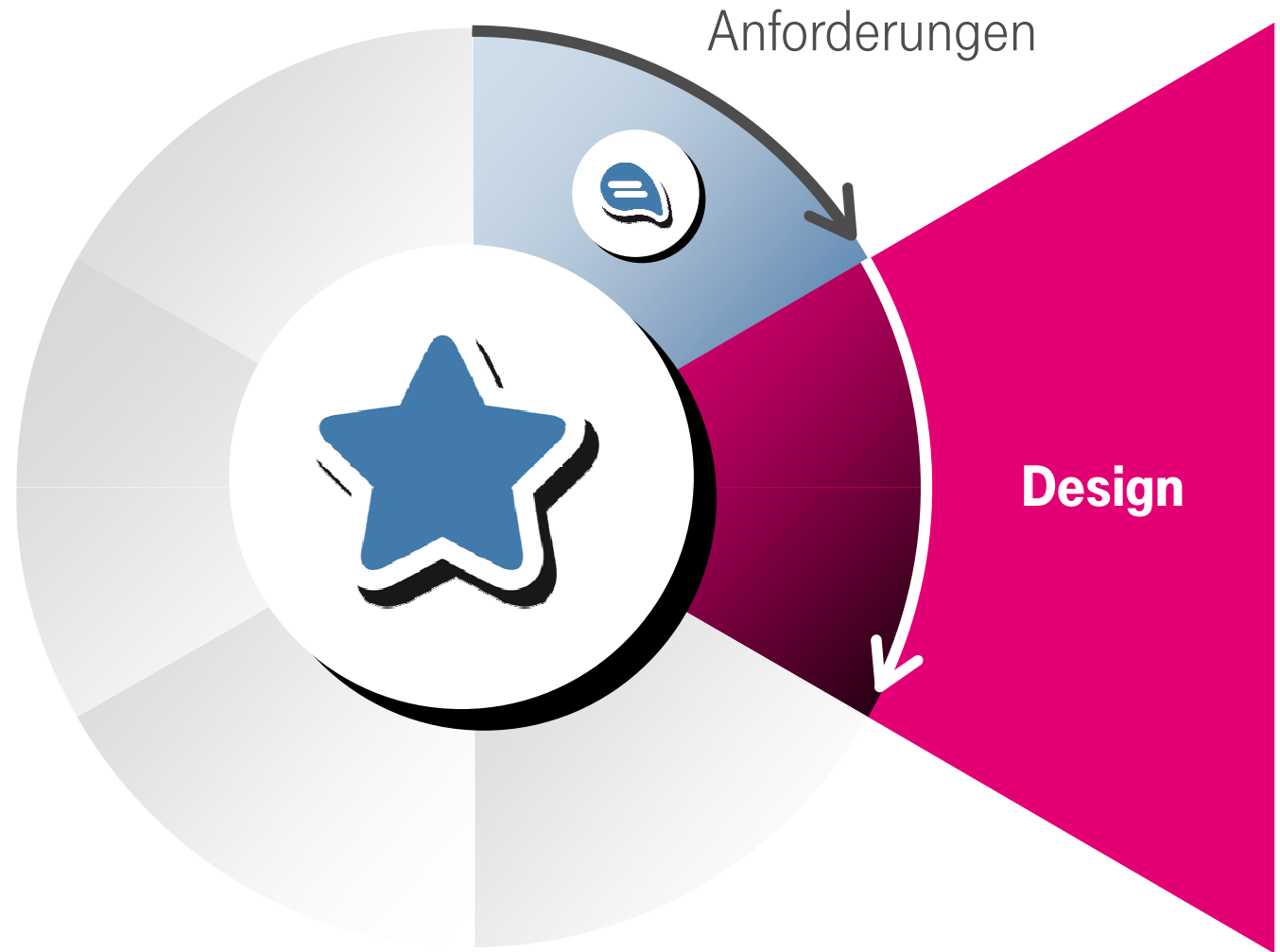


... **T** ... Systems ...

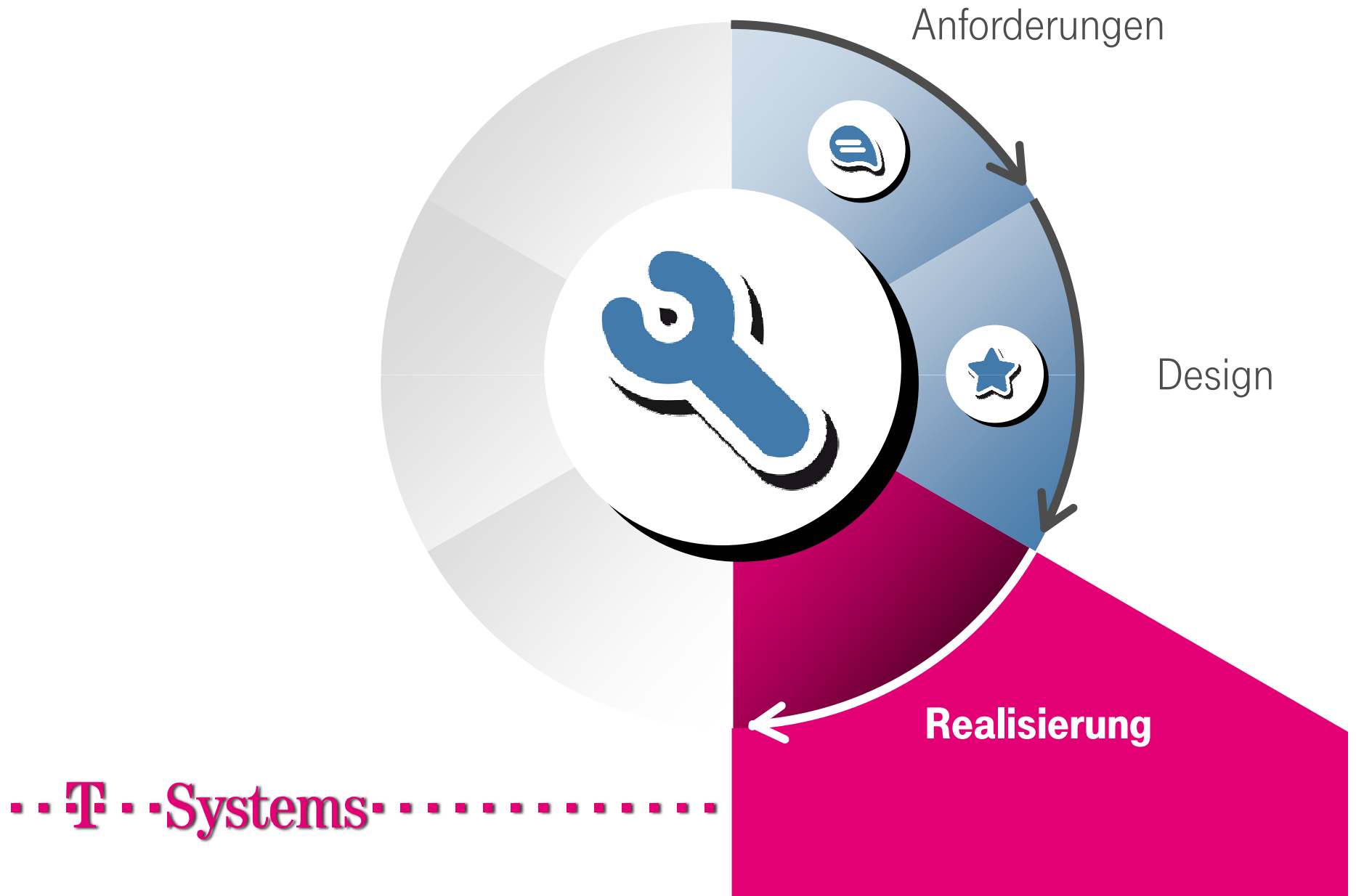


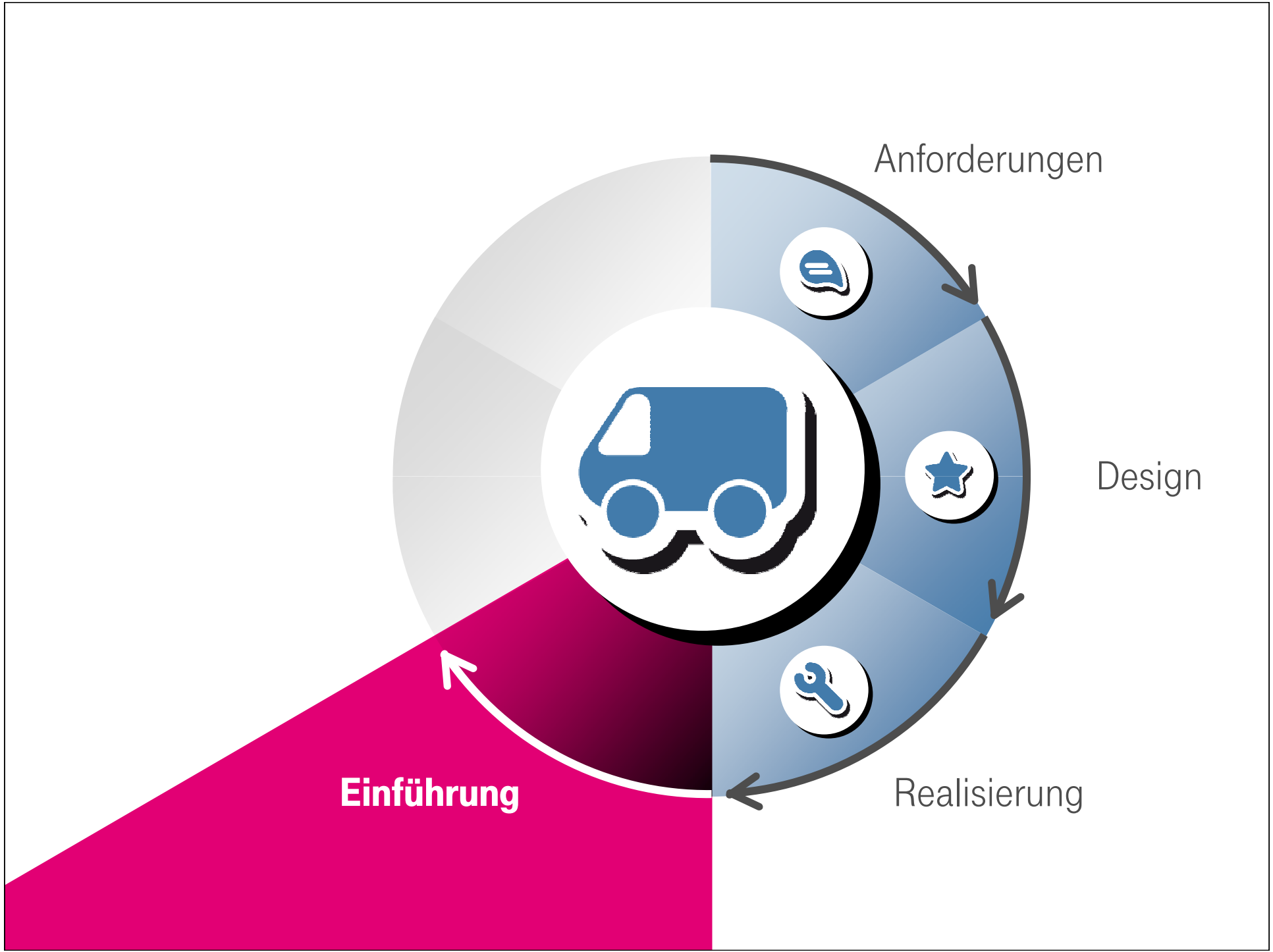


...T...Systems.....



... **T** ... Systems ...



