

KISS

Was hat KISS mit Testen zu tun?

Klassische Fehlerquellen beim Testen



Wo liegen die **Probleme**:

- Interpretations-Spielräume in der Spezifikation:
=> hat man eine Systemanforderung missverstanden,
fallen auch die Testergebnisse nicht korrekt aus
- kein systematischer Test der Software
- klassischer Entwicklungsablauf: Test am Ende,
d.h. da wo Änderungen am teuersten sind
- zu Beginn des Projekts macht sich keiner Gedanken über den
Test, es gibt angeblich wichtigere Dinge und der Test liegt noch
in weiter Ferne

Wie heißt die „Zauberformel“?



KISS

K e e p i t S i m p l e a n d S t u p i d

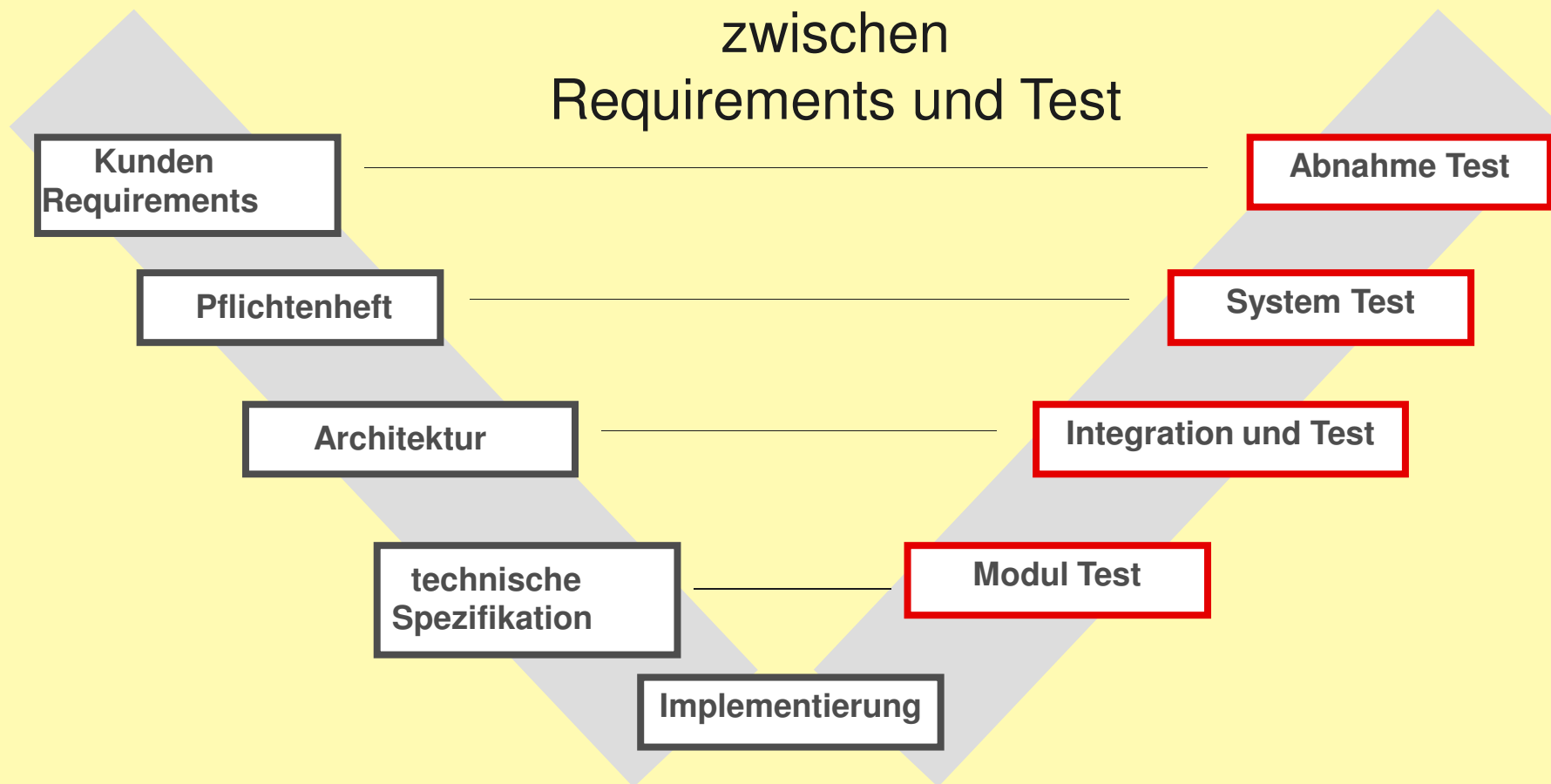
Wie vermeidet man Aufwände und Fehlerquellen beim Testen?



