Software Engineering softwareintensiver Systeme im Automobil.

Software-Qualitätssicherung.

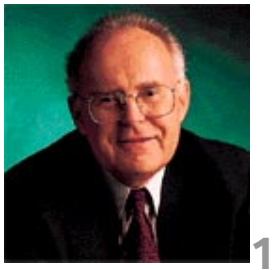
Dr. Heinz Treseler
28. Juni 2004, TU München

BMW Group

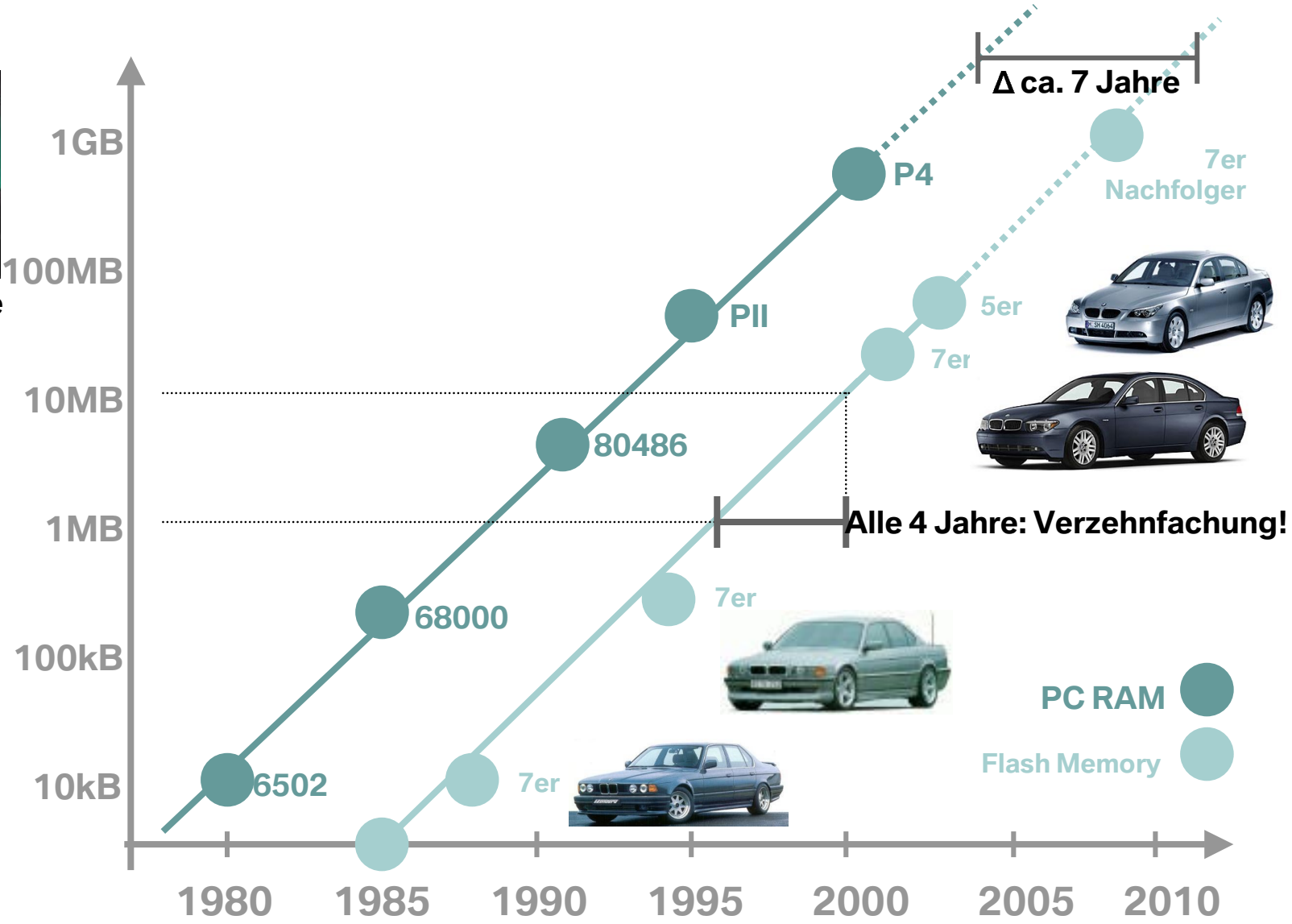
Dr. H. Treseler
BMW AG
28. Juni 2004

Elektronik und Software im Automobil.

Umfang der Software.



Gordon Moore
Intel Mitbegründer



Dr. H. Treseler
BMW AG

SW-Qualitätssicherung.
Motivation.

WALL STREET JOURNAL

Company X Orders Recall...(AP) Company X s
vehicles, includi
condensers and
the nation's larg
defective equipr
produced betwe
that will be reca
models betwee
Australia...(Staff
1998. Company
by 2000, where

THE REGISTER

Software fault forces Compa

Car giant Company X has ins
vehicles because of a progra
equipment.

A month ago, Company X de
problems had caused a num
At the time, the company bla
entangled with the throttle p

The Channel 4 programme,
British motorist Chris Merric
number of US motorists had

At the time, Company X said it there was no evidence of engine controls being to blame for 'unintended sudden acceleration' or faults with cruise control systems. It admitted that complaints about the vehicles not slowing as expected had prompted a recall in 1998, when incorrectly fitted floor mats beneath the pedals were replaced.

Today, Mike Vaughn, a spokesman for Company X in Detroit, told The Register that the programming of a chip in the powertrain control module was suspect. "In this case, it is the way it was programmed, not the way it was designed or manufactured," he said. "The fix is simply to reflash or reprogram it at the dealer."

AUTO – SPECIAL - NEWS

Firma X ruft halbe Million F zurück

D (rpo). Nach vielen anderen bekannten Auto-Herstellern hat es jetzt also Firma X getroffen: Der US-Autobauer ruft über eine halbe Million F in die Werkstätten zurück.

Dabei geht es um zwei freiwillige Rückrufaktionen für insgesamt 572.795 F-Modelle der Baujahre 2000 und 2001. Dies hat das Unternehmen am Montag in D mitgeteilt.

Eine Rückrufaktion bezieht sich auf einen Bolzen in der Vorderradaufhängung, der locker sein kann. Dies könne zu Geräuschen, Vibration oder in einer kleinen Anzahl von Fällen zu einer Trennung der Kugellagerverbindung führen.

Die zweite Rückrufaktion beziehe sich auf F-Fahrzeuge mit "Z"-Motoren. Es könnten potenzielle Probleme durch die Führung der Batteriekabel auftreten, die zu Rauch, schmelzendem Draht oder zu Feuer unter der Motorhaube führen könnten. Firma X führt die notwendigen Reparaturen kostenlos durch.

SW-Qualitätssicherung.
Gliederung.

1. Begriffe und Definitionen.

2. Überblick und Klassifizierung von Maßnahmen zur Qualitätssicherung.

3. Qualitätssicherung im Software-Entwicklungsprozess der Automobilindustrie.

4. Produktorientierte Qualitätssicherung:
   In den Software-Lebenszyklus integrierte Maßnahmen.

5. Prozessorientiert Qualitätssicherung:
   Bewertung und Verbesserung der Entwicklungsprozesse und des Software-Lebenszyklus.

Dr. H. Treseler
BMW AG

Begriffe und Definitionen.
Qualität (DIN 55350-11).

Qualität
ist die Gesamtheit von Eigenschaften und Merkmalen eines Produkts oder einer Tätigkeit, die sich auf deren Eignung zur Erfüllung gegebener Erfordernisse bezieht.

Begriffe und Definitionen.
Qualitätsmerkmale von Software (ISO 8402).

Funktionalität:
Richtigkeit, Angemessenheit, Interoperabilität, Ordungsmäßigkeit, Sicherheit

Zuverlässigkeit:
Reife, Fehlertoleranz, Wiederherstellbarkeit

Benutzbarkeit:
Verständlichkeit, Erlernbarkeit, Bedienbarkeit

Effizienz:
Zeitverhalten, Verbrauchsverhalten

Änderbarkeit:
Analysierbarkeit, Modifizierbarkeit, Stabilität, Prüfbarkeit

Übertragbarkeit:
Anpassbarkeit, Installierbarkeit, Konformität, Austauschbarkeit

Dr. H. Treseler
BMW AG

Begriffe und Definitionen.
Qualitätsmanagement (ISO 8402).

Qualitätsmanagement:
Alle Tätigkeiten der Gesamtführungsaufgabe, welche die Qualitätspolitik, Ziele und Verantwortungen festlegen sowie diese durch Mittel wie Qualitätsplanung, Qualitätslenkung, Qualitätssicherung und Qualitätsverbesserung im Rahmen des Qualitätsmanagementsystems verwirklichen.

Konkret: Qualitätsmanagement
ist die systematische Ausrichtung und Steuerung aller Aktivitäten eines Unternehmens auf die Qualität, also auf das Erfüllen von Kundenanforderungen. Die Ziele des Qualitätsmanagements sind:

- Kundengerechtes Arbeiten,
- das ständige Zufriedenstellen der Kunden,
- Vermeiden von Fehlern,
- Beseitigen von Risiken,
- zielorientiertes Handeln,
- ständige Verbesserung und Streben nach Innovation,
- motivierte und qualifizierte Manager und Mitarbeiter.