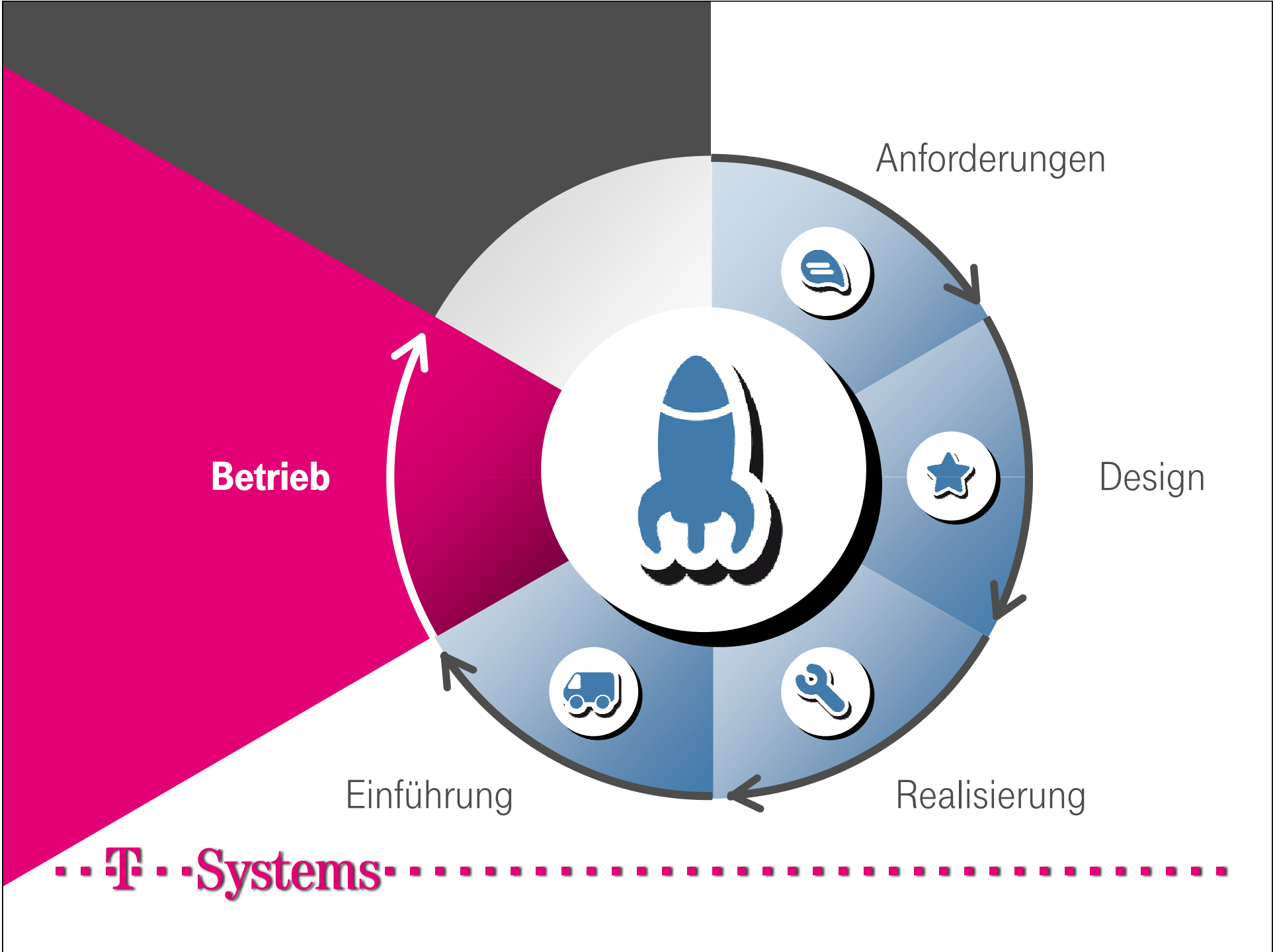


Anforderungen

Design

Realisierung

**Einführung**



Optimierung

Anforderungen

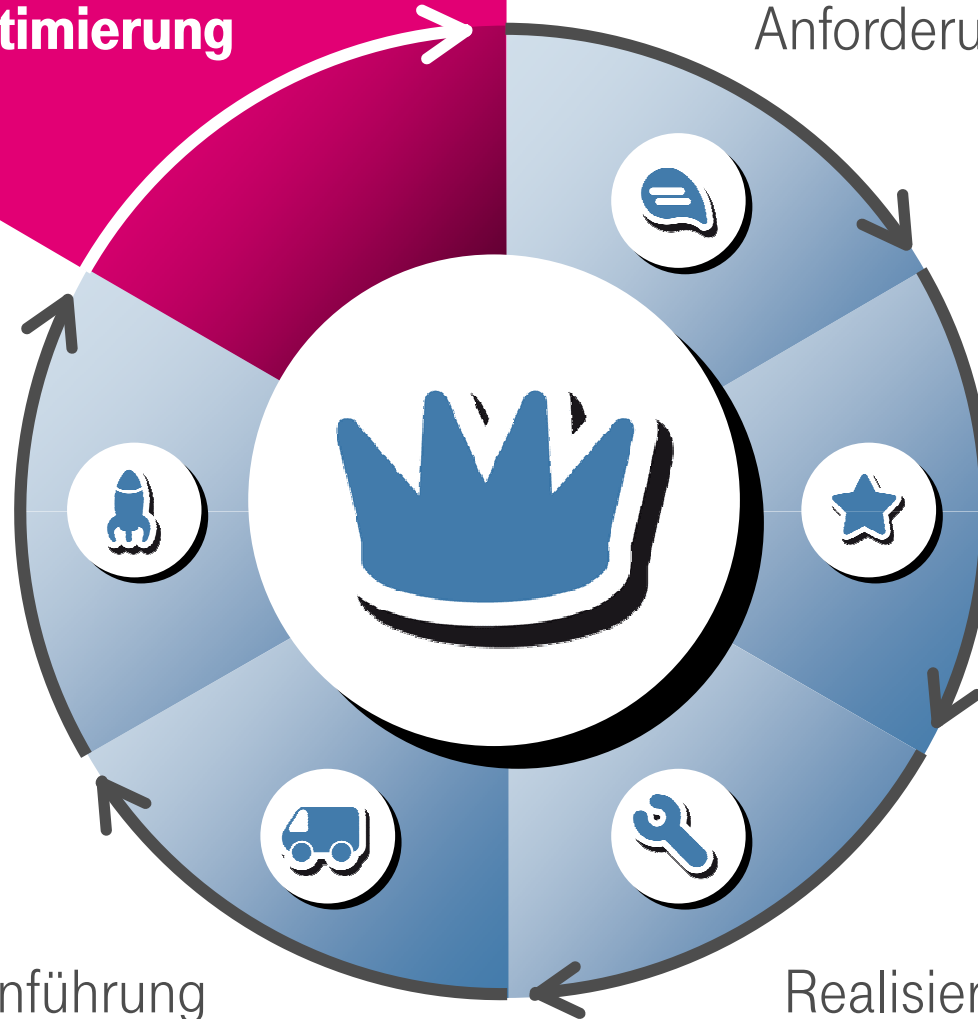
Betrieb

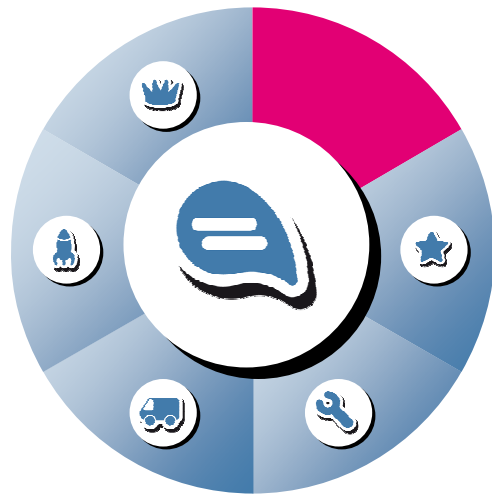
Design

Einführung

Realisierung

... **T** ... Systems ...





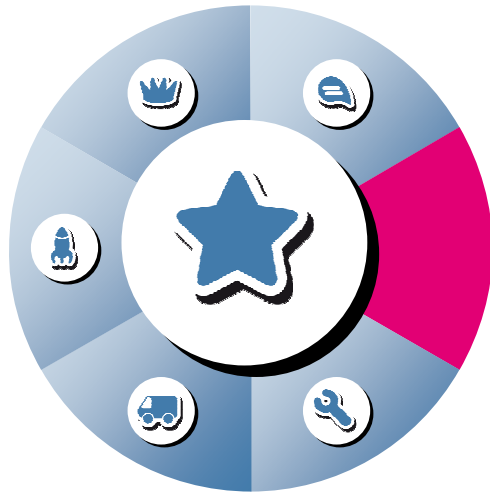
## Anforderung: Weichenstellung für Qualität

### Was ist aus Qualitäts- / Test-Sicht relevant?

- Vollständige Beschreibung der Anforderungen
- Abnahmekriterien je Anforderung / UseCase und für SW insgesamt definieren
- Testplanung / -organisation (Prozesse, Personal, Skills, Infrastruktur)

### Quality-Gate

	ja	nein
Sind alle Anforderungen vollständig, eindeutig und konsistent beschrieben?	<input type="checkbox"/>	<input type="checkbox"/>
Ist das gemeinsam getragene Vorgehen zur Qualitätssicherung bei allen Beteiligten sichergestellt?	<input type="checkbox"/>	<input type="checkbox"/>



## Design: Fehler frühzeitig finden und beseitigen

Was ist aus Qualitäts- / Test-Sicht relevant?

Review des Designentwurfes

DV technische Umsetzung der Anforderungen prüfen

Architektur, Security, Usability, Accessibility

Tests konzipieren

Quality-Gate

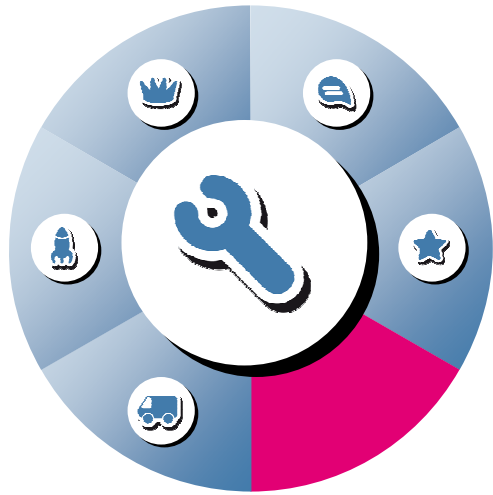
Finden sich alle Anforderungen vollständig, widerspruchsfrei im Design wieder?

ja    nein

Ist der Designentwurf korrekt?



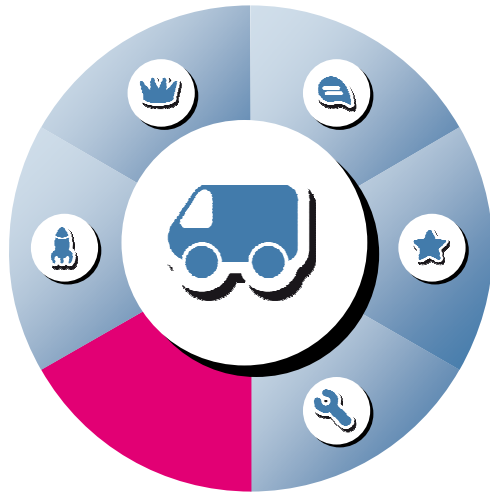


## Realisierung: Frühzeitiges, isoliertes Testen

- Was ist aus Qualitäts- / Test-Sicht relevant?
- Code Quality Management, Codereviews
  - Tests spezifizieren, vorbereiten, automatisieren
  - Entwicklungsbegleitende Tests, Testaufbau: vom Modul zur Integration
  - Security und Performanceaspekte reviewen

### Quality-Gate

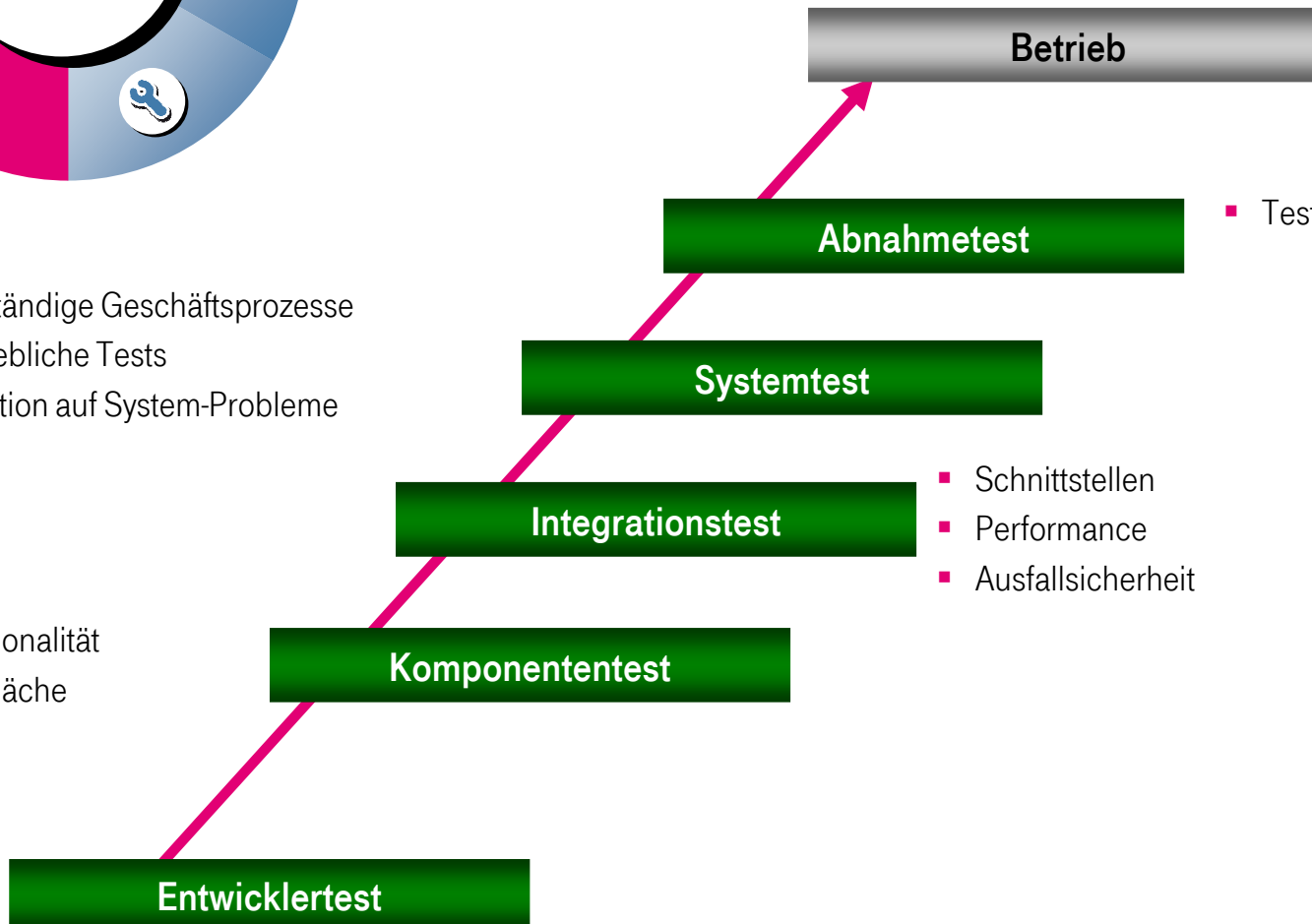
	ja	nein
Codequalität entspricht den Vorgaben	<input type="checkbox"/>	<input type="checkbox"/>
Testabdeckung ausreichend	<input type="checkbox"/>	<input type="checkbox"/>



# Einführung: Prüfen auf Herz und Nieren

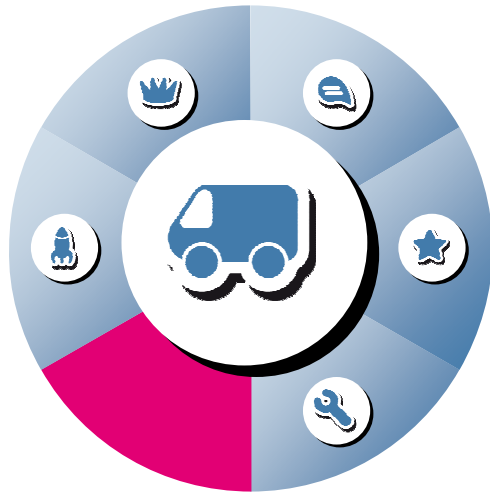
- Vollständige Geschäftsprozesse
- Betriebliche Tests
- Reaktion auf System-Probleme

- Funktionalität
- Oberfläche



- Schnittstellen
- Performance
- Ausfallsicherheit

- Test gegen Anforderungen



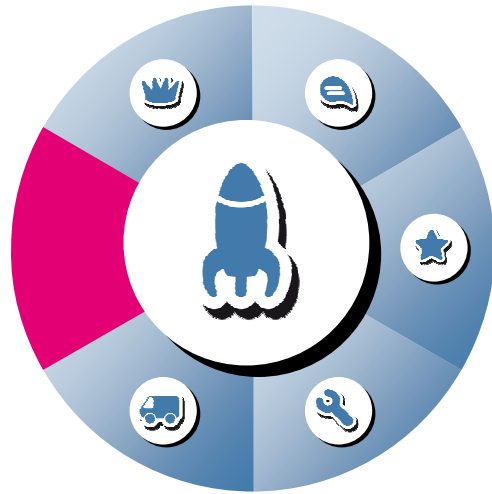
## Einführung: Prüfen auf Herz und Nieren

### Was ist aus Qualitäts- / Test-Sicht relevant?

- Durchführung Integrations-/Systemtests zu Funktionalität, Last & Performance, Security, ...
- Ausreichende Testabdeckung der Anwendung
- Effizientes Fehlermanagement

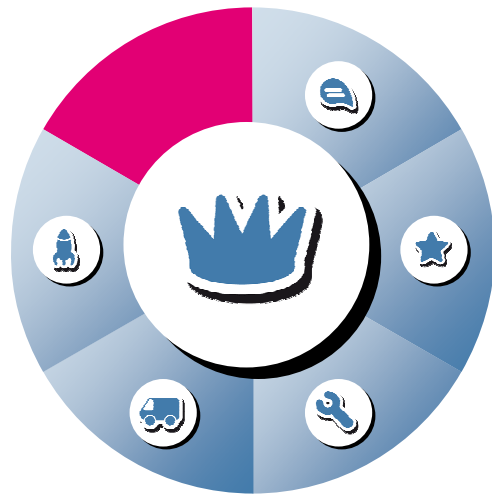
### Quality-Gate

	Ja	Nein
Wurden alle Tests in dem geplanten Umfang durchgeführt?	<input type="checkbox"/>	<input type="checkbox"/>
Wurde die geplante Testabdeckung erreicht?	<input type="checkbox"/>	<input type="checkbox"/>
Lassen die bekannten noch offenen Fehler einen Wirkbetrieb zu (Fehlerkorridor)?	<input type="checkbox"/>	<input type="checkbox"/>



## Betrieb: Qualität nachhaltig managen

- Was ist aus Qualitäts- / Test-Sicht relevant?
- Verfügbarkeit, Funktion, Performance der Anwendung überwachen
  - Analyse von Störungen, Performanceproblemen
  - Tests neuer Releases; Regressionstests
  - Performancetests bei Hardwareänderungen / Plattformausbau
  - IT-Sicherheitsmechanismen prüfen / anpassen



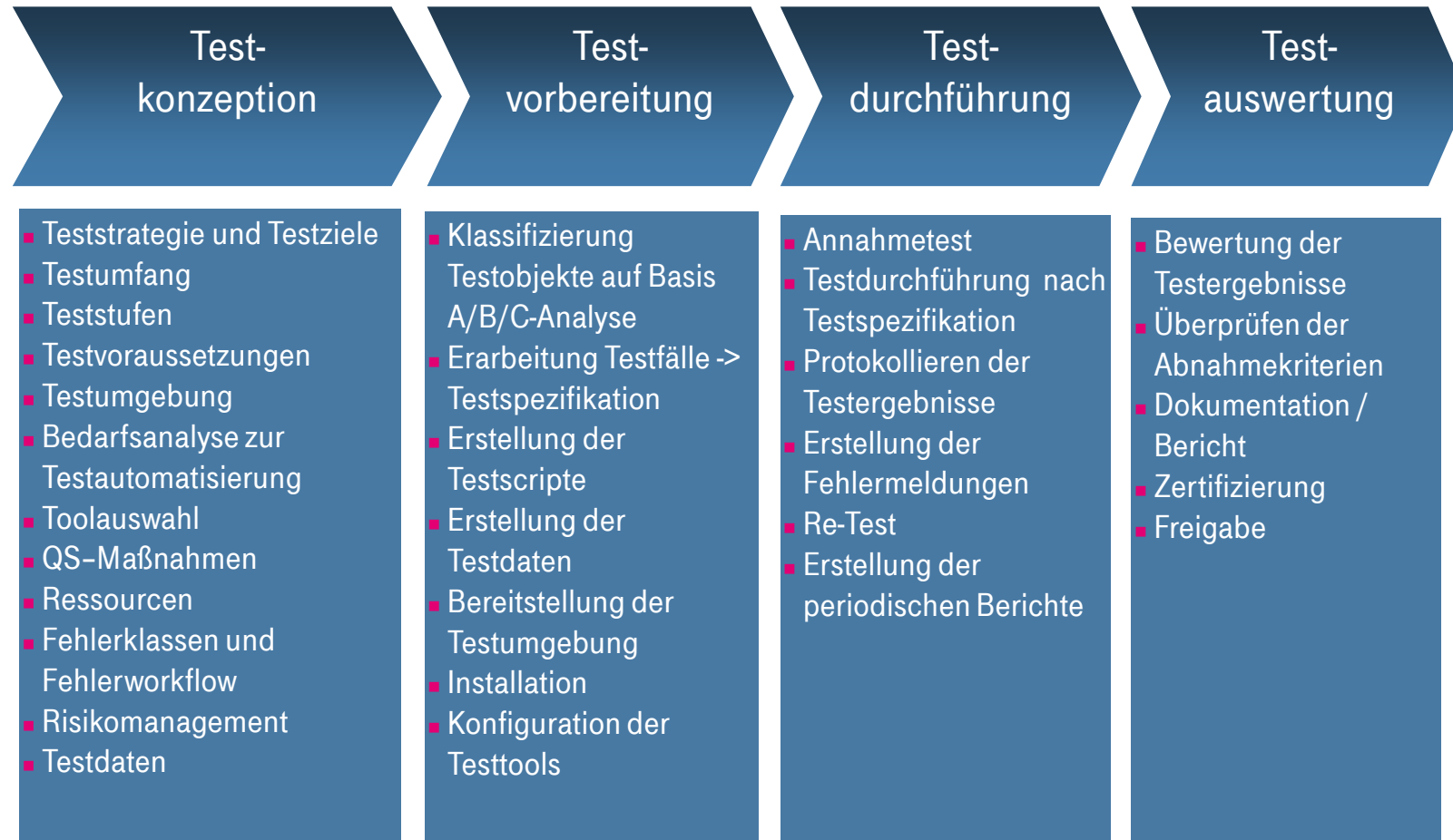
## Optimierung: Verbesserungen ableiten

Was ist aus Qualitäts- Sicht relevant?