- Rechtzeitige Planung von Tests
- Geordnete Geschäftsprozesse
- Strukturierte Vorgehensweise
- Traceability

Rechtzeitiges testen.....

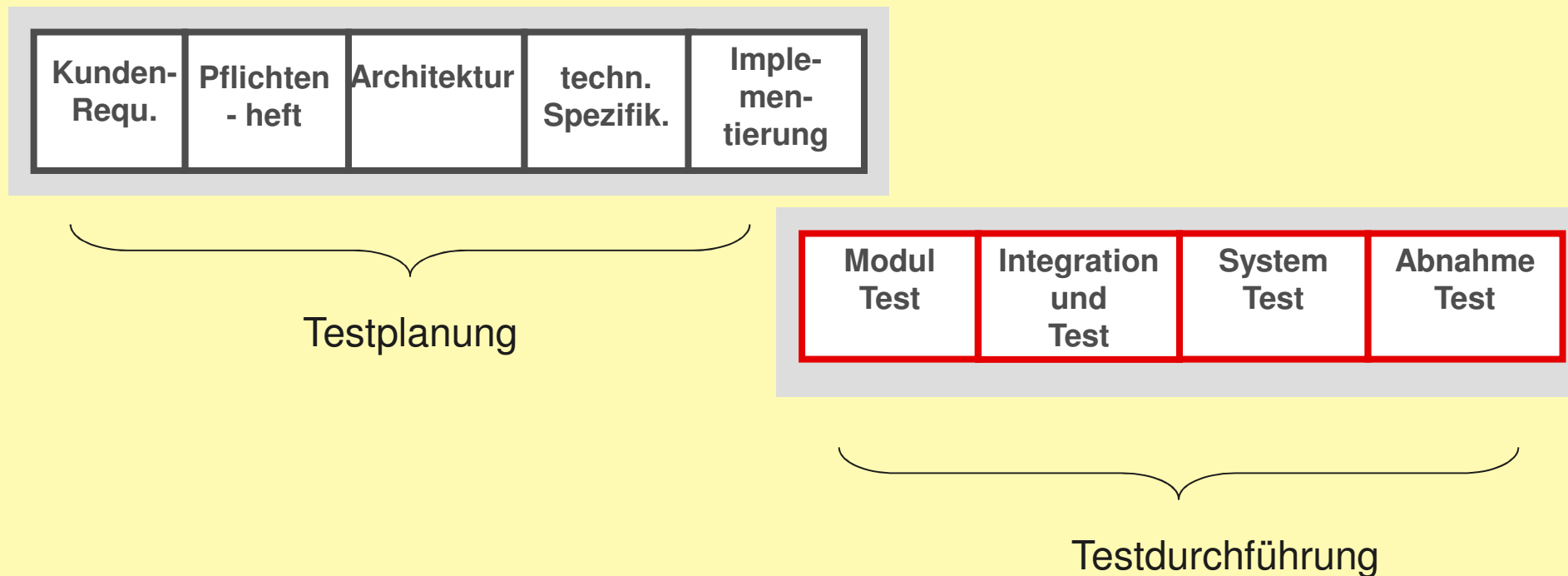
Der Zusammenhang zwischen Requirements und Test



Rechtzeitiges testen.....