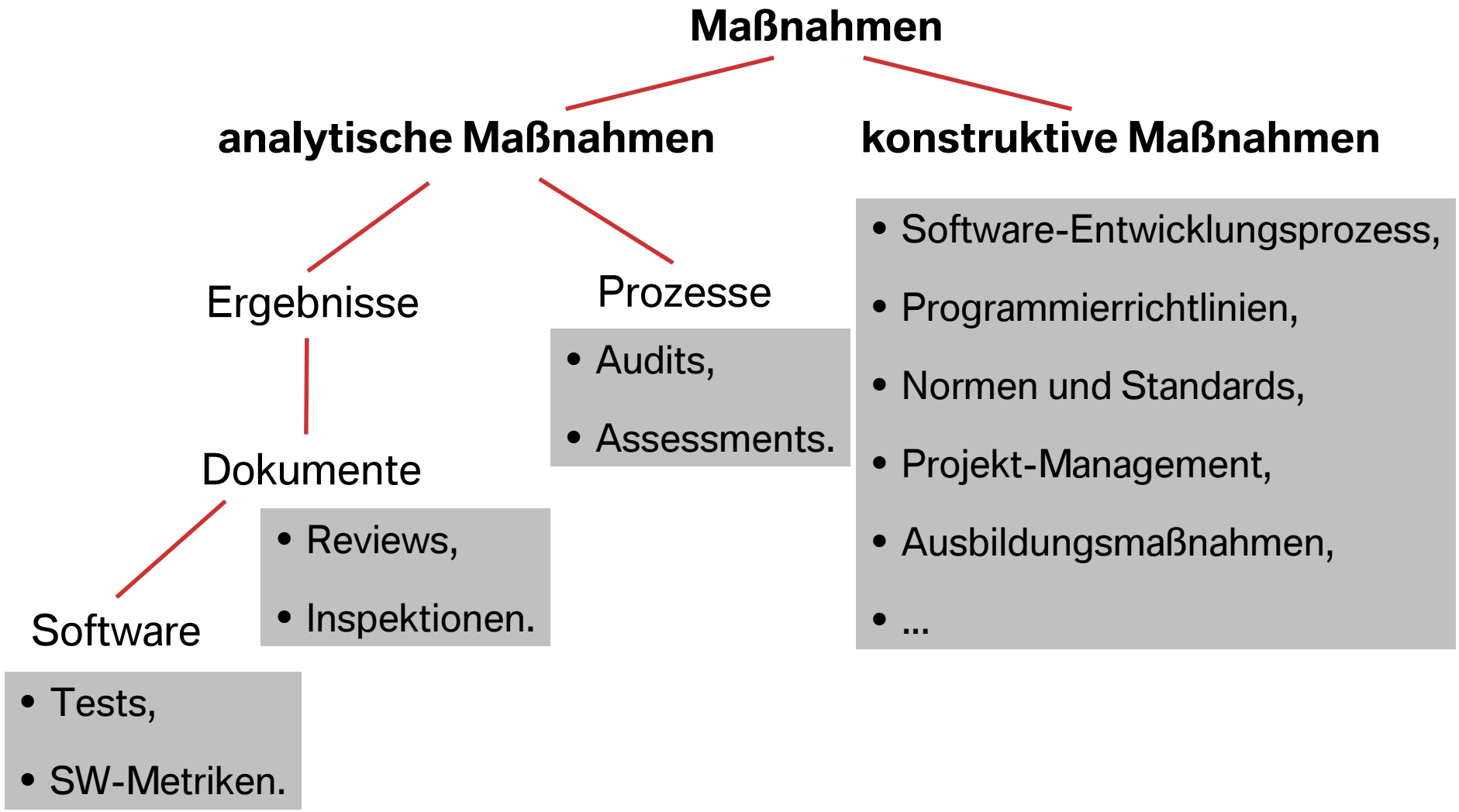
Dr. H. Treseler
BMW AG

Begriffe und Definitionen.
Qualitätssicherung (ISO 8402).

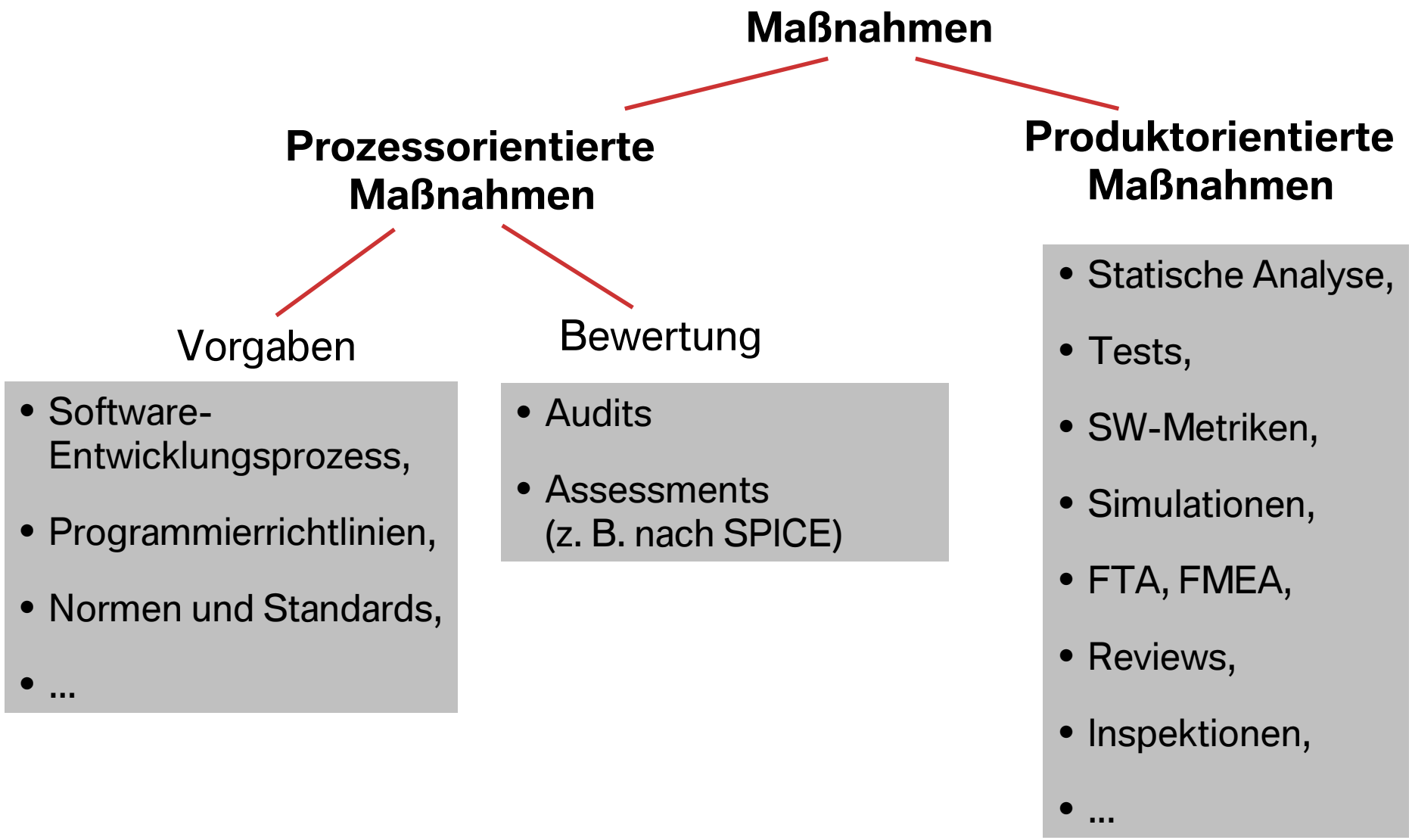
Qualitätssicherung (allgemein):
Alle geplanten und systematischen Tätigkeiten, die notwendig sind, um ein angemessenes Vertrauen zu schaffen, dass ein Produkt oder eine Dienstleistung die gegebenen Qualitätsanforderungen erfüllt.

Qualitätssicherung (softwarespezifisch):
Die Gesamtheit aller Maßnahmen und Hilfsmittel, die mit dem Ziel eingesetzt werden, die gestellten Anforderungen an den Entwicklungs- und Wartungsprozess sowie an das Softwareprodukt zu erreichen.

Begriffe und Definitionen.
Maßnahmen zur Qualitätssicherung (1/3).



Begriffe und Definitionen.
Maßnahmen zur Qualitätssicherung (2/3).



Begriffe und Definitionen.
Maßnahmen zur Qualitätssicherung (3/3).

Produktorientierte Maßnahmen

Statische Prüfungen

- Statische Analyse,
- SW-Metriken,
- symbolische Analyse,
- formale Verifikation,
- Reviews,
- Inspektionen,
- ...