- Sammeln, verdichten und aufbereiten der durch die QS-Maßnahmen gewonnenen Informationen als Input für Optimierungen, u.a.
  - CQM: Codemetriken
  - Wirkbetriebsfehler, Fehleranalysen
  - Erkenntnisse aus L&P, Security, Usabilitytests
  - Monitoringdaten

# 5. Der Testprozess und Aufgaben eines Testmanagers.

# Testvorgehen in der T-Systems MMS.



.. T .. Systems ..

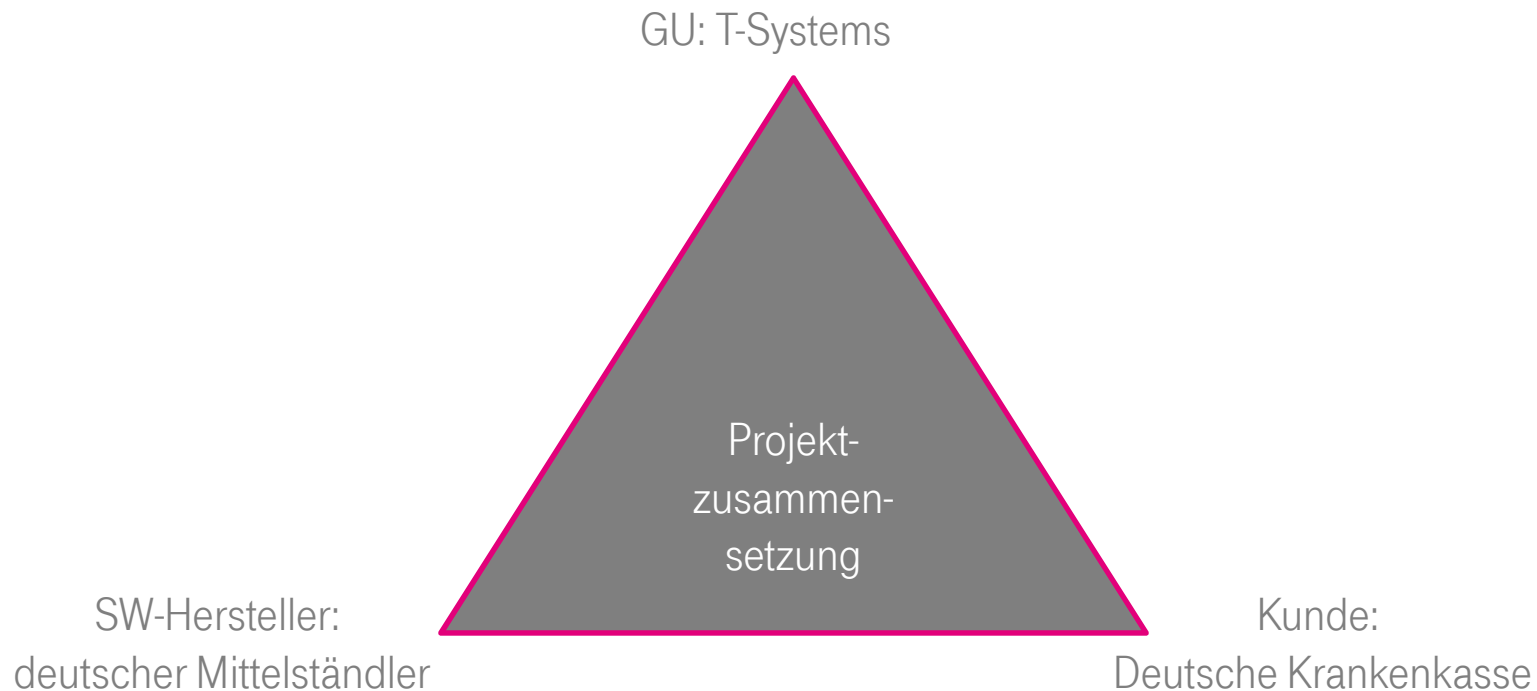




6. Auch so kann's gehen – Ein Beispiel  
aus der Praxis.

# Ausgangssituation.

- Ziel des SWE-Projektes: Bereitstellung einer Online-Plattform für die elektronische Genehmigung und Abrechnung von ärztlichen Verordnungen im Krankenkassenumfeld



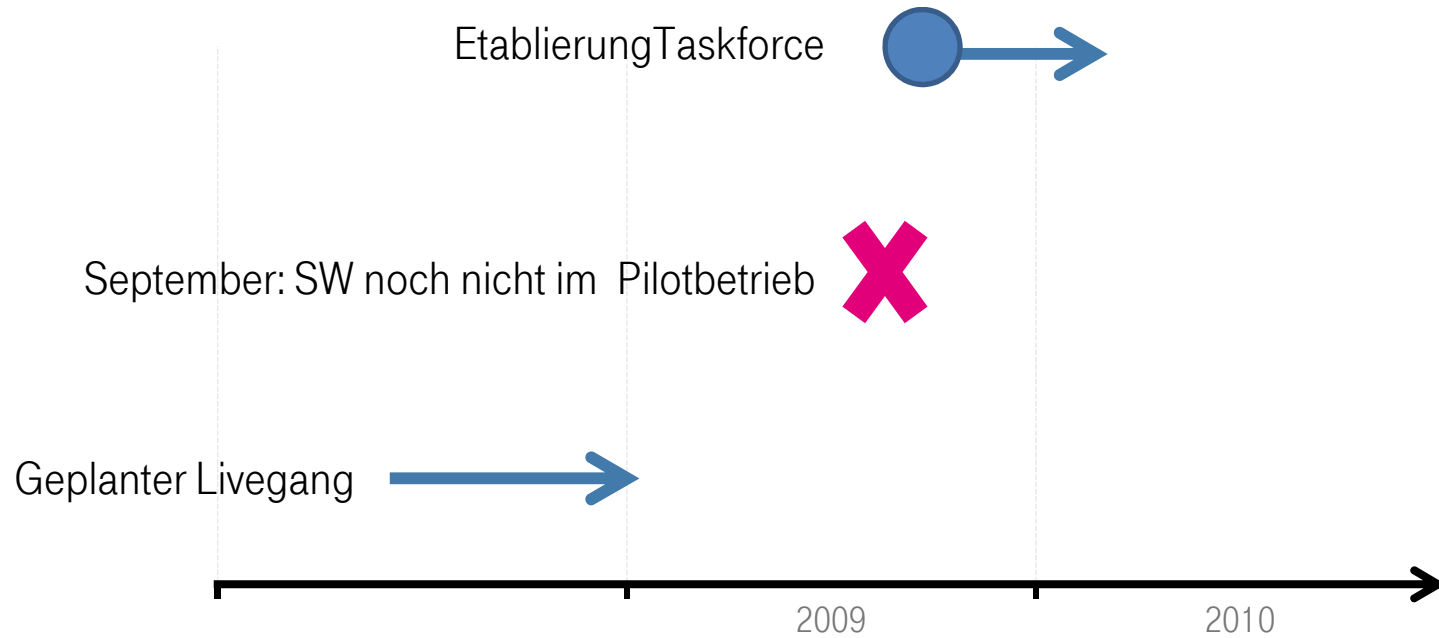
... **T** ... **Systems** ...

# Ausgangssituation.

- Keine Erkenntnis bis 2009, dass Testmanagement essentiell wichtig für den Projekterfolg ist
  - früherer Versuch der Etablierung eines Testmanagements schlug fehl
- Kunde war unzufrieden aufgrund unklarer Situation hinsichtlich Anforderungsumsetzung und Fehler-Status
- Keine Aussage möglich, wann Software in den Pilotbetrieb gehen kann
- Alleiniger Test durch SW-Unternehmen auf Basis formloser Entwicklertests, wobei Testautomatisierung vernachlässigt wurde
- SW-Unternehmen hat keine Erfahrung bzgl. Umgang mit Großkunden und komplexen Projekten

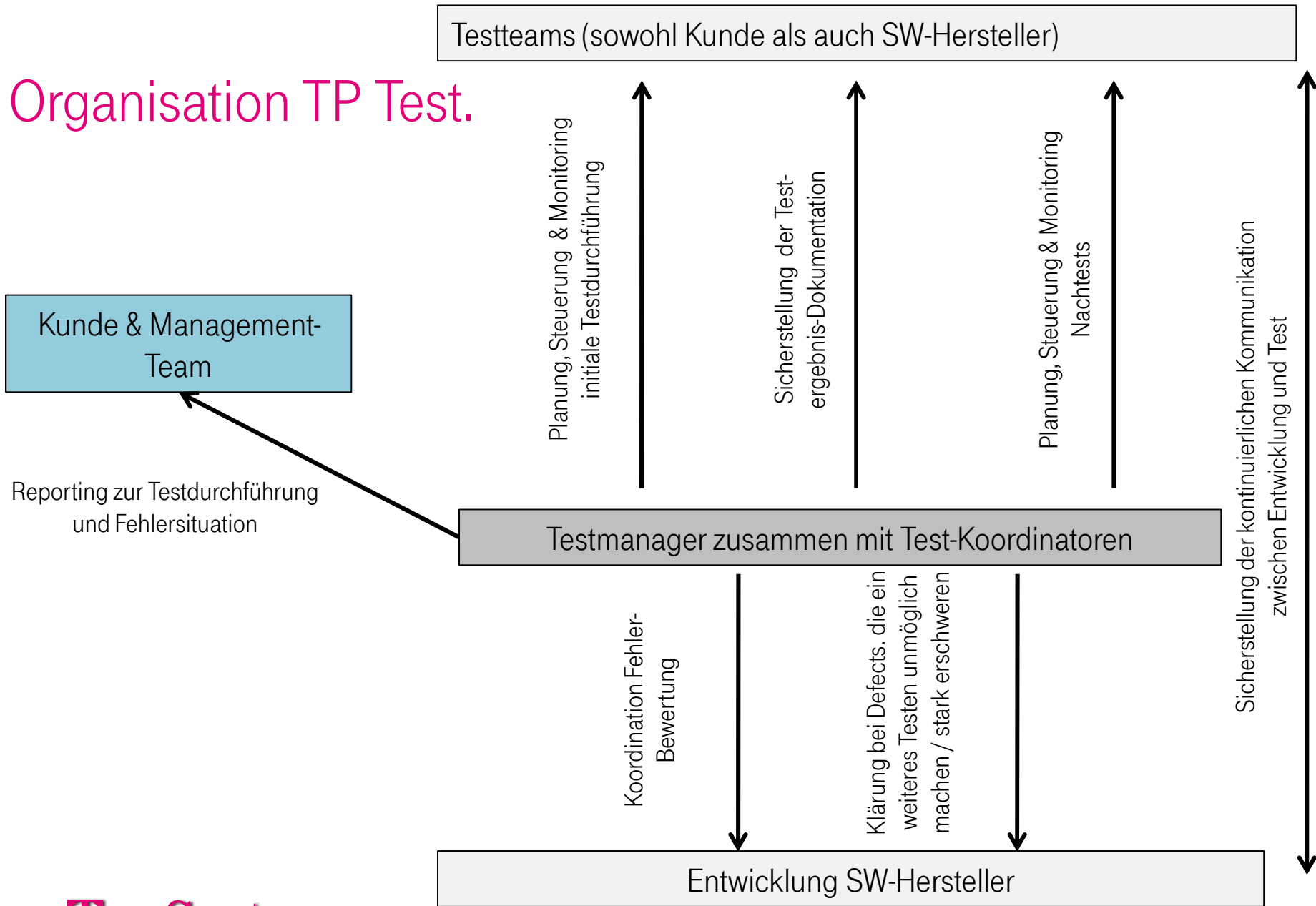
.. **T** .. **Systems** ..

# Ausgangssituation.



... **T** ... Systems ...

# Organisation TP Test.





Analyse der Ist-Situation und  
identifizierte Verbesserungspotentiale.

.. T .. Systems ..