Die zeitliche Abfolge der Test Aktivitäten

„Aufklappen“ des V-Modells, um eine zeitliche Abfolge zu erhalten

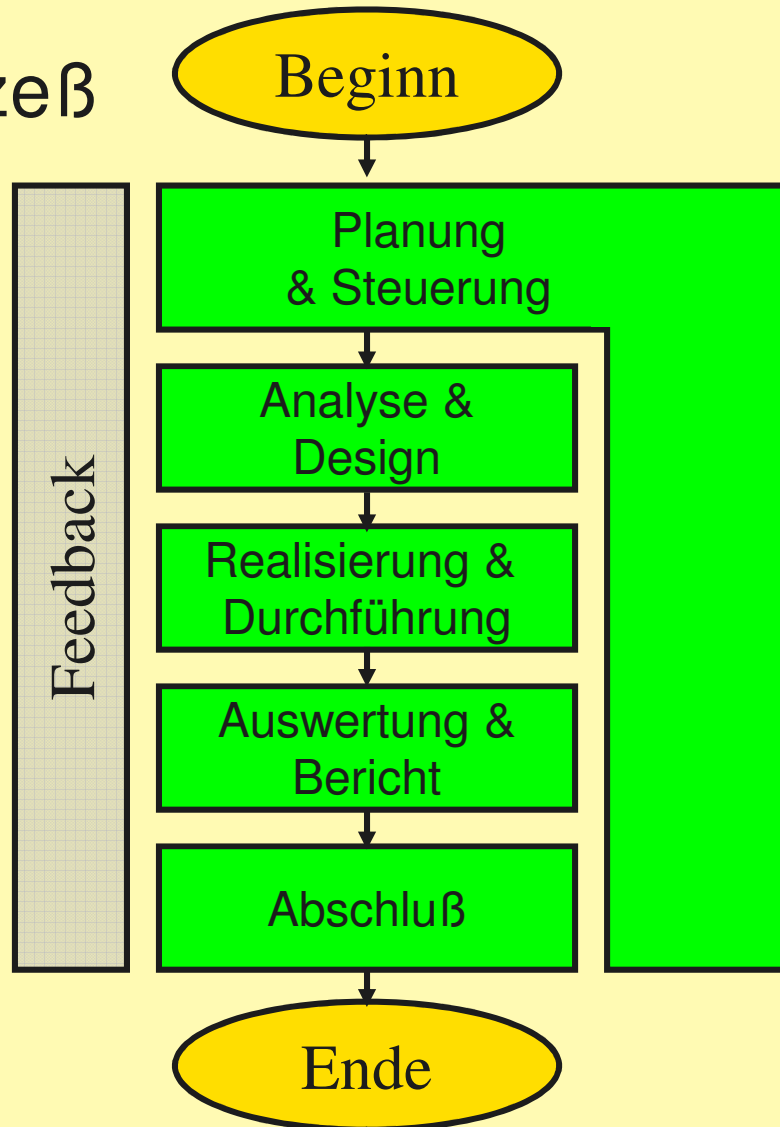


Strukturierte Vorgehensweise

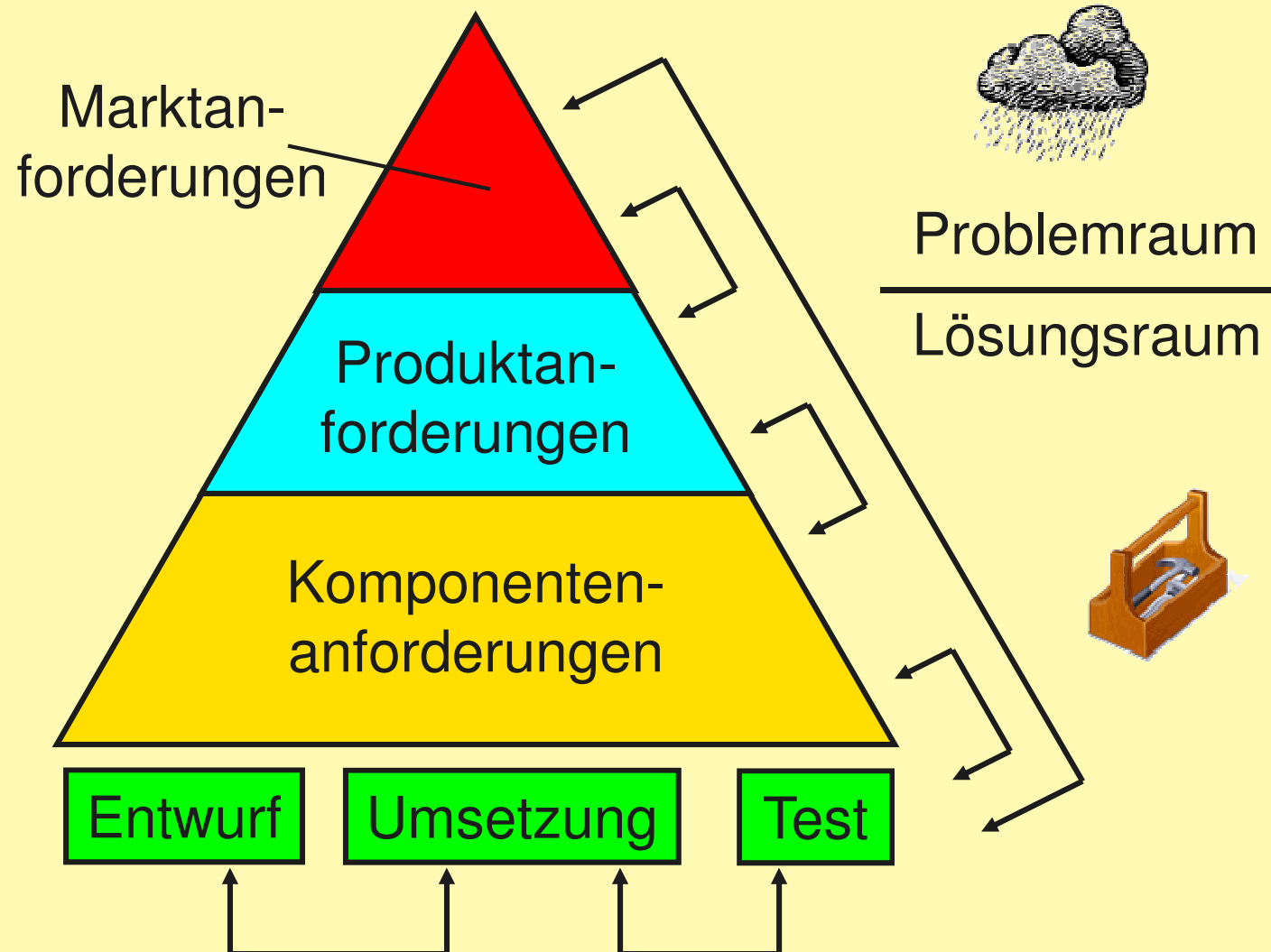


◆ Der Testprozeß nach ISTQB

Quelle: Imbus



Traceability



Wie vermeidet man Aufwände und Fehlerquellen beim Testen?



- Schaffung von Standards und Schnittstellen
- Single Source Philosophie
- Dokumenten-Management
- Modularisierung der Projekte
- Risiko-Abschätzung

- ◆ Schaffung von Standards und Schnittstellen im Unternehmen
 - zur Entwicklungsabteilung
 - zum Zulieferer
 - zum Kunden
 - ◆ Möglichst wenig, aber definierte Ansprechpartner zum Kunden und zum Zulieferer.
 - ◆ Definierte Schnittstellen zum Kunden und Zulieferer

- In allen Projekten die selbe Ordnerstruktur.
 - ◆ Leichte Orientierung in allen Projekten möglich.
- Einheitliche Systemlandschaften in den Projekten (Testautomaten, Testsoftware, Projektplanungstools, Dokumentationstools usw.)
 - ◆ Unternehmensweit einheitliches Projekt-Knowhow
 - ◆ Für den Systemlieferanten ist es leichter Speziallösungen für ein Unternehmen zu erstellen.

- ◆ Arbeiten mit Templates und Dokumentenvorlagen, die aktuell gehalten werden.