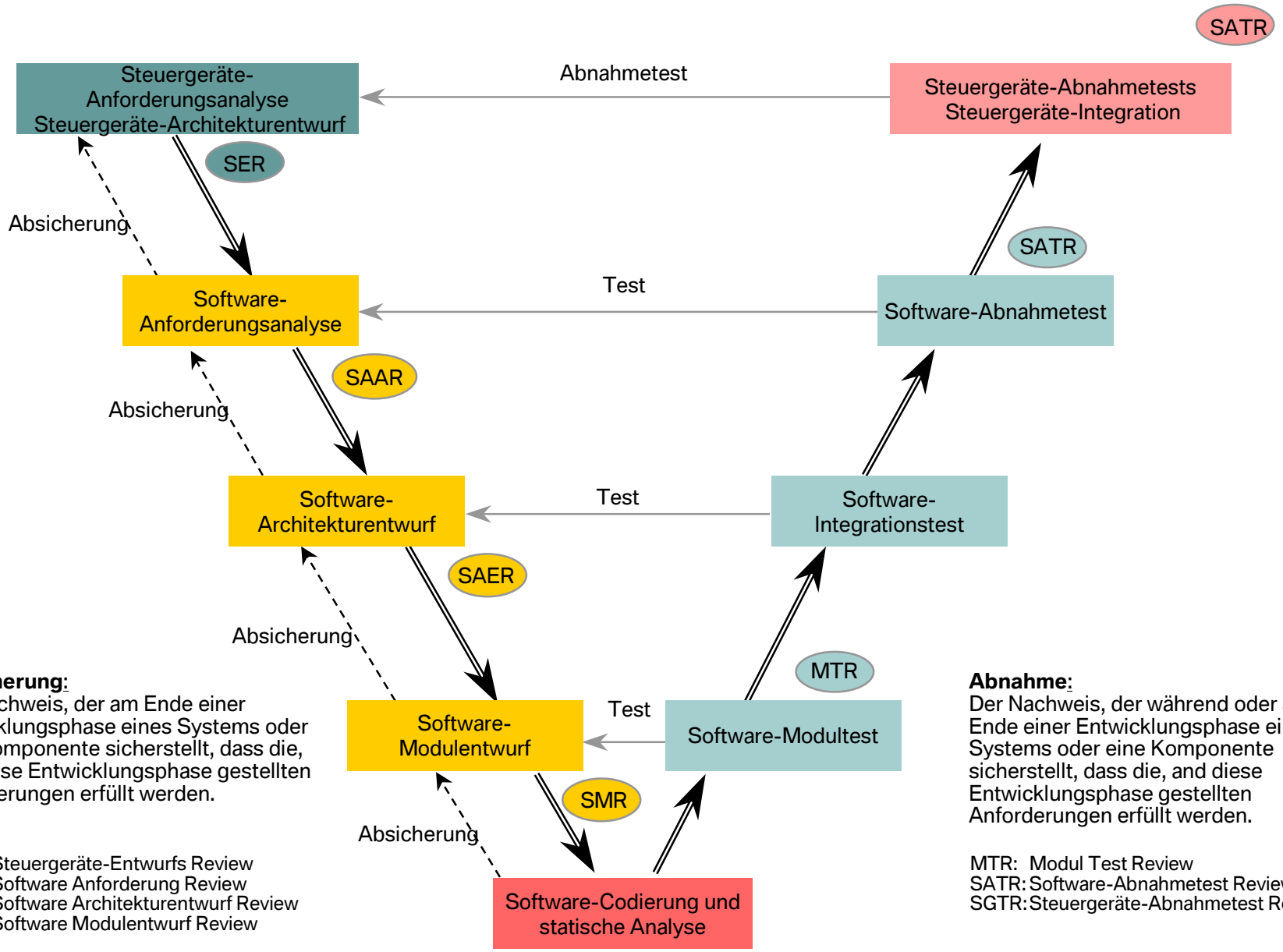
Dynamische Prüfungen

- Tests,
- Simulationen,
- ...

Dr. H. Treseler
BMW AG
28. Juni 2004

Begriffe und Definitionen.

Verfahrensmodell der Software-Entwicklung.



Absicherung:
Der Nachweis, der am Ende einer Entwicklungsphase eines Systems oder eine Komponente sicherstellt, dass die, and diese Entwicklungsphase gestellten Anforderungen erfüllt werden.

SER: Steuergeräte-Entwurfs Review
SAAR: Software Anforderung Review
SAER: Software Architekturentwurf Review
SMR: Software Modulentwurf Review

Abnahme:
Der Nachweis, der während oder am Ende einer Entwicklungsphase eines Systems oder eine Komponente sicherstellt, dass die, and diese Entwicklungsphase gestellten Anforderungen erfüllt werden.

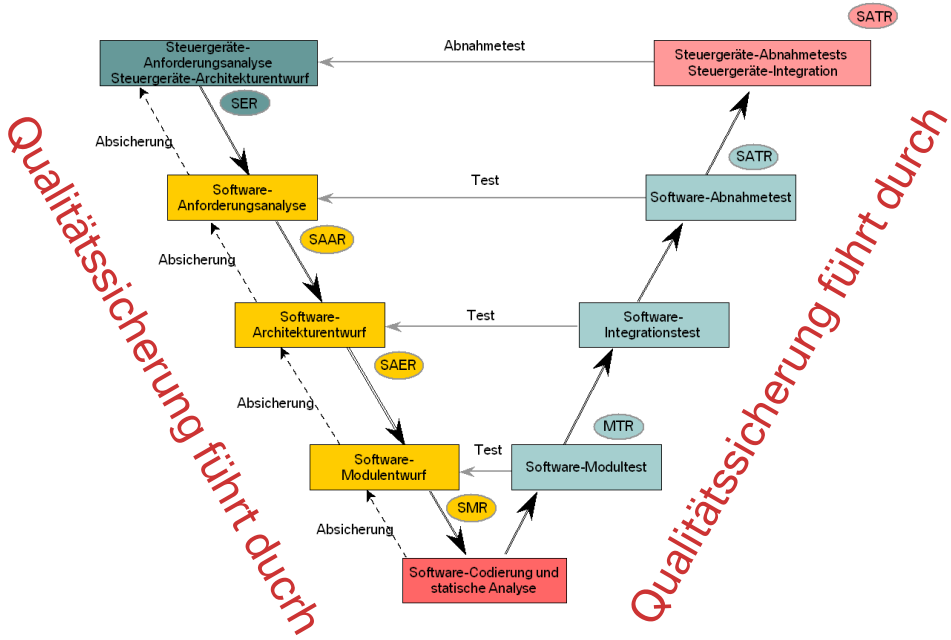
MTR: Modul Test Review
SATR: Software-Abnahmetest Review
SGTR: Steuergeräte-Abnahmetest Review

Dr. H. Treseler
BMW AG

Begriffe und Definitionen.

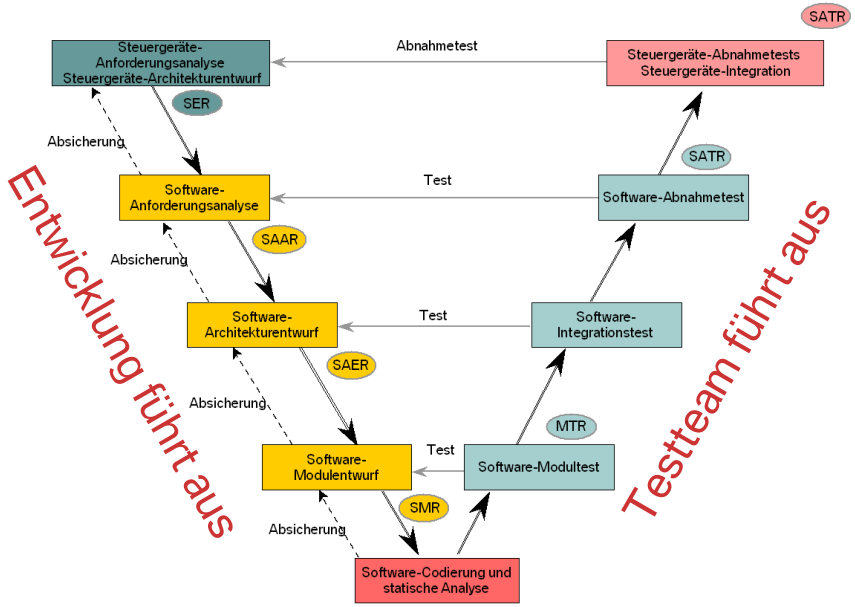
Unterschiedliche Rollen der Qualitätssicherung im Software-Entwicklungsprozess.

Ausführende Rolle:



V-Modell 97

Begleitende Rolle:



Qualitätssicherung begleitet

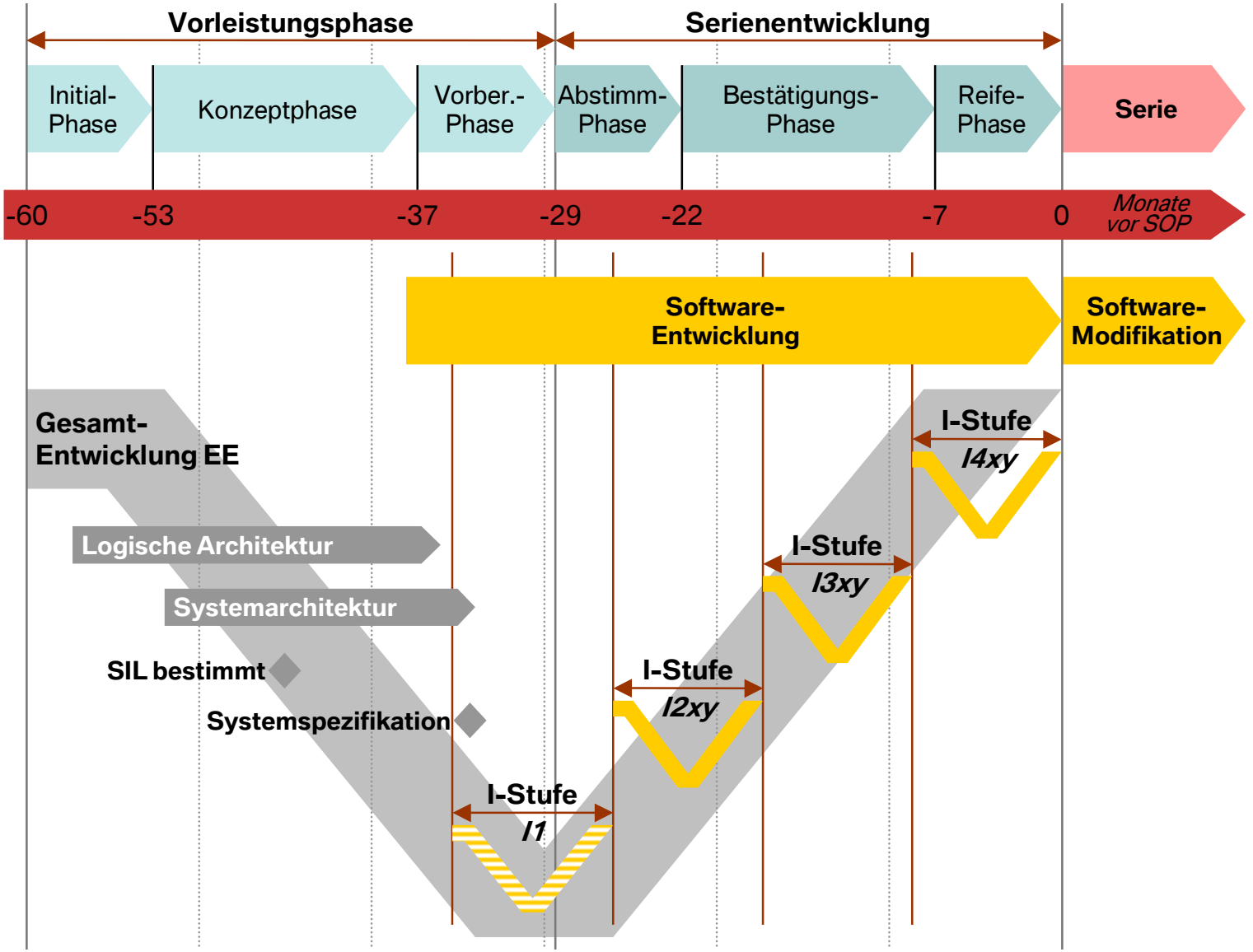
BMW Group Standard
Automotive SPICE
CMMI