Teilprozess	Output	Dokumenteinsicht	Bewertung	Antworten	Punkte
Testplanung / -steuerung	Werden die zu testenden <b>SW-Qualitätsmerkmale</b> (Funktionalität, Zuverlässigkeit, Benutzbarkeit, Effizienz, Änderbarkeit, Übertragbarkeit) für jedes Testprojekt festgelegt und priorisiert?		nein	nicht im Einzelfall spezifiziert, Funktionstest werden selbst durchgeführt (nach Entwicklung, tlw. durch Entwickler selbst), kompletter Durchlauf von Prozessen	0
	Werden die <b>Testmethoden</b> entsprechend der priorisierten <b>Q-Merkmale</b> ausgewählt?		nein	mit Testfällen werden durchlaufende Prozesse beschrieben, Lasttest und Pentest geplant, aber separat nach Fertigentwicklung	0
	Erfolgt neben der Testplanung eine übergreifende QS-Planung in einem <b>Qualitätssicherungsplan</b> ?	X	nein	Konzepte gehen zur Abnahme an den Kunden, weitere Dokumente sind nicht vorhanden	0
	Wurde eine <b>Risikoanalyse</b> zur Bestimmung der kritischen Systemteile durchgeführt?		teilweise	aus Prozesssicht ja, nicht aus Systemsicht, Testplan durch Kunde definiert, Überschriften der Testfälle hierin vorgegeben, schlechte Testkoord. beim Kunden, Fehlerqualität schwankt von Rechtschreibfehler bis hin zu Prozessausfall	1
	Wird die Risikoanalyse in einem <b>Risikoplan</b> dokumentiert einschl. Eintrittswahrscheinlichkeit, Auswirkungsgrad und Präventiv- / Korrekturmaßnahmen?	X	nein	durch Flut an Anforderungen wurde immer nur gearbeitet und umgesetzt, Anforderungsmanagement ist aus dem Ruder gelaufen	0
	Sind für alle Anforderungen an das Softwaresystem <b>Abnahmekriterien</b> vorhanden?	X	nein	Testfälle sollen nun Abnahmekriterien sein	0
	Erfolgt eine <b>Test-Priorisierung</b> anhand der Risikoeinschätzung (Verteilung der Testaktivitäten auf Systemteile je nach Risikograd)?		nein	künftig, gesamter Verordnungsprozess Prio 1 = 80% der Funktionalitäten, die 156 Testfälle sind die 80%	0
	Auf welcher Basis werden <b>Testtiefe (Teststufe)</b> und <b>Testreihenfolge</b> abgeleitet?		Aus Erfahrungswerten	2 Stufen: Entwickler- und Funktionstest (nur intern) und Prozesstest (intern und Kunde)	1
	Werden pro Testprojekt <b>Testziele</b> definiert?	X	nein	ergibt sich aus Begrifflichkeiten	0
				Testfälle bilden gesamten Prozess ab, 30 - 40	

Projektabdeckungsgrad= (Erreichte Punktzahl / Maximal erzielbare Punktzahl) \* 100

Skalen-Einordnung:

< 50% failed      >=50% akzeptabel      >=75% gut      >=95% exzellent

Ergebnis **46**      Max Punkte **122**

Ergebnis in % **38%**

## Ableitung von Verbesserungspotentialen-TOP 6.

Finding	Verbesserungspotential
Keine Traceability zwischen Testfällen und Anforderungen	Mapping der Anforderungen zu den bereits abgestimmten Testfällen
Nur partielle Risikoanalyse zur Bestimmung der SW-System-kritischen Teile vorhanden	Bestimmung der Testfälle, die 80% der Funktionalitäten der Applikation ausmachen (= riskante Systemteile) → Testfall-Priorisierung
Fehler werden durch Kunde grundsätzlich nur den Fehlerklassen 1 und 2 zugeordnet	Aufsetzen eines internen Fehlerkorridors mit Abstimmung, ob auch für Kunde im Rahmen der Übernahme anwendbar



## Ableitung von Verbesserungspotentialen-TOP 6.

Finding	Verbesserungspotential
Keine mit dem Kunden gemeinsam definierten Abnahmekriterien	Einholen des Commitments beim Kunde: definierte TF-Beschreibungen = fixierten Anforderungen inkl. Abnahmekriterien
Zuordnung der Tester zu den Testfällen erfolgt momentan nach Verfügbarkeit	Um kritische Testfälle mit entsprechender Intensität zu testen, sind in einem Teststufenplan die Testfallreihenfolge, Testaufwand und -ressourcen zuzuordnen
Testaktivitäten durch neue Anforderungen / CRs seitens Kunde stark erschwert	Abstimmung Frozen Zone mit Kunde, in der keine Umsetzung neuer Anforderungen / CRs erfolgt

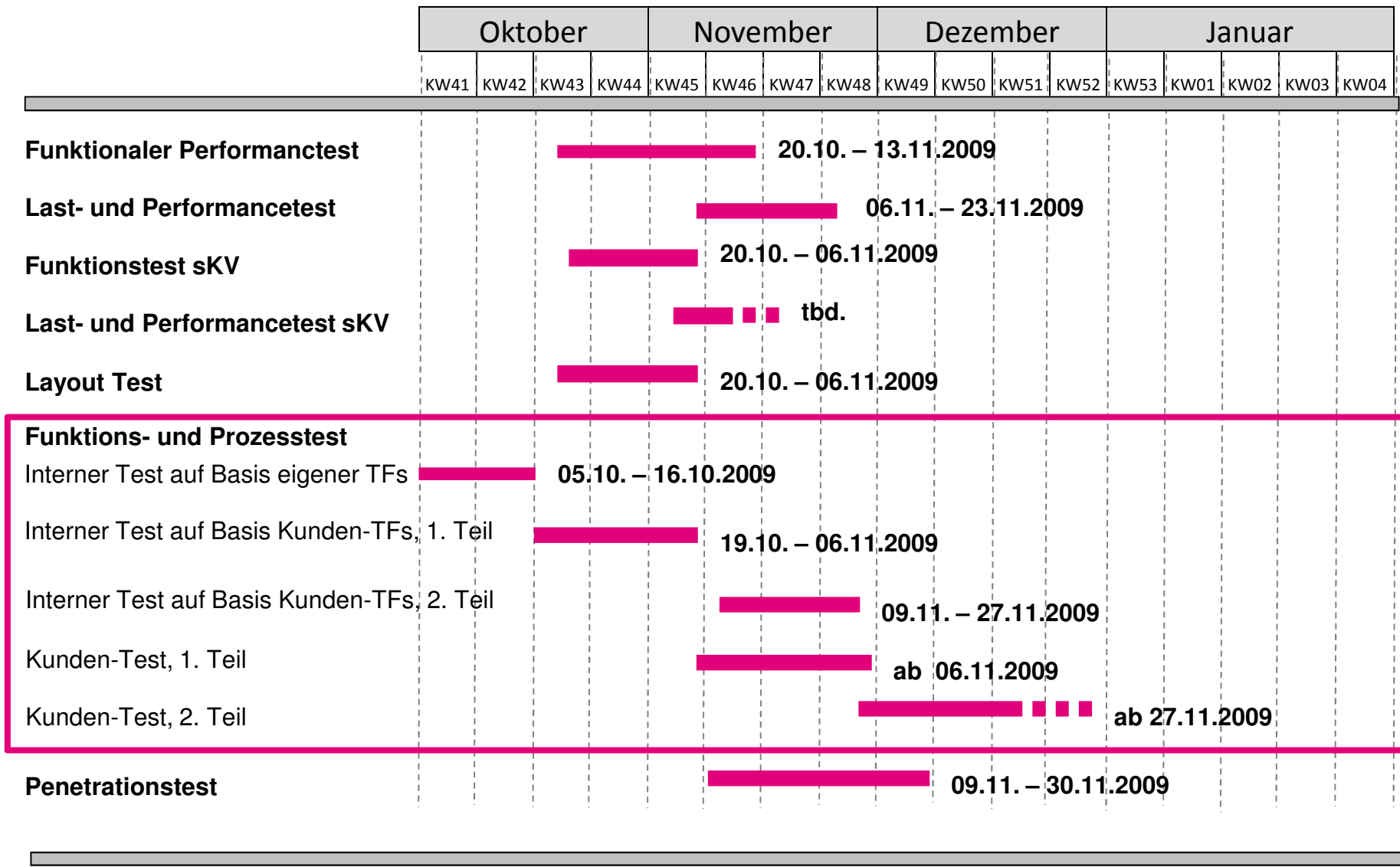
# Testkonzeption auf Basis der abgeleiteten Verbesserungspotentiale.

Mapping der Testfälle zu den grobgranular spezifizierten Anforderungen schwierig, deshalb:

- Kunde erstellt die Testfälle, die gleichzeitig die Abnahmekriterien darstellen inkl. deren Einstufung nach Priorisierungskategorien
- **1. Erkenntnis:** zum ersten Mal wurde auf Seiten des Kunden als auch SW-Herstellers die Wichtigkeit erkannt, formalisierte Testfälle zu erstellen
- Anfang Oktober 2009:
  - Auslieferung der ersten 20 Testfall-Beschreibungen durch den Kunden
  - Hochrechnung der benötigten Zeiten für weitere Testfall-Erstellung

⇒ Grundlage für die Erstellung eines belastbaren Teststufenplans





Testkoordination und -monitoring.

.. T .. Systems ..

# Ergebnisse und Konflikte.

SW-Hersteller interner Test:

- Positives Ergebnis für alle Testphasen
- Die zum Testzeitpunkt zur Verfügung stehenden Testfälle wurden fehlerfrei durchlaufen
- Alle identifizierten Fehler wurden behoben und erfolgreich nachgetestet: lediglich 1 nicht Pilot-verhindernder Fehler (FK 4-5) war nicht behoben, erfolgte im gemeinsamen Regressionstest.
- Einhaltung des intern definierten Fehlerkorridors (0/0/9/15/-)
- Konflikt: zurück gestellte Testfälle aufgrund fehlender bzw. fehlerhafter Testfallbeschreibungen.

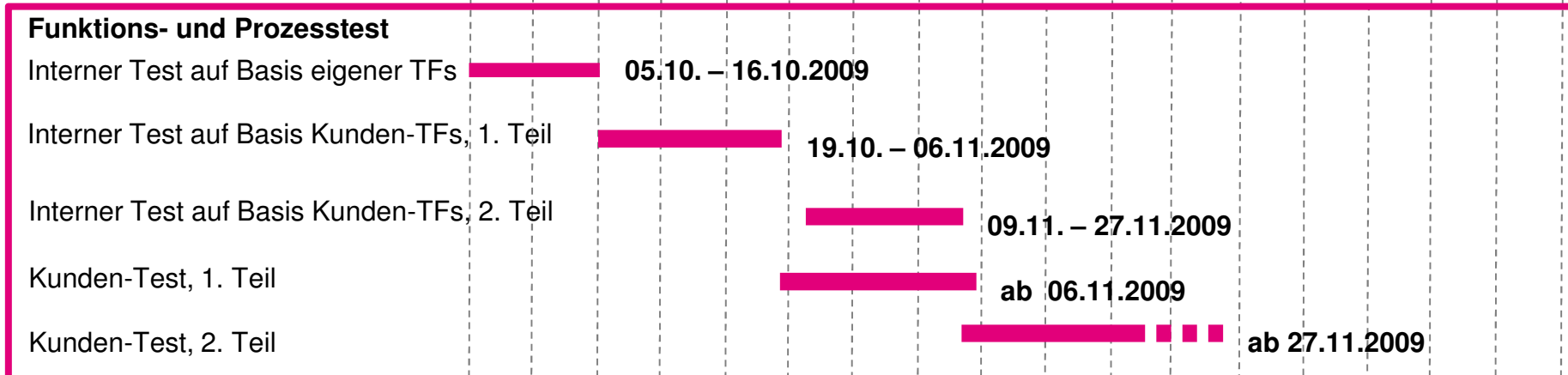
# Ergebnisse und Konflikte.

- Hohes Fehleraufkommen beim anschließenden Kunden-Test:
  - bisherige Testergebnisse wurden angezweifelt
  - aber auch kundenseitige Fehler beim Testvorgehen
  - während des Tests wird erkannt, dass TF-Korrekturen erforderlich sind
- ⇒ Entscheidung zum gemeinsamen Test mit einem gemischten Team aus Testern von der Kunden- als SW-Hersteller-Seite
- ⇒ Entscheidung zu einem abgestimmten Prozess hinsichtlich TF-Anpassungen

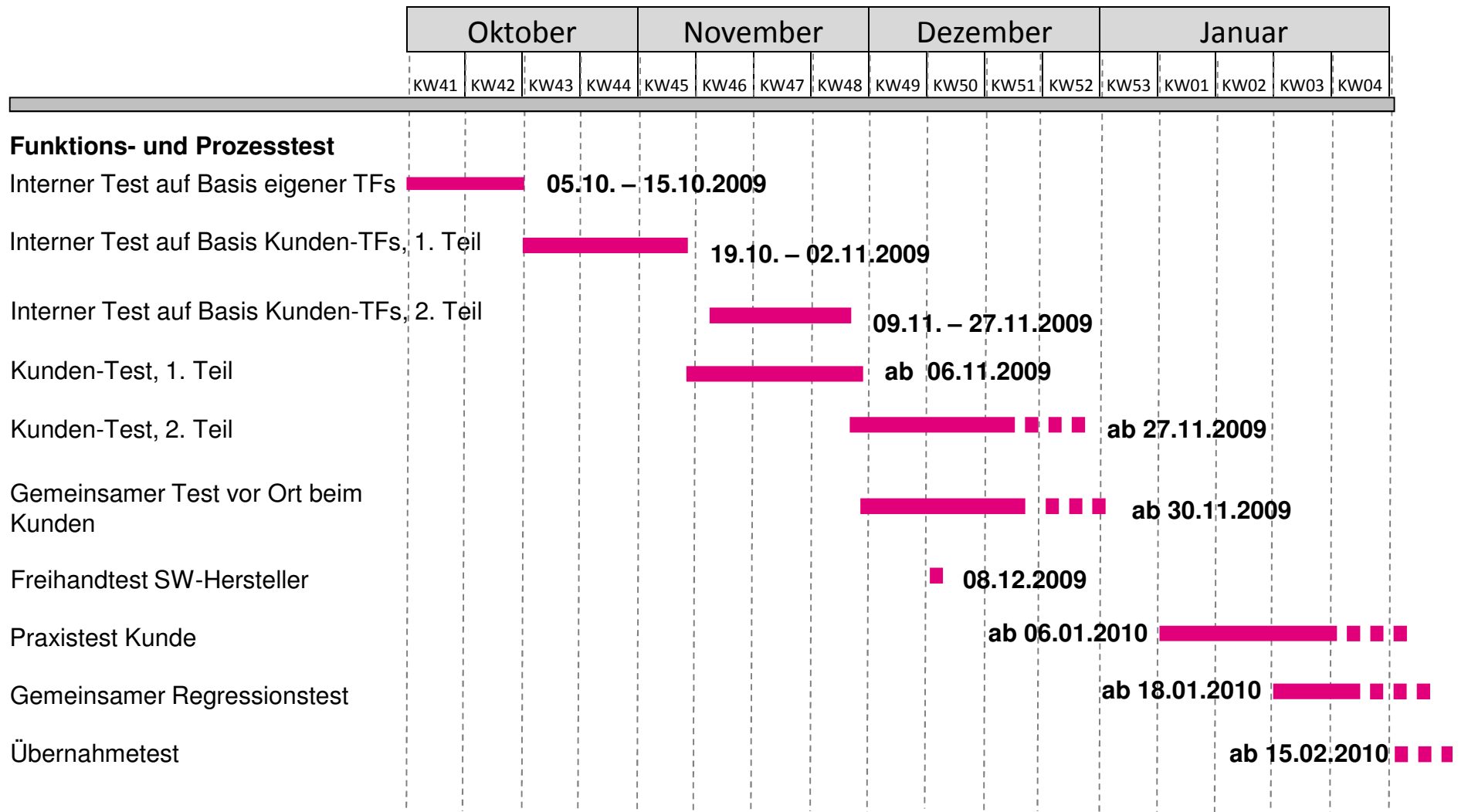
... **T** ... **Systems** ...

Oktober				November				Dezember				Januar				
KW41	KW42	KW43	KW44	KW45	KW46	KW47	KW48	KW49	KW50	KW51	KW52	KW53	KW01	KW02	KW03	KW04