 - (Achtung: Wildwuchs in den Dokumenten unbedingt verhindern und nur abgestimmte Änderungen zulassen!)

- ◆ Möglichst wenig Medienbrüche in den Projektunterlagen

=> jeder Medienbruch verursacht den doppelten Änderungsaufwand, wenn das Projekt fortgeschrieben wird.

=> Jeder Medienbruch ist eine zusätzliche Fehlerquelle!

Single- Source Prinzip



- ◆ Alle Informationen und Dokumente werden **nur einmal** und **an einem Platz** definiert und abgespeichert!
- ◆ Dateiablagen in Verzeichnissen
 - **Wichtig!** Das ganze Projekt-Team legt die Daten am gleichen Ort ab.
- ◆ Programme / Spezifikationen / Excel / Skripts
 - Spezifikationen / Programm- Routinen, Listenfelder und Tabellen
 - Alle Informationen existieren nur einmal und sind im ganzen Dokument / Projekt verfügbar.

Realisierung von datenbankähnlichen Strukturen in Excel



- ◆ Einsatz von Listenfeldern
- ◆ Anwendungsbeispiele:
 - Intelligente (Test-) Spezifikationen
 - Testreports
 - Interaktive Kataloge

	J	K
	Test Fehleranalyse	
	Bewertung nicht notwendig	
	nicht durchgeführt	
	SW-Fehler identifiziert	
	SW-Fehler nicht	
	SW-Fehler vermutet	
	Test fehlerhaft	
	Bewertung nicht notwendig	
	nicht durchgeführt	
	SW-Fehler identifiziert	
	SW-Fehler nicht wahrscheinlich	
	SW-Fehler vermutet	
	Test fehlerhaft	