Unklar definierte Rolle: ISO 12207

Dr. H. Treseler
BMW AG
28. Juni 2004

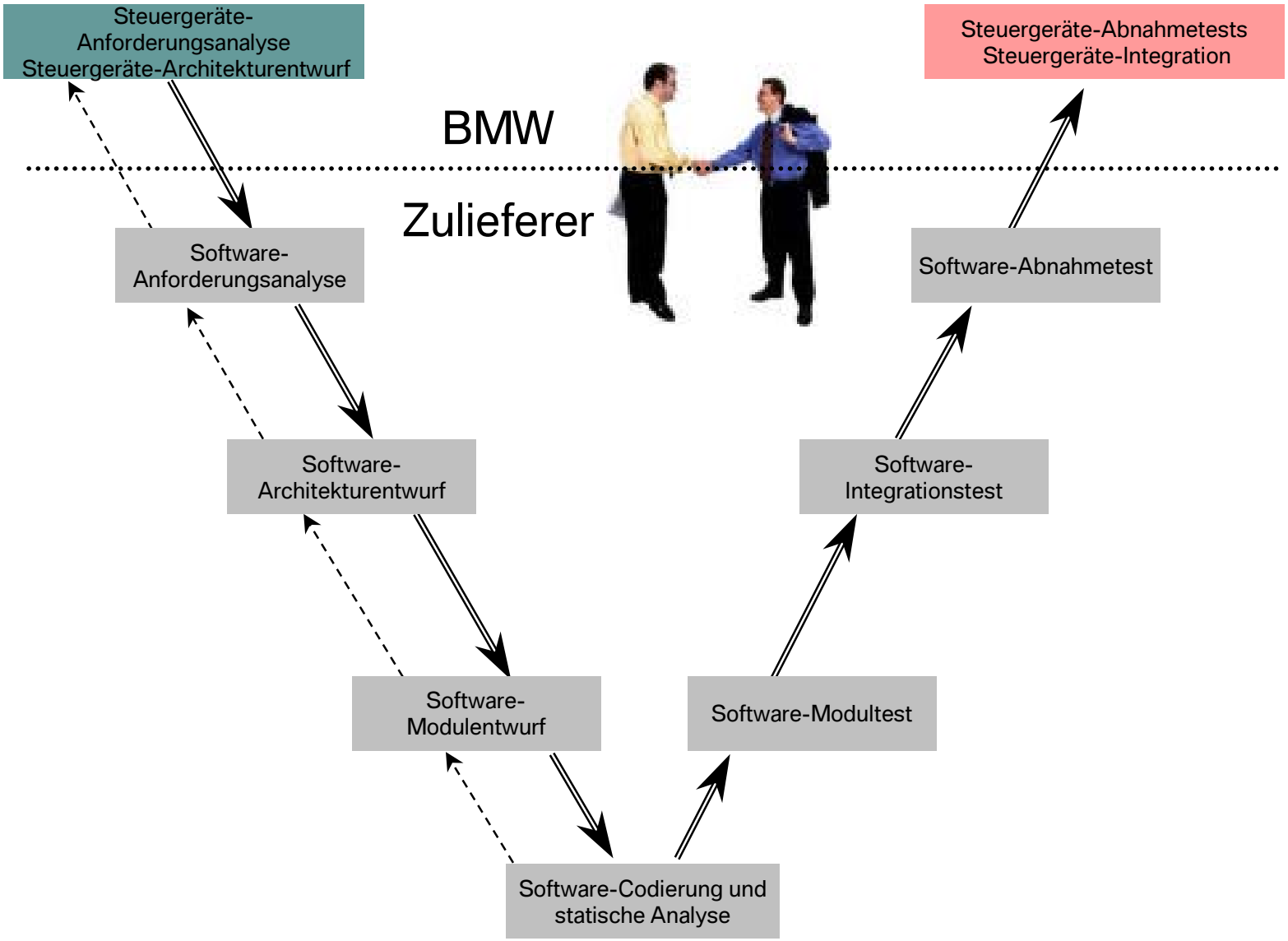
Automobil-Entwicklungsprozess.

Software-Entwicklung im Gesamtprozess.



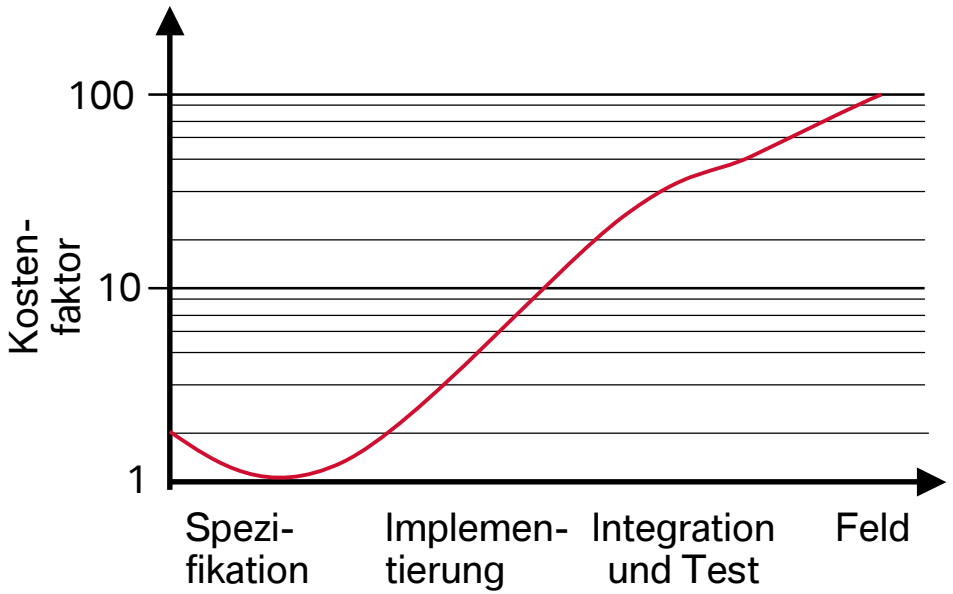
Dr. H. Treseler
BMW AG

Automobil-Entwicklungsprozess.
Verteilte Entwicklung mit Zulieferern.



Dr. H. Treseler
BMW AG

Automobil-Entwicklungsprozess.
Kosten einer Fehlerbehebung.



⇨ Fehler frühzeitig erkennen.
⇨ Fehler vermeiden.

Produktorientierte Qualitätssicherung.
Methoden.

- Statische Analysen:
  - Überprüfung von Programmierrichtlinien
  - Symbolische Interpretation
  - SW-Metriken
- Testen:
  - Testprozess
  - SW-Modultests
  - Testmanagement
  - Test-Toollandschaft
- Formale Verifikation
- Inspektion

Dr. H. Treseler
BMW AG

Produktorientierte Qualitätssicherung.

Statische Code-Analyse: Konzept.

Definition eines Programmierstandards zur Gewährleistung von
- Lesbarkeit,
- Änderbarkeit / Portierbarkeit,
- Stabilität und
- Testbarkeit.

Automatisierte Überprüfung des Programmierstandards durch spezielle Analyse-Werkzeuge.

Nur eingeschränkte manuelle Reviews zur Kontrolle an ausgewählten Stellen.

⇨ Qualitative Verbesserung des Codes.
⇨ Qualifikation der Entwickler.

Produktorientierte Qualitätssicherung.
Statische Code-Analyse: Software-Richtlinien.

- Basis der BMW Group Programmierrichtlinien für mit C programmierte Steuergeräte-Software: MISRA
  (*Development Guidelines for Vehicle Based Software*,
  Motor Industry Software Reliability Association, UK, 1994),
  127 Regeln, z. B.:
  - Keine Bit-Operationen mit vorzeichenbehafteten Integern,
  - keine implizite Typumwandlungen,
  - keine „Überschattung“ globaler Variablen durch lokale Variablen,
  - keine Lesezugriffe auf nicht initialisierte Variablen,
  - keine konträren Variablen- / Funktionsdeklarationen,
  - kein unerreichbarer Code,
  - keine Verwechslung von Zuweisungen mit logischen Vergleichen.
- Weitere Programmierrichtlinien für C++ und Java sind in Arbeit.

Produktorientierte Qualitätssicherung.
Statische Code-Analyse: Anwendung bei BMW.

- Start der Pilotierung Anfang 2001:
  - Evaluation verfügbarer Tools.
  - Anpassung der MISRA-Regeln an BMW-Gegebenheiten.
  - Durchführung von 20 Pilotprojekte:
    überwiegend positives Feedback.

- Verpflichtende Anwendung bei allen neuen Fahrzeugen.

Produktorientierte Qualitätssicherung.

Statische Code-Analyse: Beispiel.

Exemplarische Regeln	Strict (93)	Med. (56)	Weak (26)	Anz. Warn.
ISO 9899 Standard-C verwenden (Portabilität).	+	+	+	**196**
Initialisierung aller Variablen (höchst gefährliche Gefahrenquelle).	+	+	+	**11**
if- und *else if*-Konstrukte mit *else* abschließen (gefährliche Gefahrenquelle).	+	+	+	**1**
Keine Veränderung der Zählvariablen innerhalb eines *for*-Blocks (Gefahrenquelle).	+	+		**4**
Bit-Felder nur als *unsigned int* oder *signed int* definieren (Portabilität).	+			**15**
Keine überlappenden Speicherbereiche (*union*) benutzen (gefährliche Gefahrenquelle).	+	+	+	**69**
...	...	...	...	**...**
Anzahl verletzter Regeln	**43**	**32**	**20**	**43**
Anzahl der Warnungen	**8.845**	**5.148**	**2.827**	**8.845**

Produktorientierte Qualitätssicherung.

Statische Code-Analyse: Erweiterung um symbolische Interpretation.

Statische Analyse auf Einhaltung der MISRA-Programmierrichtlinien:

- **Ziel**: Fehler vermeiden, Wartbarkeit & Portierbarkeit gewährleisten
- Findet Fehler wie z.B. Verwechselung von Zuweisung und Vergleich (=, ==) sowie Verwendung von gefährlichen Programmierkonstrukten, z.B. Zeiger auf Zeiger auf Zeiger.