## Zur Erinnerung: ursprüngliche Testplanung



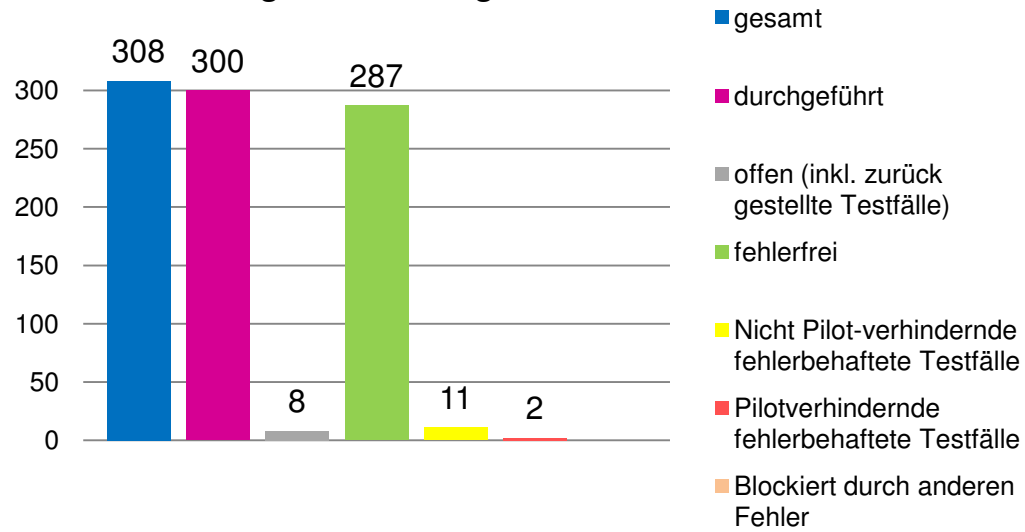




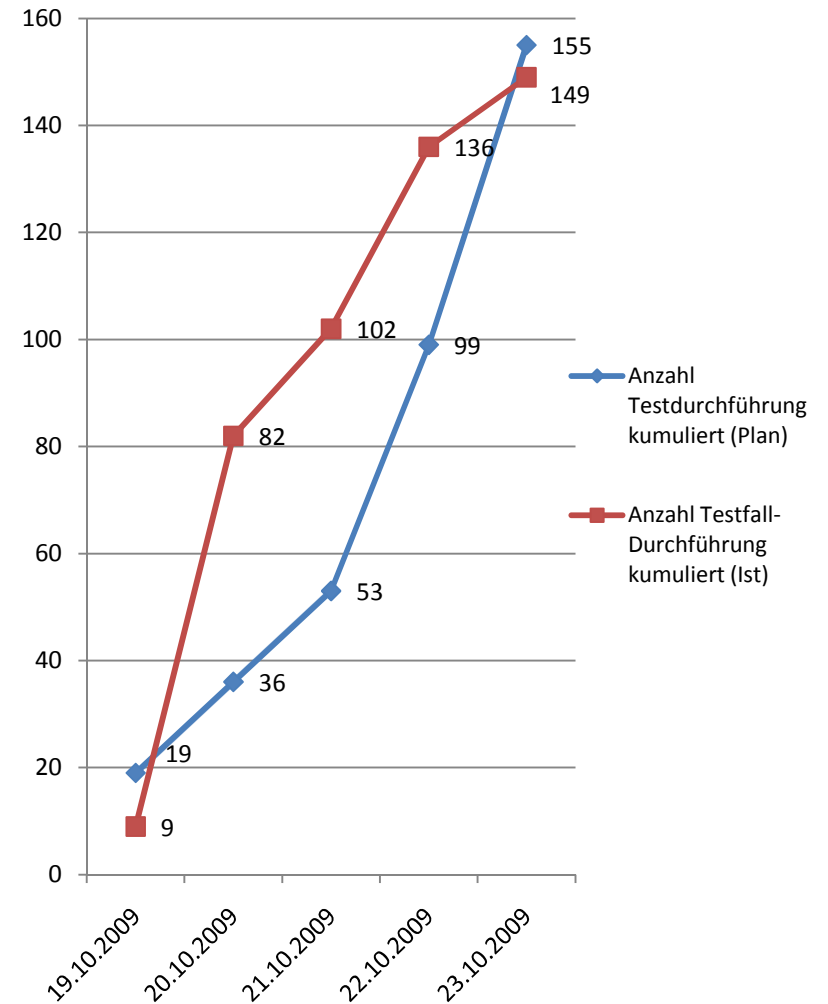
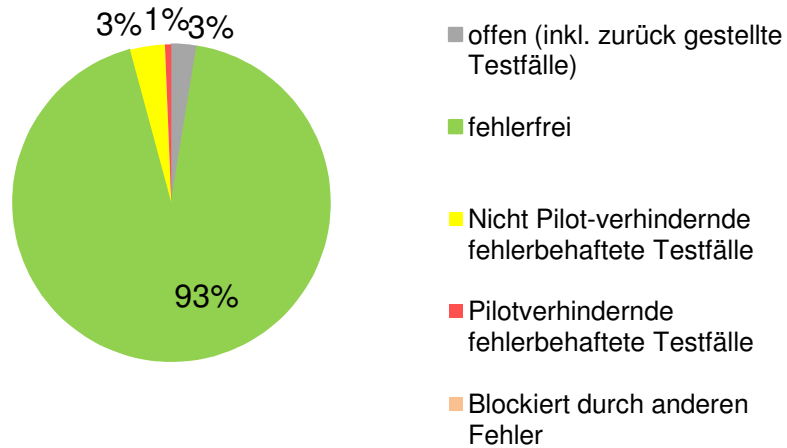
**Redaktionsschluss Trial-System: 11.03.2010, Schwenk zum Produktivsystem**

Bearbeiter: Seidel		Version: 1.0		Detaillierter Plan und Status							
Regressionstest		Datum: 21.01.2010									
TestszENARIO	Testfall-Name	Ausprägung	Test-Durchführung					Nachtest			
			Kategorie	Kommentar	Test-Datum	Status Testdurchführung	Fehler-ID in Trac	Tester	Datum Nachtest	Status Nachtest	Tester Nachtest
13	Suche / Sortierung	g2)	1		21.01.2010	fehlerfrei		Team 4			
13	Suche / Sortierung	g3)	1		27.01.2010	fehlerfrei mit leichten Mängel	398	Team 1		offen	
13	Suche / Sortierung	g4)	1		21.01.2010	fehlerfrei		Team 2			
13	Suche / Sortierung	g5)	1	Aufgrund Testdaten nicht 100% prüfbar, da nicht mit Echtdaten kompatibel	20.01.2010	fehlerfrei		Team 3			
13	Suche / Sortierung	h) LE-Suche (mit Parametern)	1		21.01.2010	fehlerfrei mit leichten Mängel	357	Team 4	09.02.2010	offen	Team 4
13	Suche / Sortierung	i) HM-Nummernsuche	1		27.01.2010	fehlerfrei		Team 1			
13	Suche / Sortierung	j) PZN-Suche	1		21.01.2010	fehlerfrei	392	Team 2	01.02.2010	erfolgreich nachgetestet (für Pilot)	Team 2
13	Suche / Sortierung	k) Vertragspreissuche	1		21.01.2010	fehlerfrei	392	Team 3	01.02.2010	erfolgreich nachgetestet (für Pilot)	Team 3
14	Druck	a1) Anlage drucken	1		21.01.2010	fehlerfrei mit leichten Mängel	119	Team 4		offen	
14	Druck	a2)	1		27.01.2010	fehlerfrei		Team 1			
14	Druck	a3)	1		21.01.2010	fehlerfrei		Team 2			
14	Druck	b) Vorgangsdruk	1	an Hr. .... weitergeleitet wegen Parkgrund	20.01.2010	fehlerfrei		Team 3	25.01.2010	erfolgreich nachgetestet (für Pilot)	Team 3
14	Druck	c) Dokumentendruk	1	Dokumentenlayout	04.02.2010	fehlerhaft mit schweren Fehlern	471 (311)	Team 4	09.02.2010	offen (für Pilot)	Team 4
14	Druck	d) Genehmigungsdruck	1		21.01.2010	fehlerfrei		Team 1			
14	Druck	e) Genehmigungsdruck mit Betreuer	1		21.01.2010	fehlerfrei		Team 2			
14	Druck	f) Genehmigungsdruck für unter 15jährige	1		20.01.2010	fehlerfrei		Team 3			
14	Druck	g) Übersichtsdruck	2		21.01.2010	fehlerfrei		Team 4			
15	Schnittstellen	a) DDG Genehmigungsdatensatz Dauerverordnung	1		21.01.2010 03.02.2010	fehlerfrei	408	Team 1			
15	Schnittstellen	b) DDG Genehmigungsdatensatz Fallpauschale	1		21.01.2010	fehlerfrei		Team 2			

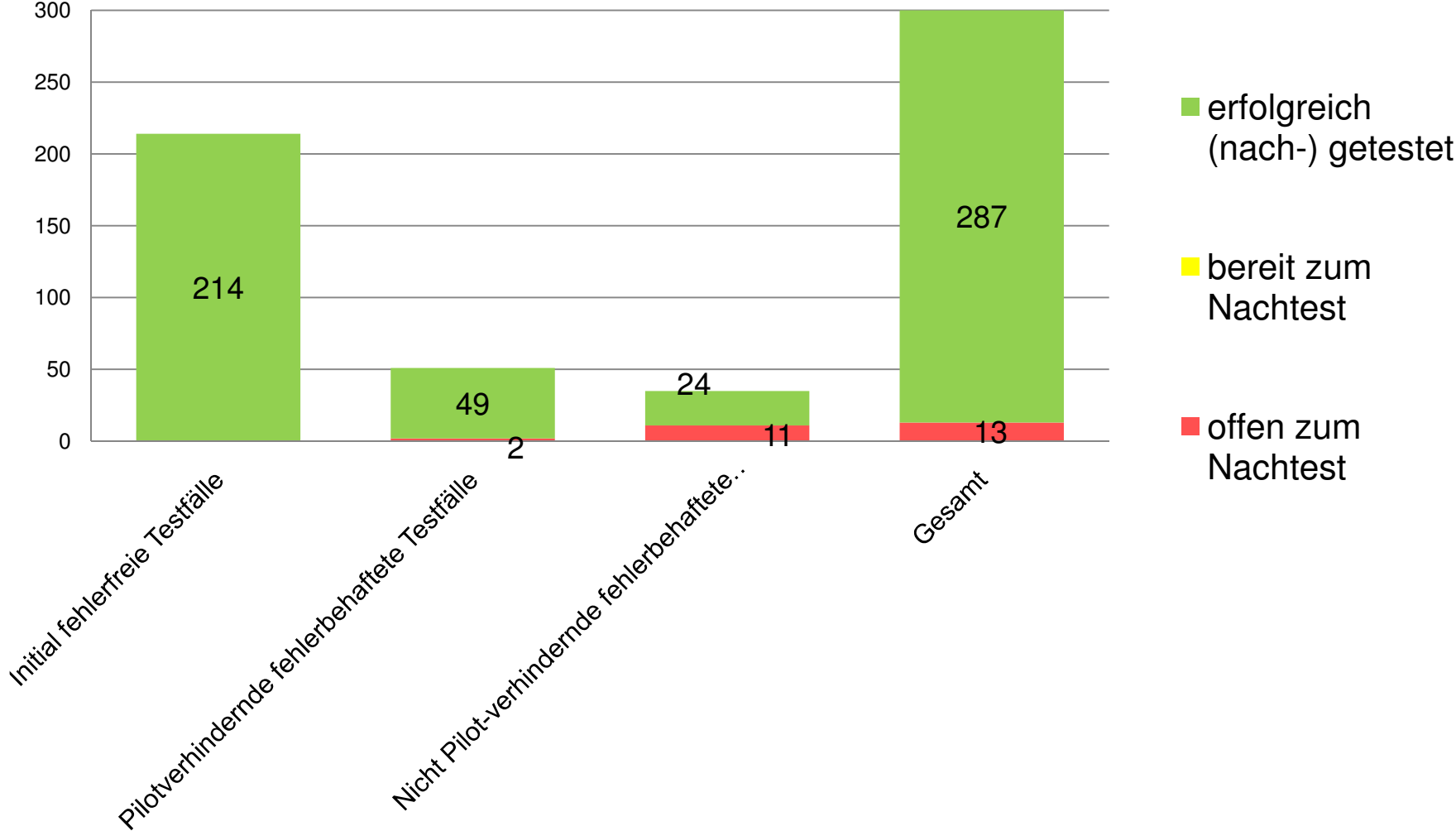
## Testdurchführung - Abarbeitung Testfälle



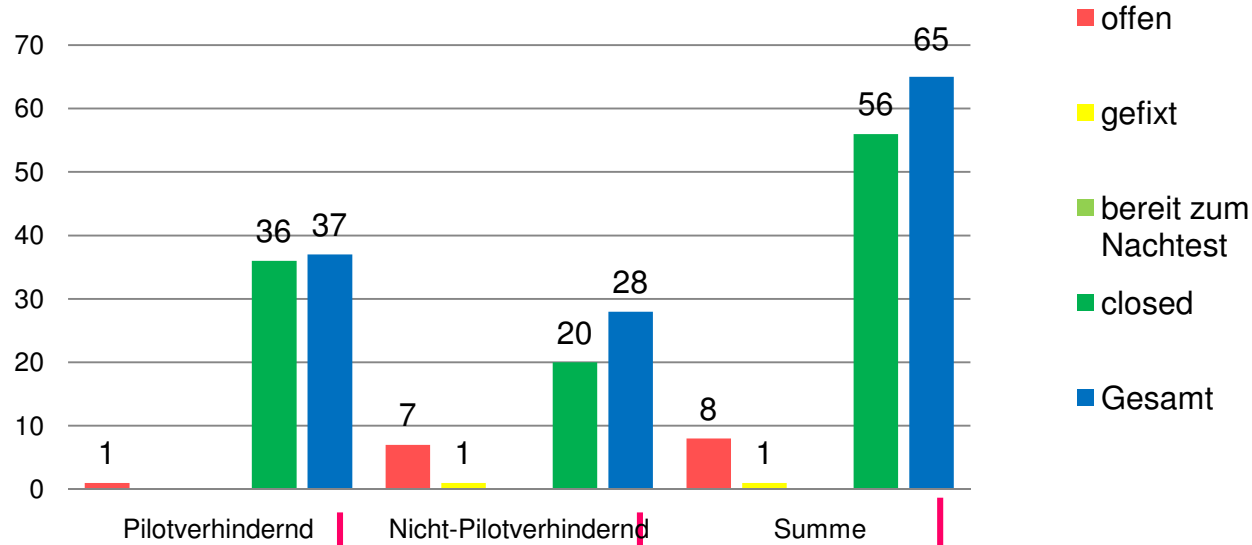
## Testdurchführung insgesamt



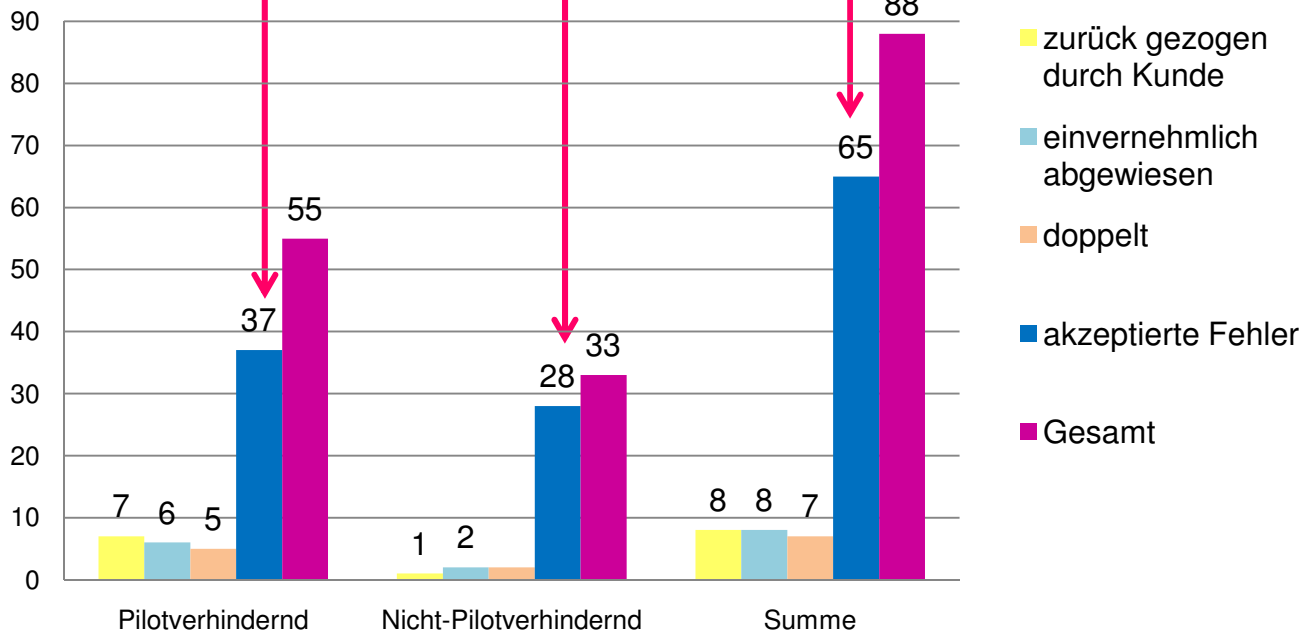
Aufteilung Testfall-Status nach Fehlerfreiheit