Realisierung von datenbankähnlichen Strukturen in Excel



- ◆ Einsatz der SVerweis-Funktion
- ◆ Anwendungsbeispiele:
 - Lookup-tables
 - Testreports
 - ◆ automatische Gewichtung von Fehlern
 - Interaktive Kataloge
- ◆ Vorteil:
 - Kriterien werden nur an einem einzigen Platz in der Datei geändert (Single Source-Philosophie)

- ◆ Nur das Testen was notwendig ist!
 - 100% Testtiefe bei kritischen Funktionen (neue oder geänderte Module)
 - verringerte Testtiefe bei unveränderten Modulen, die von den Änderungen in den anderen betroffen sind.
 - Wenn noch Zeit ist, die restlichen Module testen

Priorisierung von Fehlern

		Schadenshöhe		
		kein Schaden, Schönheitsfehler	Schaden	Großer Schaden
Auftreten von Problemen	wahrscheinlich	2	1	1
	möglich	3	2	1
	unwahrscheinlich	3	2	2

1	hohe Priorität
2	mittlere Priorität
3	niedrige Priorität

- ◆ Wie reportet man Testergebnisse so, dass Sie von anderen verstanden werden?

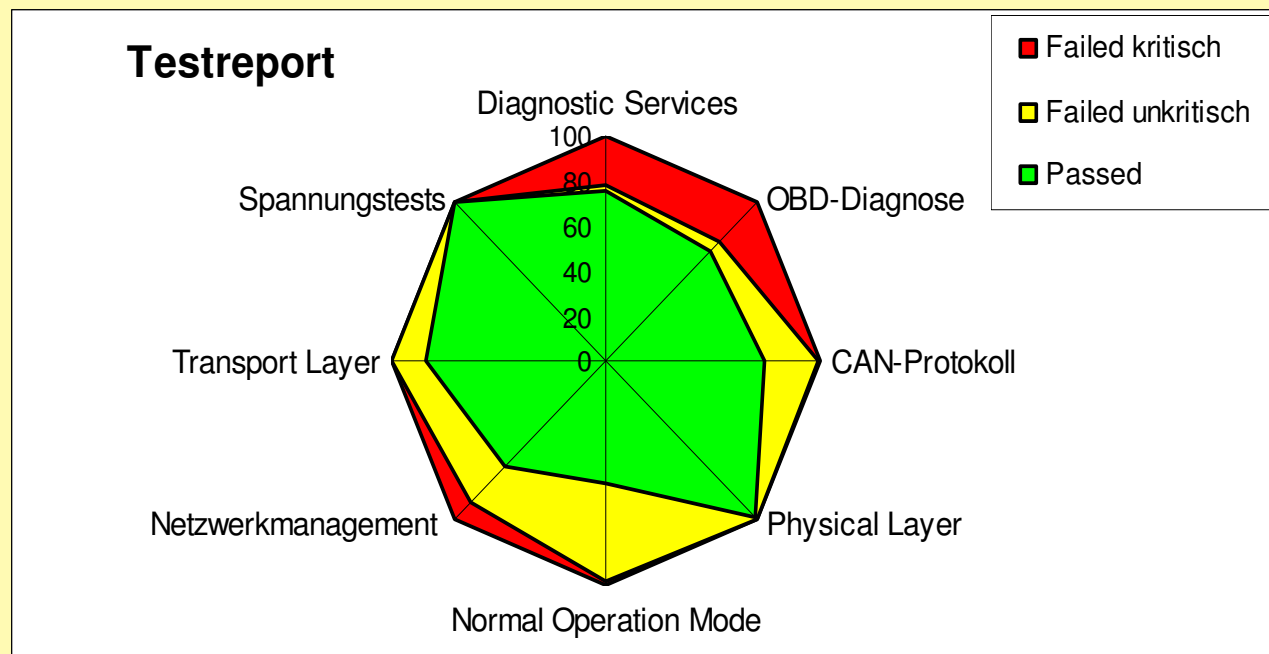
- ◆ Klare Strukturen
 - Gleiche Schnittstellen zu allen Teilnehmern im Projekt

- ◆ Klare Strukturen im Dokument
 - Gruppieren von Ergebnissen
 - Funktionen
 - Fehlerklassen
 - Priorisierung der Testfälle
 - verschiedene Aggregationsebenen
 - Übersichtsblatt

Ganz Wichtig!

- ◆ Nachvollziehbarkeit
 - Randbedingungen
 - Startbedingungen
 - Testablauf

- ◆ Grafische Darstellung der Testergebnisse
 - Spinnennetzdiagramm
 - Tortendiagramme



KISS



Viel Erfolg im
neuen Projekt!