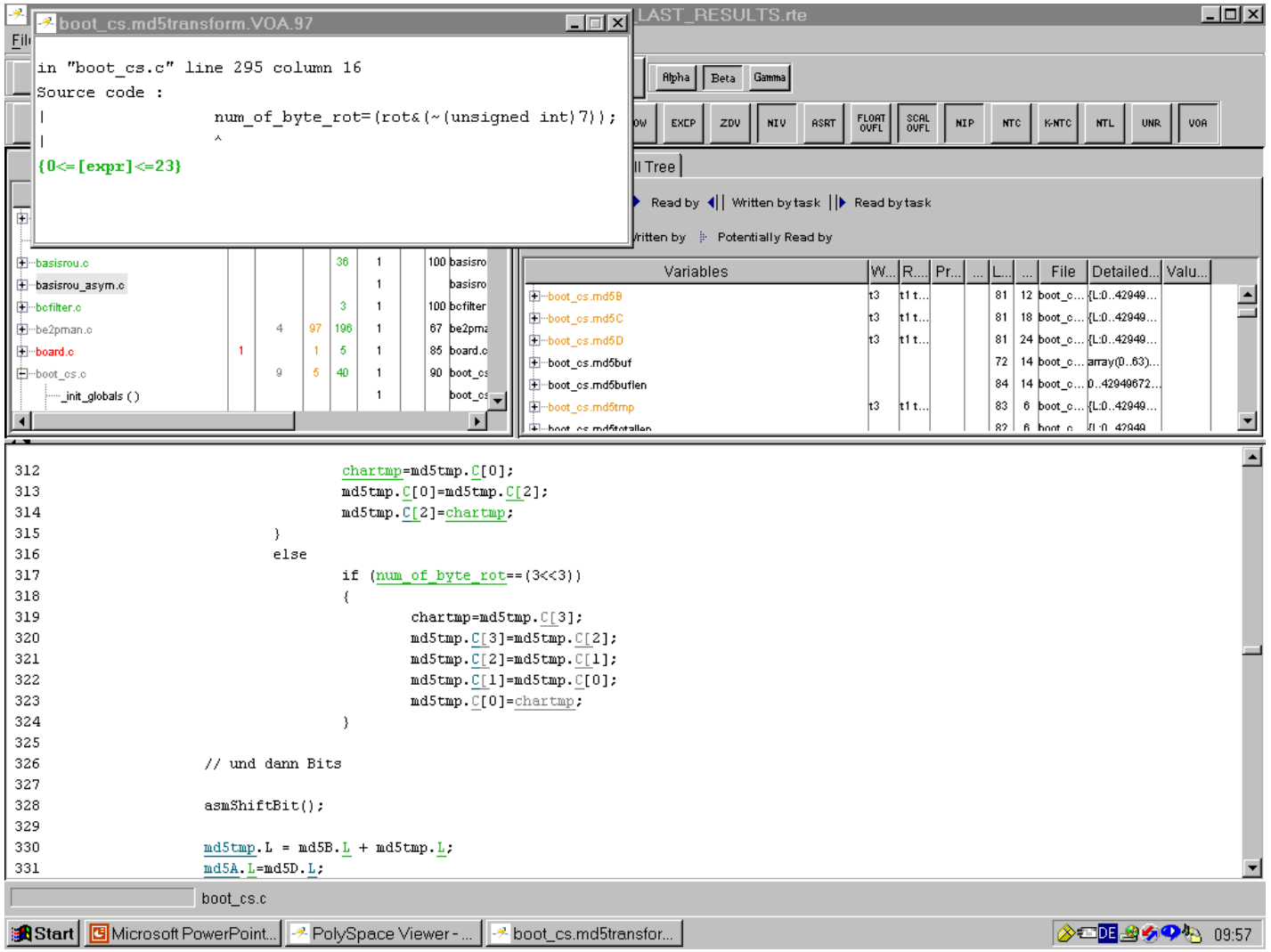
Erweiterung um symbolische Interpretation:

- Mathematische Analyse des Quellcodes,
- **Ziel**: Fehler finden,
- findet Fehler die zur Laufzeit auftreten, z.B. Überschreitung von Feldgrenzen, Division durch Null, Toter Code.

Dr. H. Treseler
BMW AG
28. Juni 2004

Produktorientierte Qualitätssicherung.

Statische Code-Analyse mit symbolischer Interpretation: Beispiel: Toter Code.



Produktorientierte Qualitätssicherung.
Statische Analyse: Software-Codemetriken.

- Bewertung der Komplexität des Codes.
- Bewertung weiterer Qualitätsmerkmale, wie z. B. Wartbarkeit und Testbarkeit.

Wichtige durch Code Checker berechenbare Metriken:

Metrik	min	max
Zyklomatische Komplexität (Anzahl der möglichen Verzweigungen + 1)	1	10
Maximale Verschachtelungstiefe	0	5
Verhältnis von Kommentar zu Codezeilen	0,2	1
Zahl der GOTOs	0	0
...		

⇨ Qualitative Verbesserung des Codes.
⇨ Qualifikation der Entwickler.

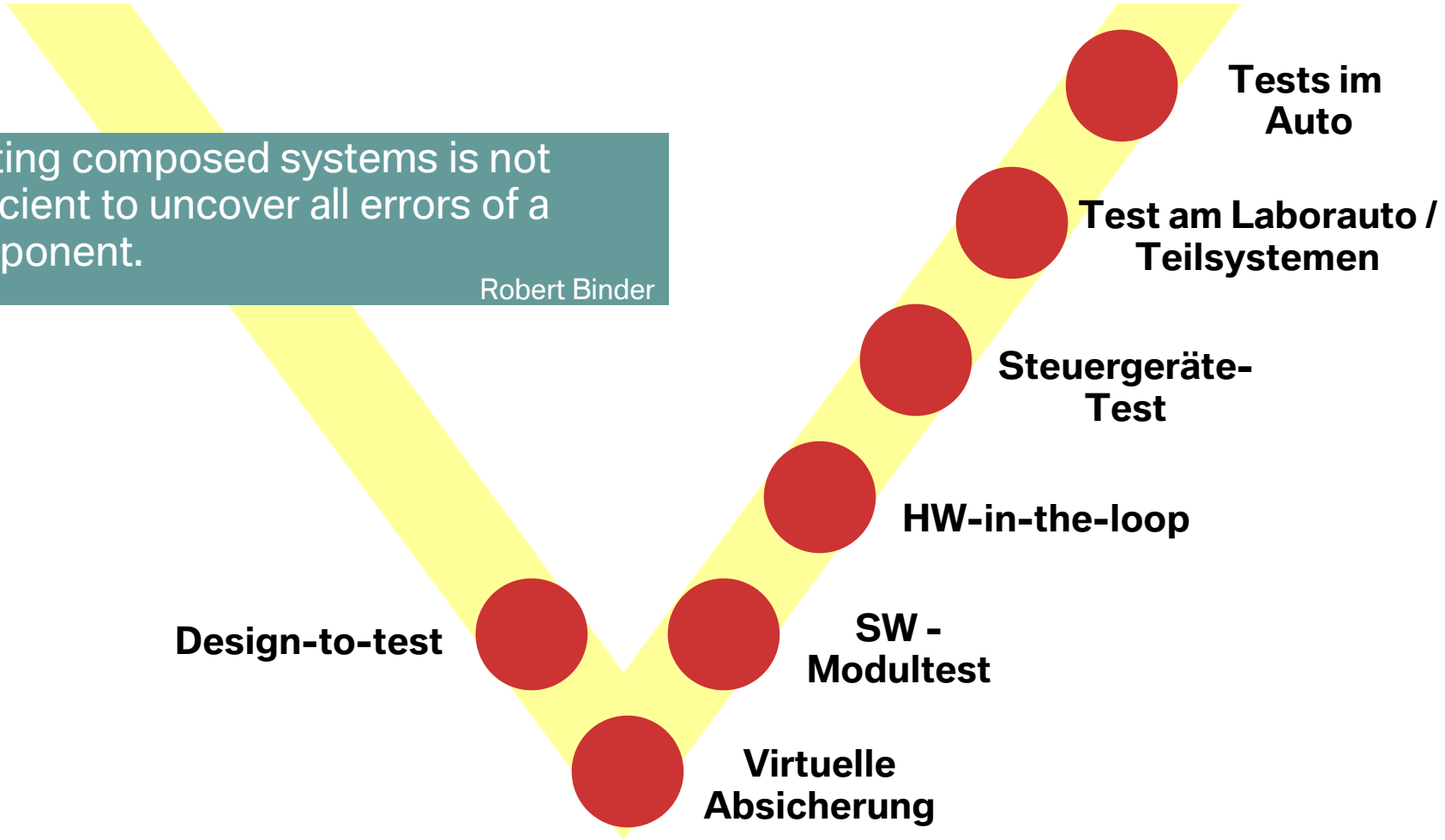
Dr. H. Treseler
BMW AG

Produktorientierte Qualitätssicherung.
Hierarchische Teststrategie.

Anforderungen

Testing composed systems is not sufficient to uncover all errors of a component.

Robert Binder



Produktorientierte Qualitätssicherung.
Testprozess.