### Fehlerstatus akzeptierte Fehler nach Fehlerschwere



### Fehlerstatus Meldungen gesamt



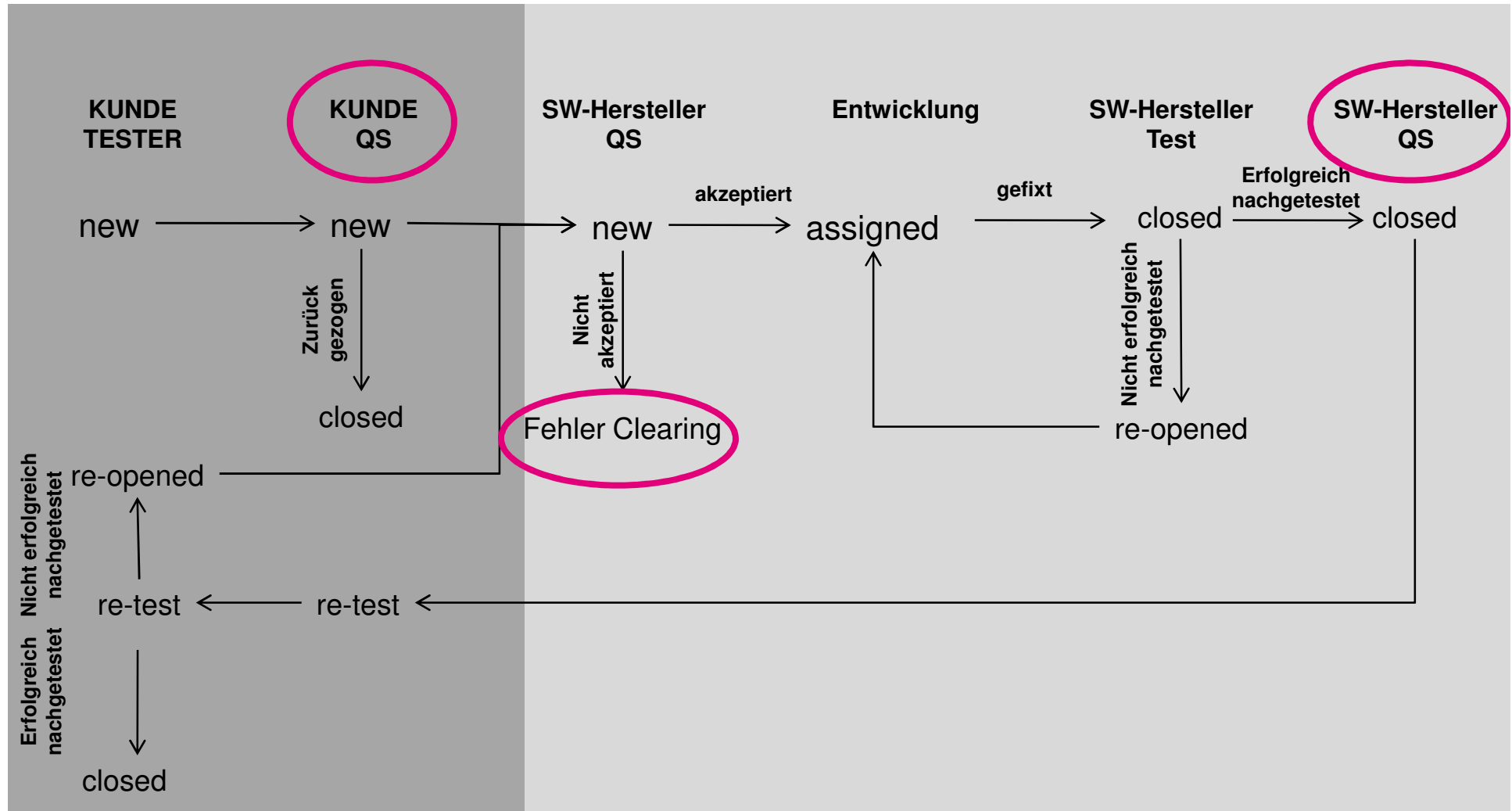
Fehlermanagement.

.. T .. Systems ..

# Fehlermanagement – Allgemeiner Workflow.

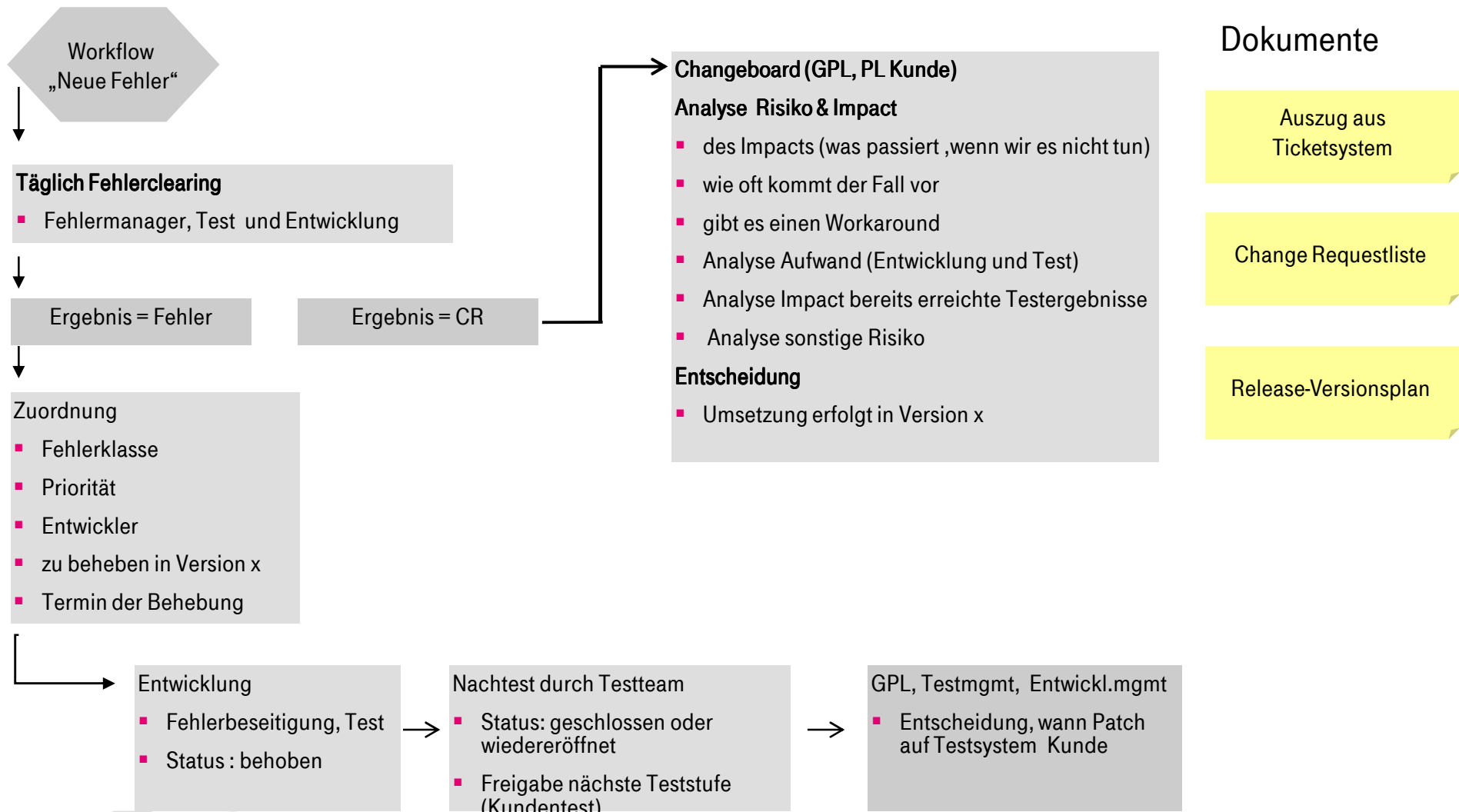
- **Bislang:**
  - Fehler wurden unbewertet auf Zuruf in internes Ticket-System des SW-Herstellers eingestellt und unkoordiniert abgearbeitet
- **Nun:**
  - Alle festgestellten Fehler, Abweichungen, Probleme, usw. werden unmittelbar im Fehlermanagementsystem (TRAC) als „Fehler“ erfasst und dokumentiert sowohl durch den Kunden als auch SW-Hersteller
  - Bewertung Fehler-Klassifizierung und Behebungstermin mittels eines „Fehler Clearings“
  - Vorab-Bewertung durch Entwicklungsteam des SW-Herstellers
  - Tägliches Fehler-Reporting an den Kunden

# Fehlermanagement – Allgemeiner Workflow.





# Fehlerprozess – Klassifizierung und Priorisierung.

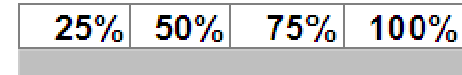


## Dokumente

- Auszug aus Ticketsystem
- Change Requestliste
- Release-Versionsplan

# HOTSPOT Teilprojekt Test-Fehlermanagement

Fertigstellungsgrad



### Status & Results

- Übernahmetest bei finalisiert: 502 von 564 Testfällen durchgeführt: 496 Testfälle fehlerfrei (= 89% aller 564 Testfälle, 99% alle getesteten 502 Testfälle), keine Testfälle mit Prio 1-Fehlern, alle Testfälle der Kategorie 1 wurden getestet (162/163 fehlerfrei)
- Schulungssystem: keine Tickets im Status „offen“
- Regressionstest: keine offenen Pilot-verhindernden Tickets, 5 nicht Pilot-verhindernde Tickets mit Status „offen“ (Prio 4 und 5)
- Praxistest: 1 Pilot-verhinderndes Ticket in Klärung (Druckerschacht-Ansteuerung, wird im Pilotbetrieb über Workaround betrieben), 6 nicht Pilot-verhindernde Tickets mit Status „offen“ (Prio 4 und 5)
- Gemeinsamer Test (1. gemeinsame Testphase): keine offenen Showstopper, 3 offene Tickets mit Prio 4 und 5

### Problems & Risks

- Loadbalancing-Problem im RZ ...
- -> gelöst! Durch Umkonfiguration des Loadbalancers und Änderungen am Cookie Handling tritt das Problem derzeit nicht mehr auf.
- Noch offene, umzusetzende Punkte beim PEN-Test.

### Next Milestones

	Plan	Actual
• Lasttest	Ende Februar	März
• Nachtest der noch offenen Punkte aus dem Pen-Test (ongoing)	nächste Woche	
• Start Pilotbetrieb	15.03.10	

### Customer View

- Übernahmetest seit 15.02.10 beim Kunden vor Ort, dieser wurde am 11.03.10 finalisiert
- Offene Nachtests und offene Testfälle aus Übernahmetest werden im Pilotbetrieb realisiert
- Redaktionsschluss Trial-System am 11.03. mittags, danach erfolgt auf diesem System der Schwenk zum Produktivsystem
- Kunde sieht dem Pilotstart aufgrund der positiven Übernahme-Testergebnisse (alle mit Prio 1-Tickets versehenen Testfälle erfolgreich nachgetestet, keine offenen Prio 1-Tickets) positiv entgegen

### Actions, deadline

- Festlegung des Gremiums für LE-Anforderungen
- Bzgl. Loadbalancer-Problematik werden zwei Lösungswege verfolgt:
  - 1.) Behebung der Probleme auf dem derzeitigen Loadbalancer:  
akt. Stand: Durch Konfigurationsänderungen auf dem Loadbalancer treten keine Probleme mehr beim Zugriff aus dem Internet auf. Lediglich beim Zugriff aus dem Internet kommt es noch zu Session-Problemen. An der Behebung wird gearbeitet.
  - 2.) Es wird auf einen F5 Loadbalancer gewechselt. Dieser ist bereits bestellt. Wann dieser verfügbar ist, ist derzeit noch unklar.
- > Siehe „Problems & Risks“, Problem tritt derzeit nicht mehr auf.

### Comments



Fazit.

.. T .. Systems ..

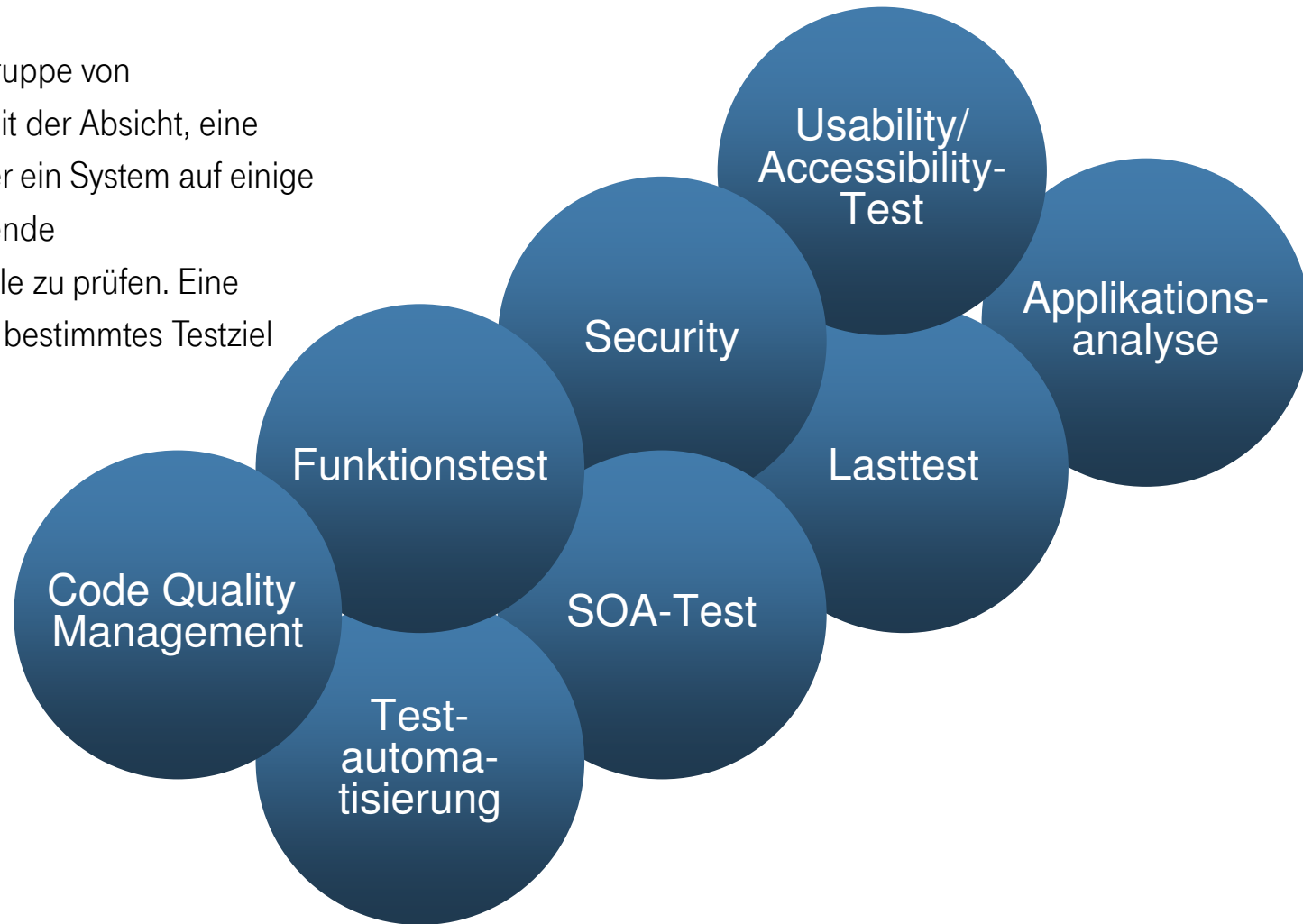
## Fazit.

- Die saubere Aufnahme und Abstimmung von Anforderungen zusammen mit dem Kunden ist das A und O für erfolgreiche Projekte.
- Die tägliche Kommunikation im Projektteam als auch mit dem Kunden hat die nötige Transparenz bzgl. notwendiger Maßnahmen sowie zum Status geschaffen. Das schaffte vor Allem Vertrauen auf Kundenseite.
- Aktives Fehler Clearing zusammen mit dem Kunden führte zur zügigen Umsetzung / Behebung der aus seiner Sicht wichtigsten Pain Points.
- Der gemeinsame Test unter Beteiligung des SW-Herstellers und des Kunden, konnte frühzeitig Anwendungsfehler aufdecken, um „falsche“ Fehler zu verhindern.
- Die Anwendung befindet sich heute seit fast einem halben Jahr im Wirkbetrieb.

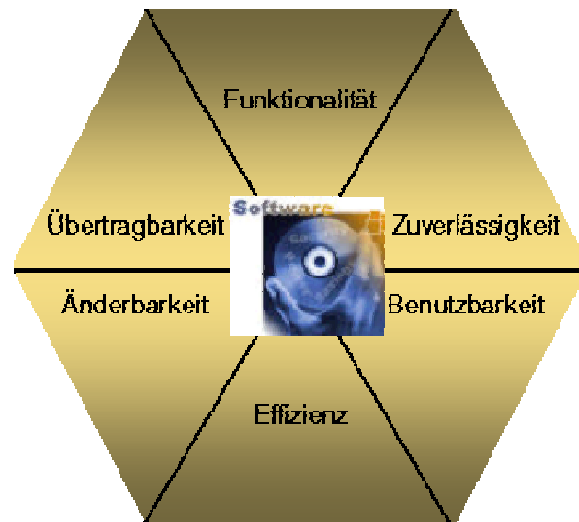
# 7. Überblick ausgewählter Testarten.

# Testarten im Überblick.

**Testart** = Eine Gruppe von Testaktivitäten, mit der Absicht, eine Komponente oder ein System auf einige zusammenhängende Qualitätsmerkmale zu prüfen. Eine Testart ist auf ein bestimmtes Testziel fokussiert.



# Qualitätsmerkmale.



Systematik entspricht den DIN-ISO-Normen 9126 und 12119.

Qualitätsmerkmale von Softwaresystemen:

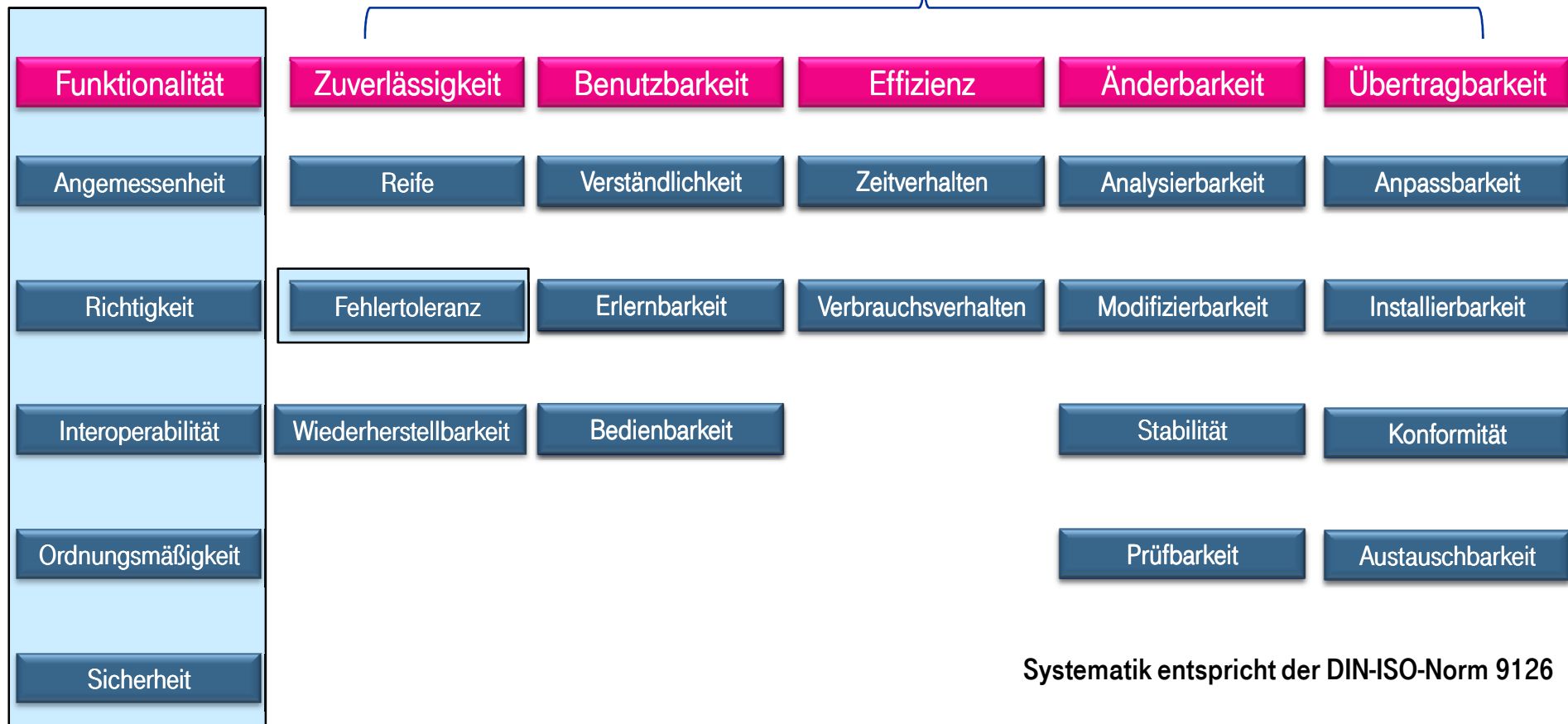
- Funktionsumfang – kann die Software alle Aufgaben erfüllen?
- Zuverlässigkeit – wie reagiert das System auf die Fehleingaben?
- Benutzbarkeit – ist das System verständlich und leicht bedienbar?
- Effizienz – wie sind die Antwortzeiten ?
- Änderbarkeit – kann die Software erweitert werden?
- Übertragbarkeit – läuft die Software unter verschiedenen Betriebssystemen oder Browsern?

Funktionstest.



# QUALITÄTSMERKMALE EINES SOFTWARESYSTEMS.

## Nicht-funktionale Q-Merkmale



Systematik entspricht der DIN-ISO-Norm 9126

# FUNKTIONSTEST.

## Motivation

## Vorgehen

- Tut die Anwendung genau das, was in den funktionalen Anforderungen beschrieben wurde?
- Wo gibt es Abweichungen und welchen Auswirkungsgrad besitzen diese?
- Was muss zuerst/am intensivsten getestet werden, wo sind einfachere Test ausreichend?
- Wie wird die Wirtschaftlichkeit des Tests sichergestellt?
- Wie gut ist die Qualität der Entwicklung?

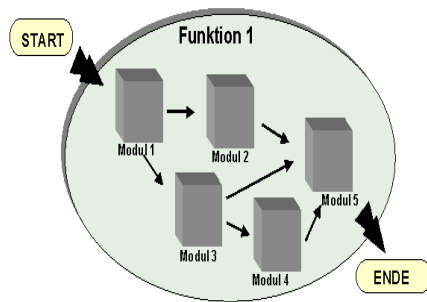


# FUNKTIONSTEST.

Motivation

Vorgehen

Testelemente  
abgrenzen



Priorisieren

- Komplexität
- Häufigkeit
- Schätzung durch AG
- Schätzung durch Entwicklung

Testmethode

- Äquivalenzklassen
- Grenzwertanalyse
- Zustandsbasiert
- Intuitiv
  
- Manuell
- Automatisiert
- SOA

Testspezifikation

Lasttest.

# LASTTEST.

## Motivation

## Vorgehen

## Nutzen

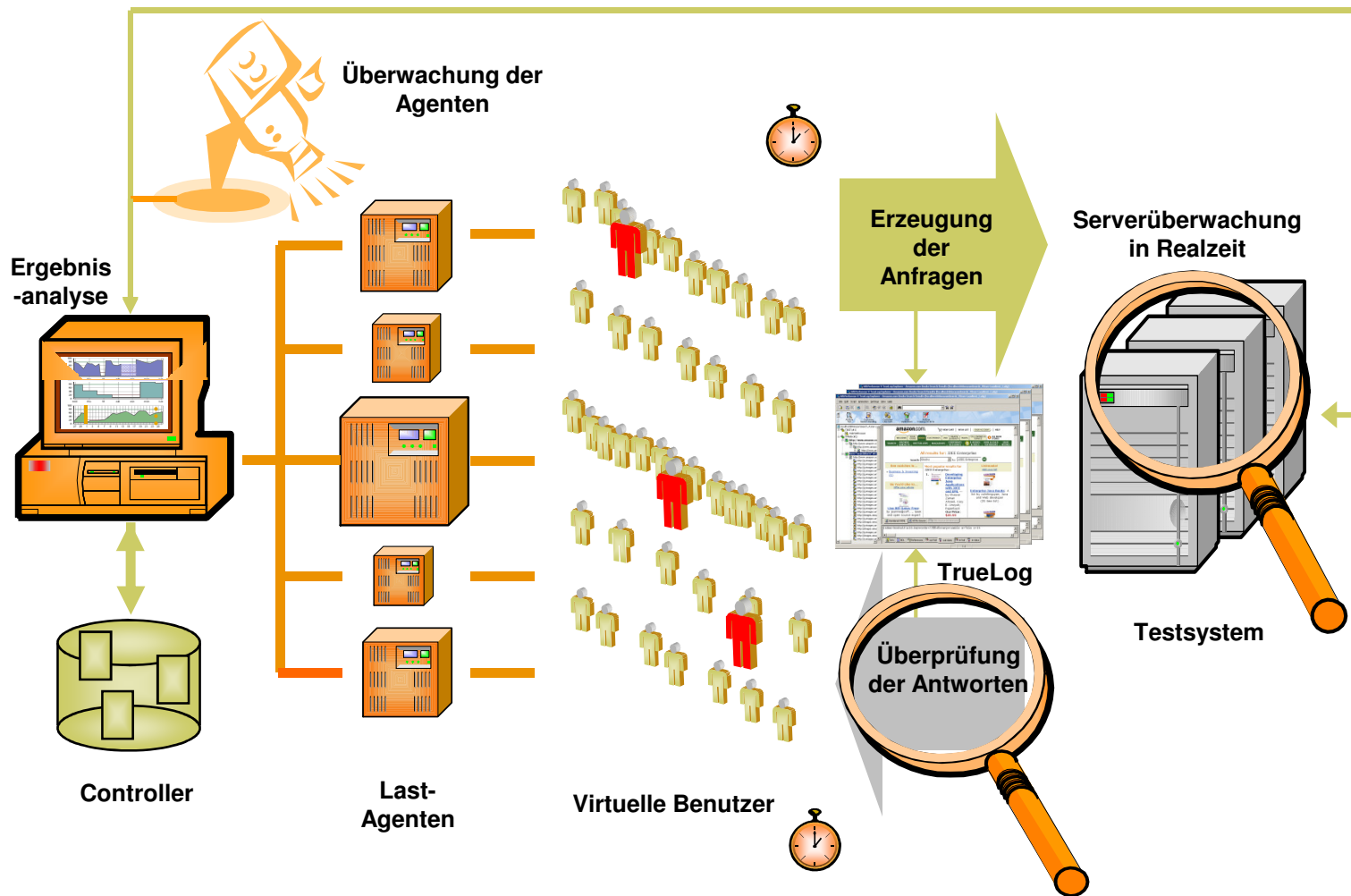
### Warum Lasttest?

- Steigende Nutzer- und Zugriffszahlen können längere Antwortzeiten, Fehler und Abstürze provozieren
- Wirtschaftliche Schäden durch Ausfall und Störungen entstehen
- Imageverlust durch schlechte Performance
- 8 Sekunden Regel - Kunde macht sein Geschäft woanders!

### Beantwortung folgender Fragen:

- Wie viele User verkraftet die Anwendung?
- Wo liegen Performance-Engpässe unter Last?
- Wie lange müssen die Kunden bei steigender User-Anzahl warten?
- Wie sind meine Server ausgelastet?
- Wann muss die Hardware erweitert werden?

# LASTTEST.



# Usability/Accessibility-Test.

# USABILITY/ACCESSIBILITY-TEST.

## Motivation

Vorgehen

Nutzen

- Wie gut kommen die Nutzer mit der Anwendung zu recht?
- Spricht die Anwendung die Sprache der Nutzer?
- Wo sind kritische Punkte bei der Nutzung?



# USABILITY.

Motivation

**Vorgehen**

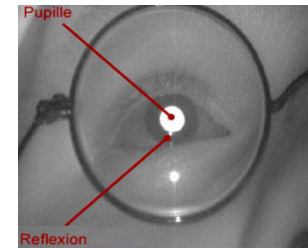
Nutzen

## Usability-Experten-Evaluation

- Umfassende Evaluation der Nutzungsoberfläche durch mindestens 2 Experten
- Usability-Experten versetzen sich in Rolle repräsentativer Nutzer
- Überprüfen auf Basis anerkannter Heuristiken
- Ausführliche Erläuterung aller identifizierten Usability-Schwachstellen
- Konkrete und praxisnahe Verbesserungsvorschläge zu allen gefundenen Schwachstellen
- 30- bis 40-seitiger Testbericht

## Usability-Test mit Probanden

- Nutzertest durch 6-8 Probanden, welche typische Aufgaben am zu testenden System durchführen
- Eye-Tracking Analyse einer Schlüsselseite durch einen repräsentativen Probanden
- Ausführliche Erläuterung aller identifizierten Usability-Schwachstellen
- Konkrete und praxisnahe Verbesserungsvorschläge zu allen gefundenen Usability-Schwachstellen
- 30- bis 40-seitiger Testbericht inkl. ausgesuchter Kommentare der Probanden
- Highlight-Video mit ausgewählten Szenen der Probanden