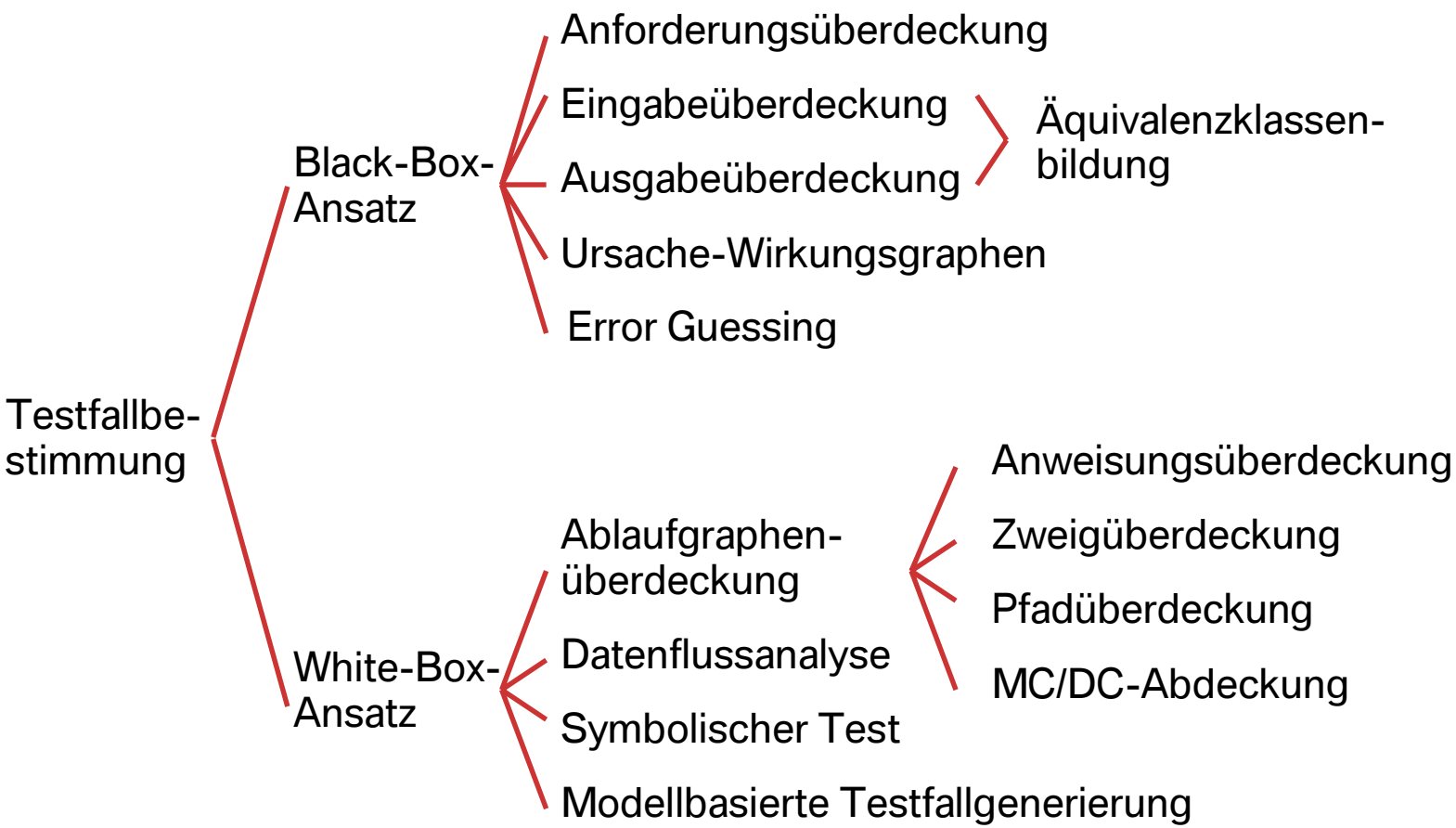
Der Testprozess umfasst folgende Tätigkeiten:

1. Planung der Tests
2. Spezifikation der Tests
3. Durchführung der Tests
4. Dokumentation der Tests
5. Abschluss der Tests
6. Validierung der funktionalen Sicherheit

Der Testprozess erstreckt sich über mehrere Teststufen:

- Komponententest
  - Virtuelle Absicherung
  - Software-Modultest
  - Hardware-in-the-loop
  - Steuergeräte-Test
- Teilsystemtest
- Test am Laborauto
- Tests im Auto

Produktorientierte Qualitätssicherung.
Bestimmung der Testfälle.



Dr. H. Treseler
BMW AG

Produktorientierte Qualitätssicherung.

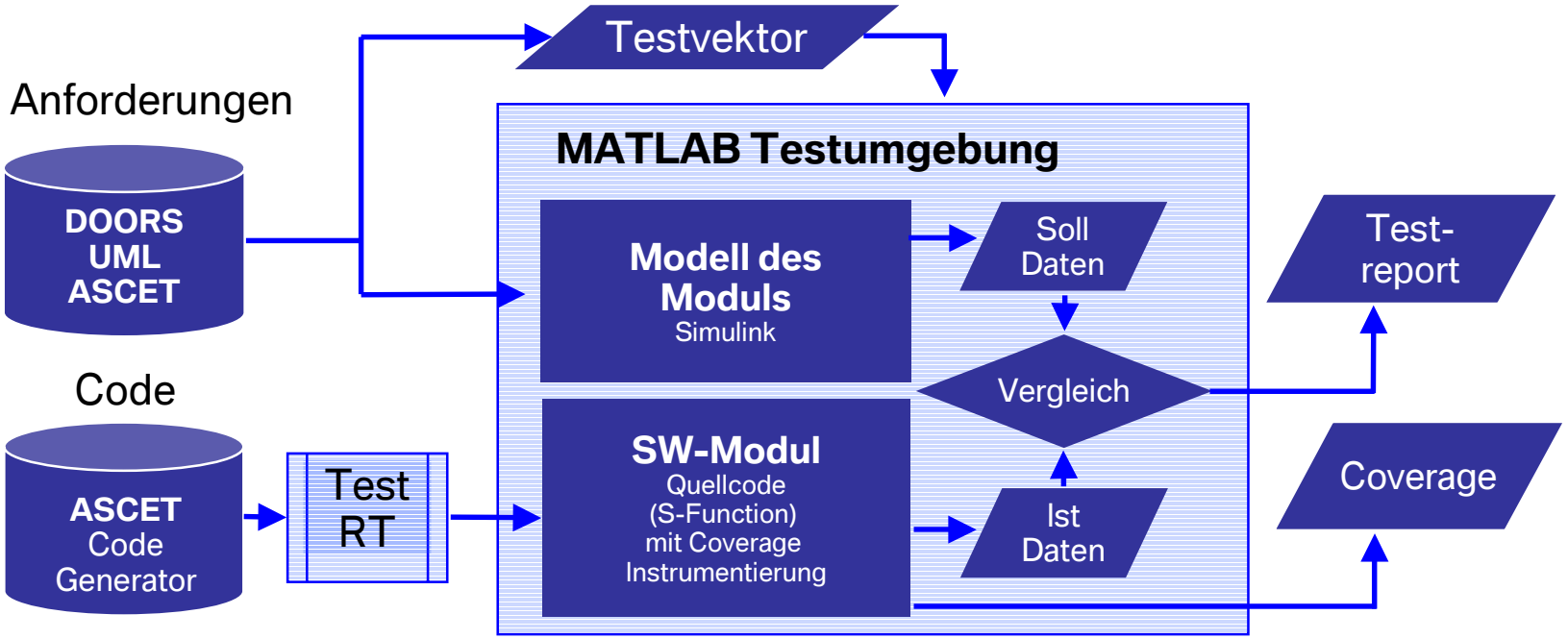
SW-Modultest: Diversitärer Testansatz.

Mögliche Testumgebungen:
- Host (PC)
- Evalboard
- Target (Steuergerät)

Code-Basis:
- Simulationscode
- µC-Code

Vorschlag:
- Plattform: Host, später Target
- Testumgebung: MATLAB
- Coverage: Rational TestRealtime

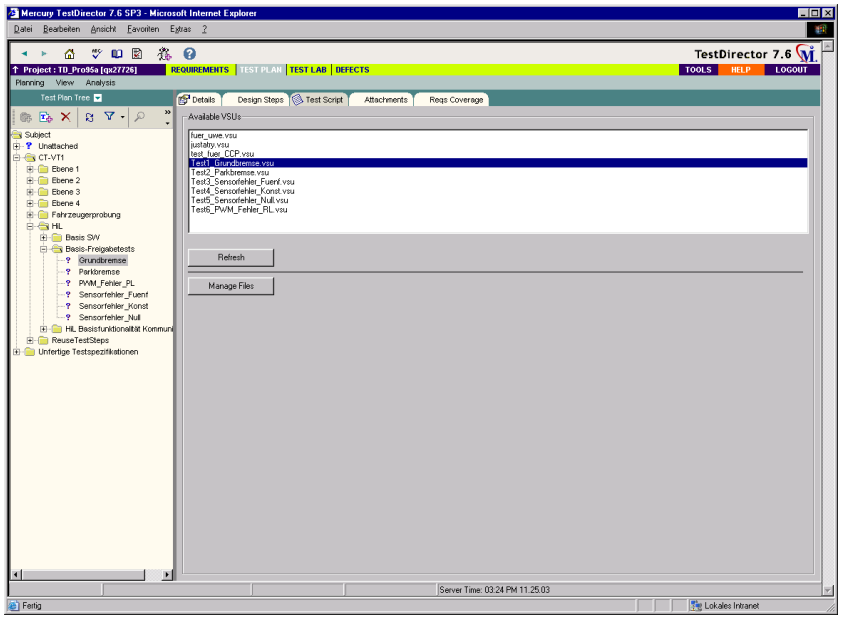
Code-Basis:
- µC-Code



Produktorientierte Qualitätssicherung.

Servergestütztes Testmanagement.

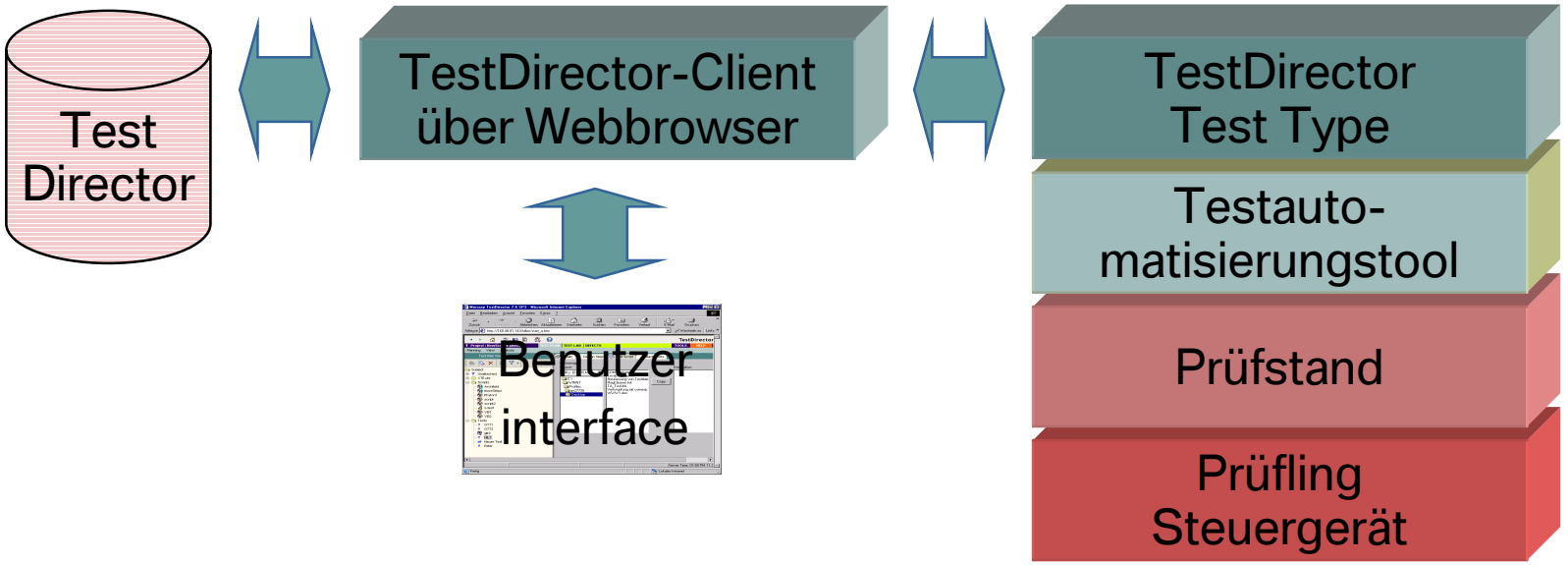
- Toolunterstützte Testplanung
- Zentrale Datenhaltung der Testfälle und Testergebnisse
- Verfolgbarkeit von Anforderungen und Tests
- Einfache Testnachweise für sicherheitsrelevante Systeme
- Benutzerrechte für Testingenieure und Tester
- Leichte Erstellung von Reports und Analysen
- Defect- und Problem-Management



TestDirector

Produktorientierte Qualitätssicherung.

Testmanagement: Architektur.



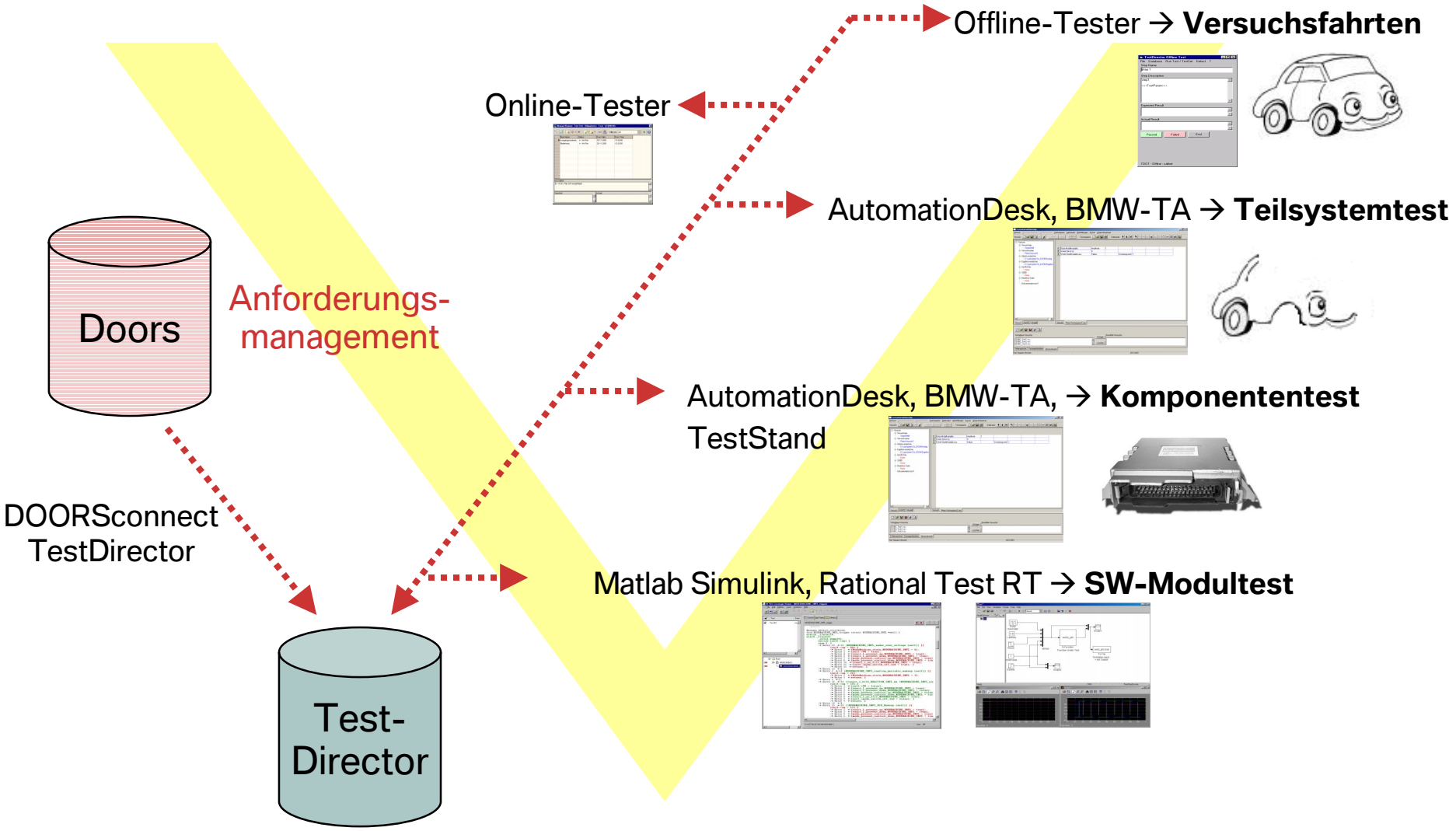
Vorteile der Integration Testmanagement / Testautomatisierung:

– Zentrale Datenablage für Testfälle und Testergebnis
– Einheitliche Benutzeroberfläche für Tester
– Schichtenarchitektur ermöglicht flexible Erweiterungen

Ausblick:

– Einheitliches Datenformat für Testfälle, Testergebnisse und Fehler
– Anbindung Messdatenbank (ASAM ODS)

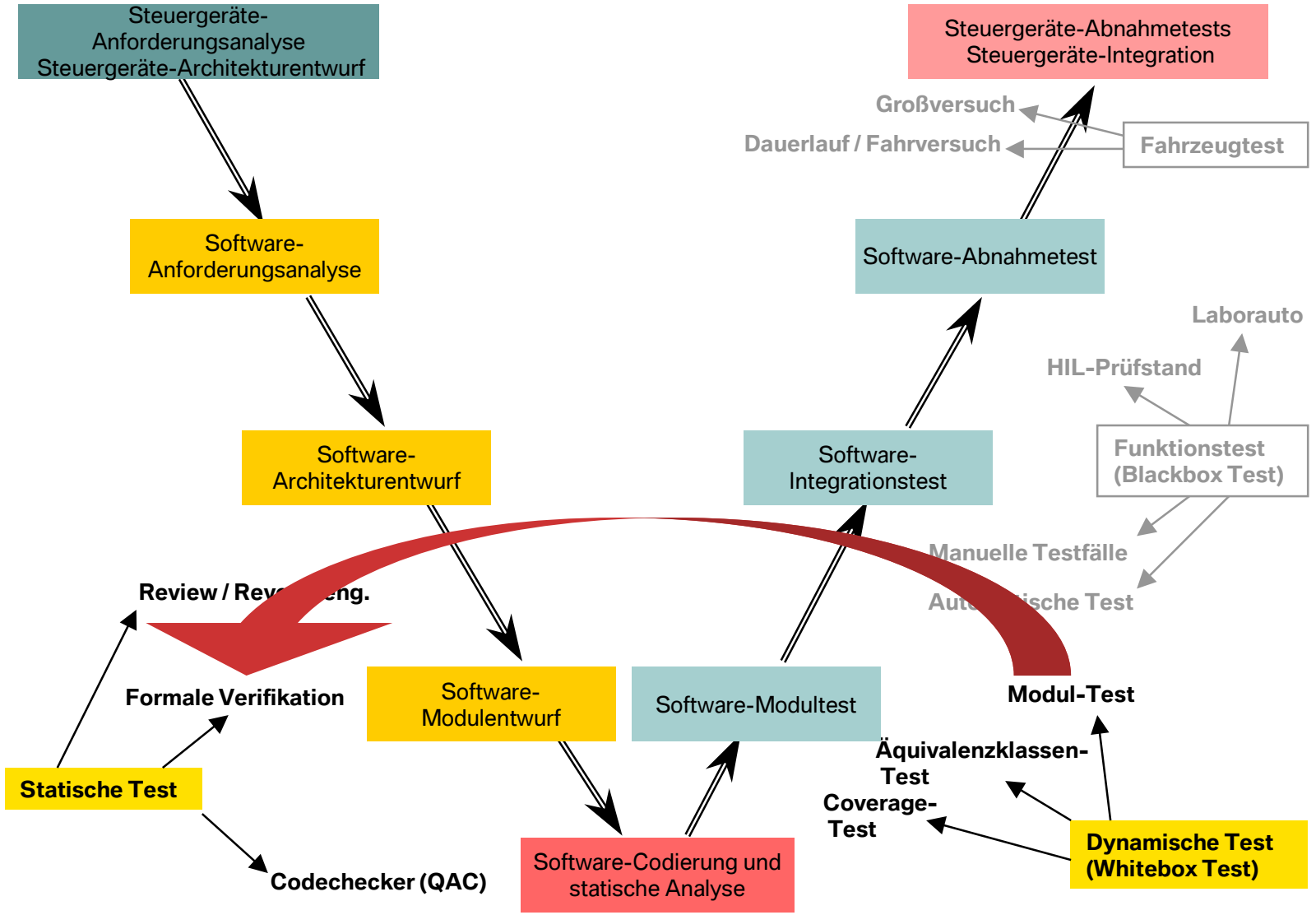
Produktorientierte Qualitätssicherung.
Toollandschaft Testen.