# USABILITY/ACCESSIBILITY-TEST.

Motivation

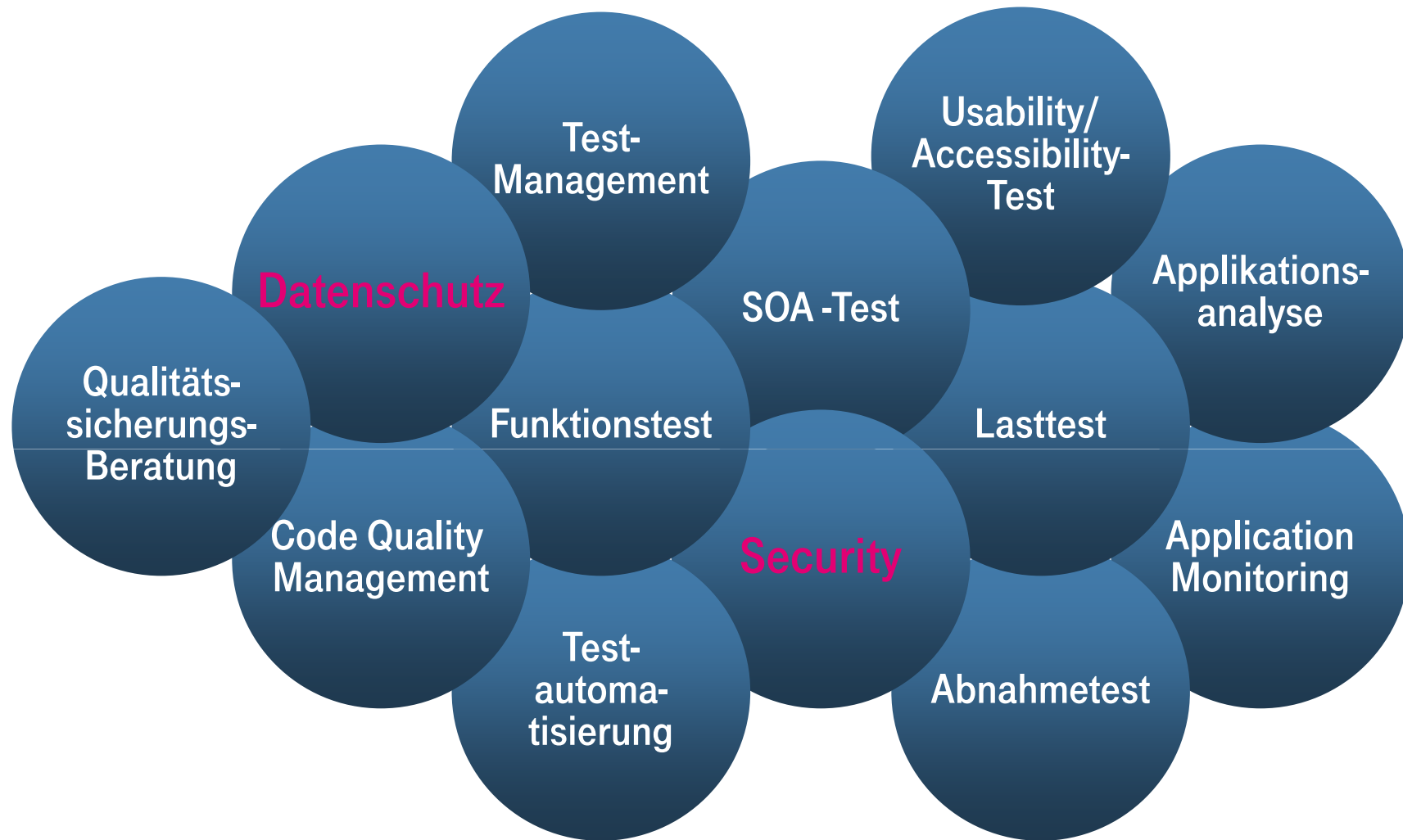
Vorgehen

Nutzen

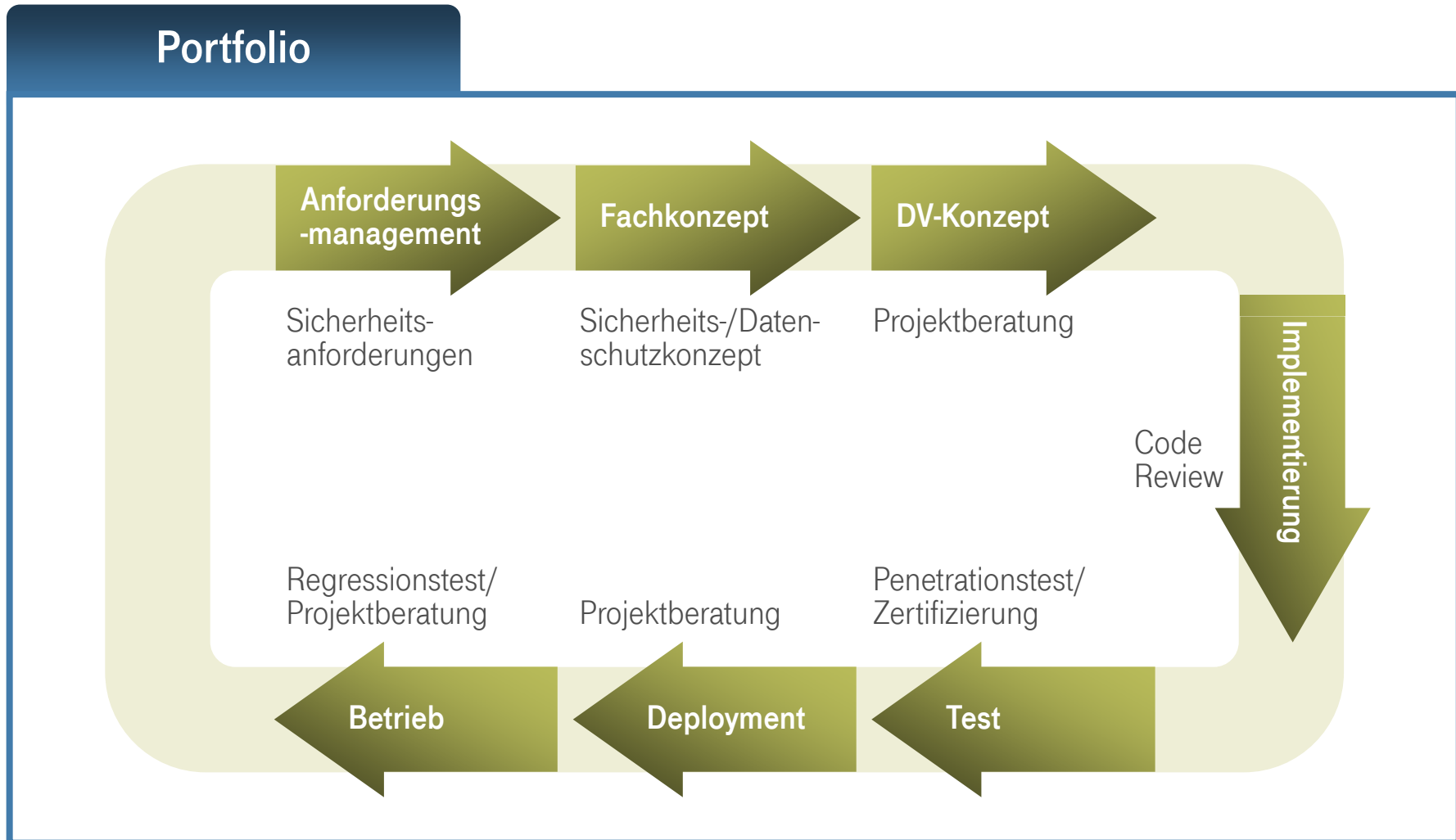
The screenshot displays a usability testing software interface. On the left, there is a project tree for 'BA Usability-Test' with a list of markers and segments. Below it, a 'Details' panel shows the current marker's definition and time. A video feed at the bottom left shows a user's face. The main window shows a recording of a user's interaction with a website, with a timeline at the bottom. A search results table is visible at the bottom right.

Elapsed	Event	Name	Notes	Creator	Marker Type	Definition	Created In	Time/Date
0:01:09.07	Marker	InfoSubs...		man	S	Start task	Planrole V1...	11:51:52 / ..
0:02:29.81	Marker	ms, help C...		man	Q	Quote, User comment	Planrole V1...	11:52:23 / ..
0:03:09.99	Marker	Marker		gach	E	End task	Manager	09:04:42 / ..

Und noch viel mehr...



# Data Privacy and Security

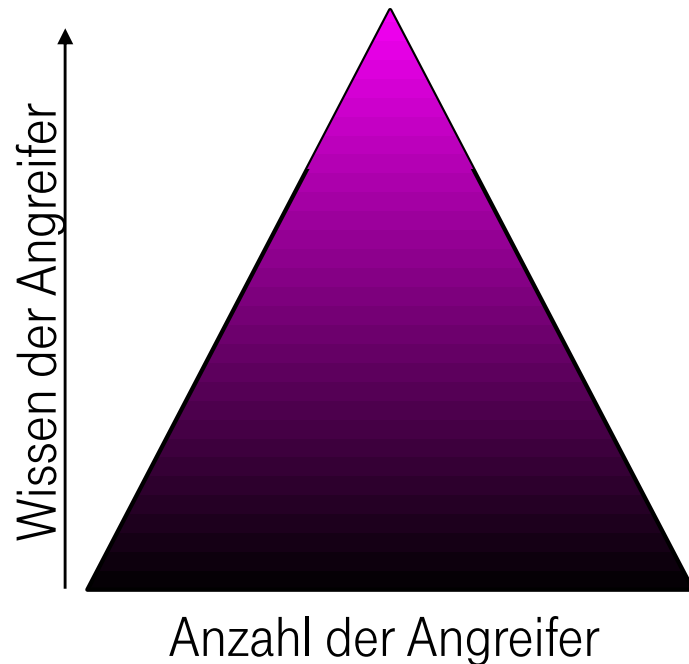


# 8. Data Privacy and Security.

Umfeld.

# Data Privacy and Security

## Wissen der Angreifer



- Weitergabe von Informationen
- Schnelle Verbreitung
- Verstärkter Einsatz von Werkzeugen
- Notwendiges Wissen zum Ausnutzen von Schwachstellen sinkt
- Anzahl der möglichen „Angreifer“ steigt

# Data Privacy and Security

## Formen von Bedrohungen

Neue Angriffsformen mit alten Hilfsmitteln

Häufige Angriffsformen, die „nur“ einen Browser benötigen:

- Cross Site Scripting
- {SQL, Command, HTML, PHP, ...} – Injection
- Informationspreisgabe

Weitere erfolgreiche Angriffsformen:

- Social Engineering



# Data Privacy and Security

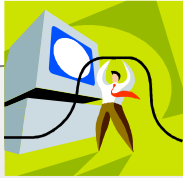
## OWASP Top 10

OWASP Top 10 – 2007 (Previous)	OWASP Top 10 – 2010 (New)
A2 – Injection Flaws	A1 – Injection
A1 – Cross Site Scripting (XSS)	A2 – Cross Site Scripting (XSS)
A7 – Broken Authentication and Session Management	A3 – Broken Authentication and Session Management
A4 – Insecure Direct Object Reference	A4 – Insecure Direct Object References
A5 – Cross Site Request Forgery (CSRF)	A5 – Cross Site Request Forgery (CSRF)
<was T10 2004 A10 – Insecure Configuration Management>	A6 – Security Misconfiguration (NEW)
A10 – Failure to Restrict URL Access	A7 – Failure to Restrict URL Access
<not in T10 2007>	A8 – Unvalidated Redirects and Forwards (NEW)
A8 – Insecure Cryptographic Storage	A9 – Insecure Cryptographic Storage
A9 – Insecure Communications	A10 - Insufficient Transport Layer Protection
A3 – Malicious File Execution	<dropped from T10 2010>
A6 – Information Leakage and Improper Error Handling	<dropped from T10 2010>

# Data Privacy and Security.

Arten von Schwachstellen.

# Data Privacy and Security



## Nutzerseitige Angriffe

- Cross Site Scripting
- Cross Site Tracing
- Inhaltsmanipulation



## Authentifizierung

- Brute Force
- Unzureichende Authentisierung
- Schwache Passwort Wiederherstellung
- Cross Site Authentication



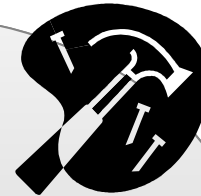
## Autorisierung

- Berechtigungen vorhersagen
- Unzureichende Autorisierung
- Unzureichende Sitzungsbeendigung
- Sitzungs Fixierung



## Informationsoffenlegung

- Auflisten Verzeichnis
- Informationsleck
- Pfadmanipulation
- Vorhersagbare Quellen



## Befehlsausführung

- Injection Angriffe (Einschleusung)
- SQL Injection



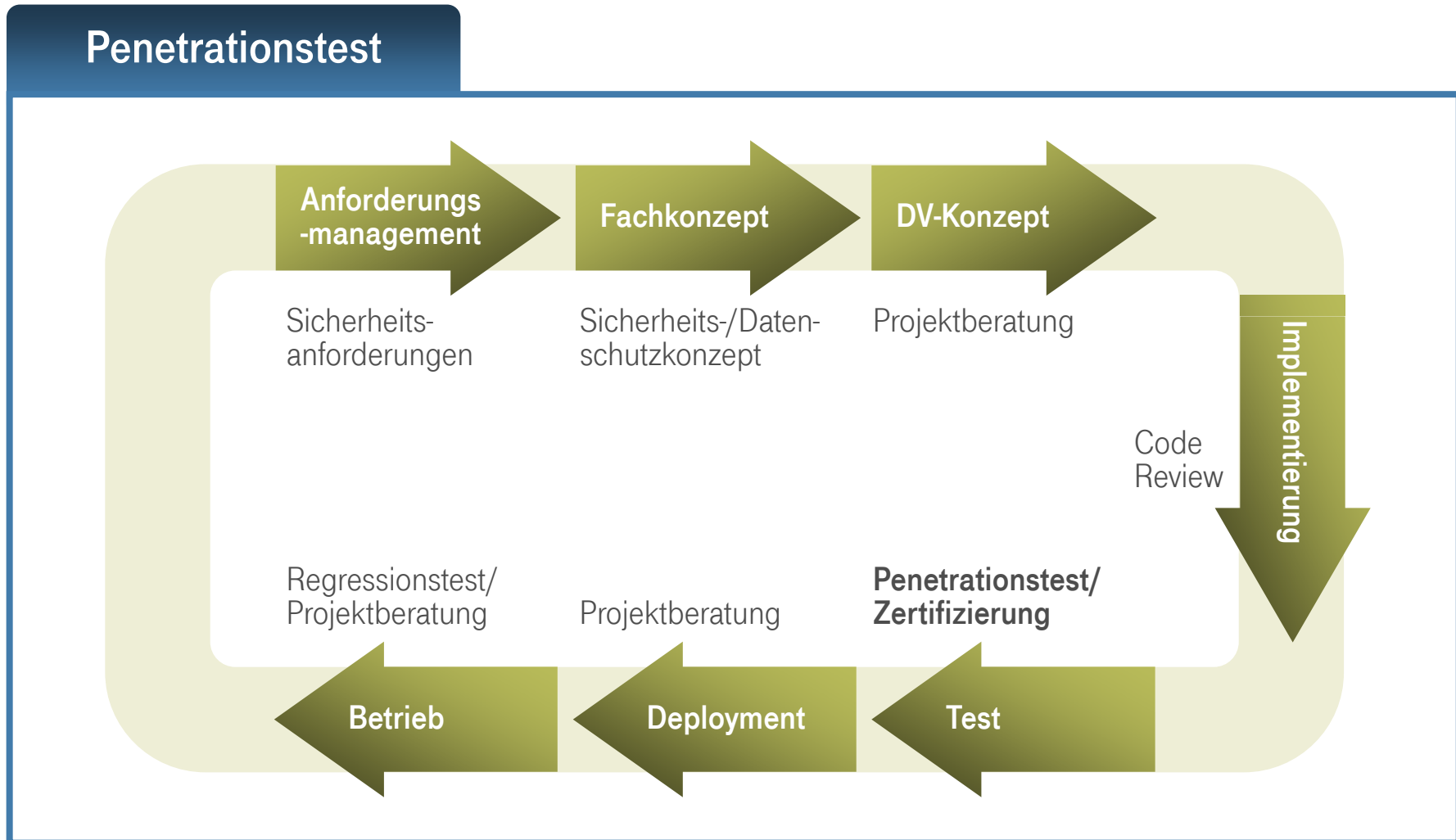
## Logische Angriffe

- Unzureichende Anti-Automation
- Unzureichende Prozessprüfung
- Cross Site Request Forgery

# Data Privacy and Security.

Durchführung Penetrationstest.

# Data Privacy and Security



# Data Privacy and Security

## Penetrationstest

## Vorgehen

## Praktische Beispiele

- Anforderungsmanagement, Projektberatung.
  - Anforderungen an Betrieb, Entwicklung und Fachseite wurden definiert.
- Fachkonzepte.
  - Sicherheitskonzept wird erstellt (gilt als Testgrundlage).
  - Datenschutzkonzept wird erstellt.
- **Penetrationstest.**
  - **Überprüfung der Wirksamkeit der Maßnahmen aus dem Sicherheitskonzept.**

# Data Privacy and Security

Penetrationstest

**Vorgehen**

Praktische Beispiele

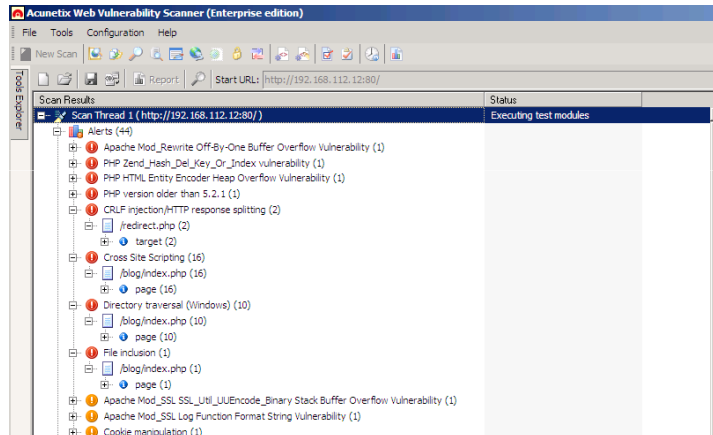
- Vorbereitung
- Informationsbeschaffung
- Bewertung
- Aktive Eindringversuche
- Auswertung

# Data Privacy and Security

Penetrationstest

Vorgehen

Praktische Beispiele



## 4.6.1 Beispiel-Dateien mit Cross-Site-Scripting-Schwachstellen

Beschreibung:

Generell sollte kein Bedarf für die Verfügbarkeit von Beispiel-Dateien vorliegen, so dass sie gelöscht werden sollten.

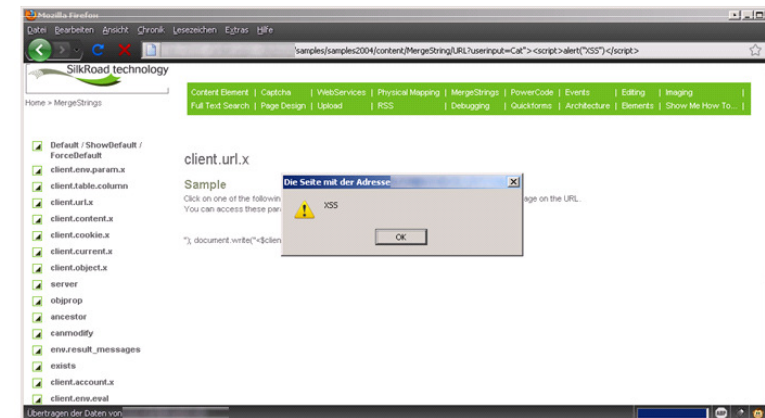


Abbildung 13 Cross-Site-Scripting in einer Beispiel-Datei

Risiko: Hoch

Maßnahme:

M.01. Löschen aller nicht benötigter Dateien



Danke für Ihre Aufmerksamkeit!

# Kontakt

Ute Seidel

Telefon: 0351 28 20 2873

E-Mail: [Ute.Seidel@t-systems.com](mailto:Ute.Seidel@t-systems.com)

Thomas Haase

Telefon: 0351 28 20 2206

E-Mail: [T.Haase@t-systems.com](mailto:T.Haase@t-systems.com)