Dr. H. Treseler
BMW AG
28. Juni 2004

Produktorientierte Qualitätssicherung.

Formale Verifikation: Integration in den Absicherungsprozess.

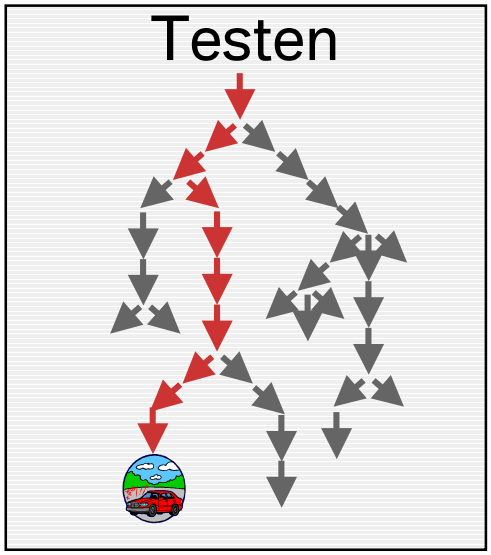


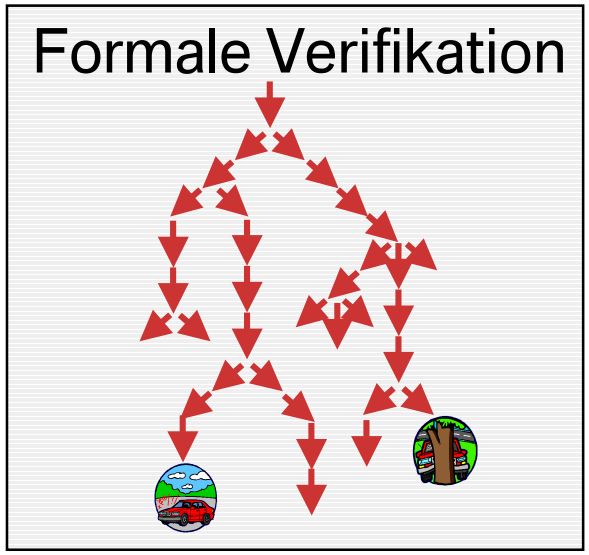
Dr. H. Treseler
BMW AG

Produktorientierte Qualitätssicherung.

Formale Verifikation: Integration in das Case-Tool ASCET-SD

Formale Verifikation
– deckt Fehler und Lücken der formalen Spezifikation auf
– ermöglicht die vollständige Absicherung der Spezifikation

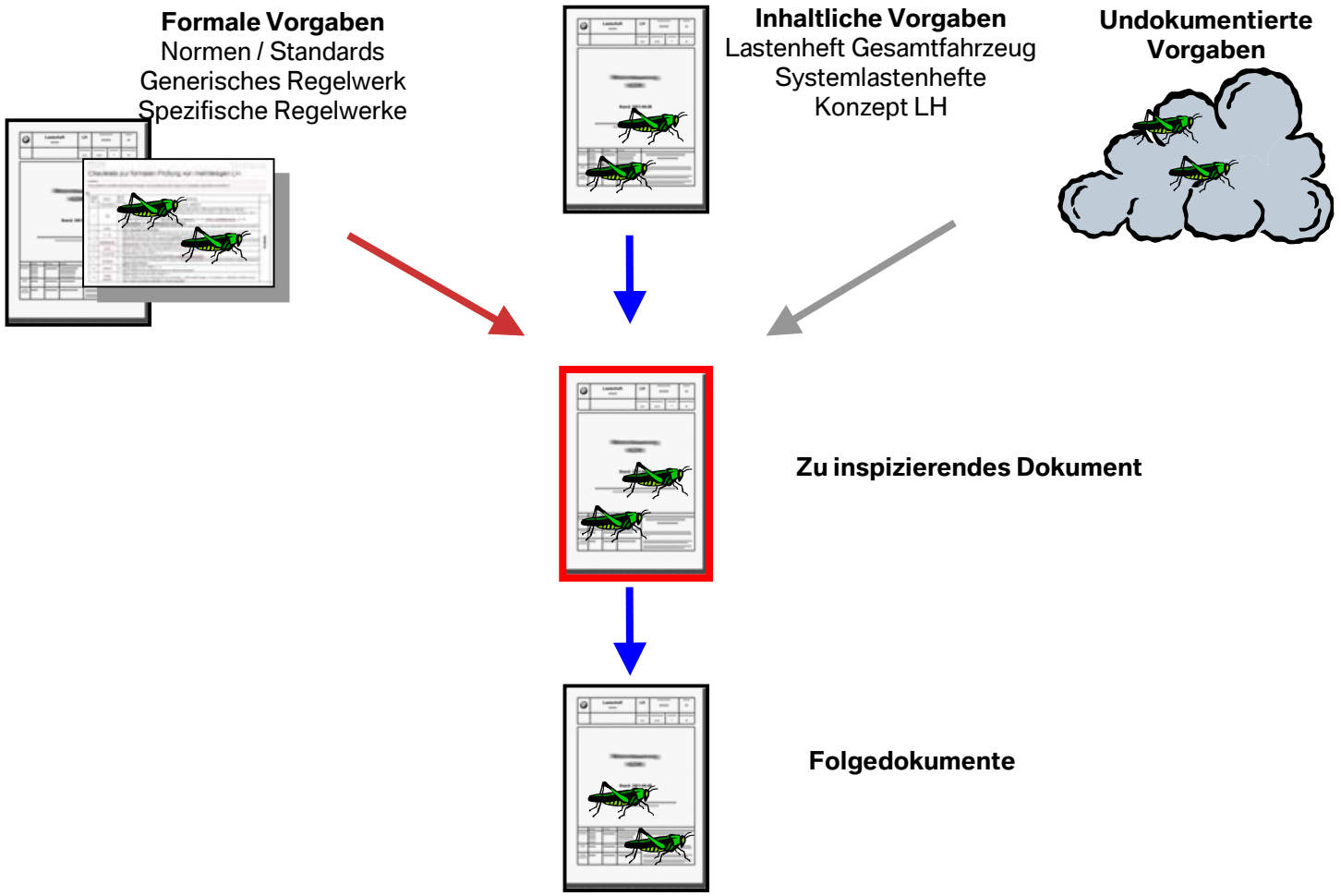




– reduziert Entwicklungszeiten
– wird für sicherheitsrelevante Systeme empfohlen
(IEC 61508 als recommended für SIL3)

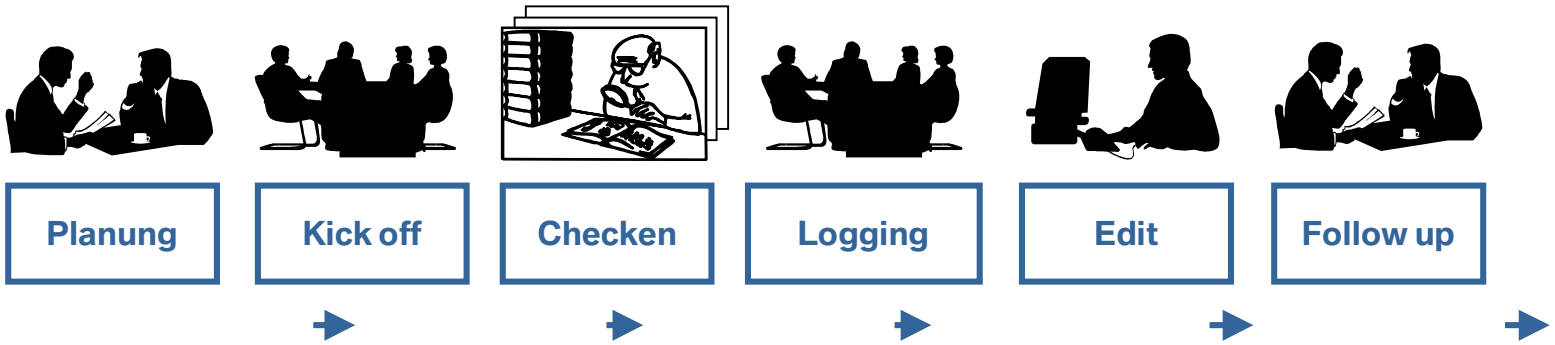
Dr. H. Treseler
BMW AG

Produktorientierte Qualitätssicherung.
Inspektionen.



Quelle: BMW Inspections-Prozess

Produktorientierte Qualitätssicherung.
Inspektionen: Rollen.



Inspektions-Leiter
- Koordination der Aktivitäten
- Dokumentation der Ergebnisse

Autor
- Ersteller bzw. Verantwortlicher des Dokuments
- Prüfer

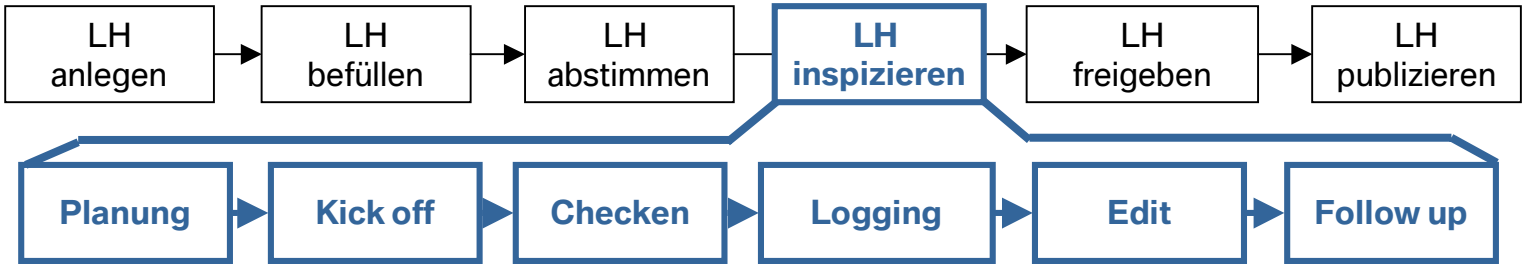
Prüfer
- Überprüfung eines bestimmten Teils des Dokumentes aus einer vereinbarten Sichtweise heraus

Quelle: BMW Inspections-Prozess,
nach Thomas Gilb: *Software Inspections*, 1993.

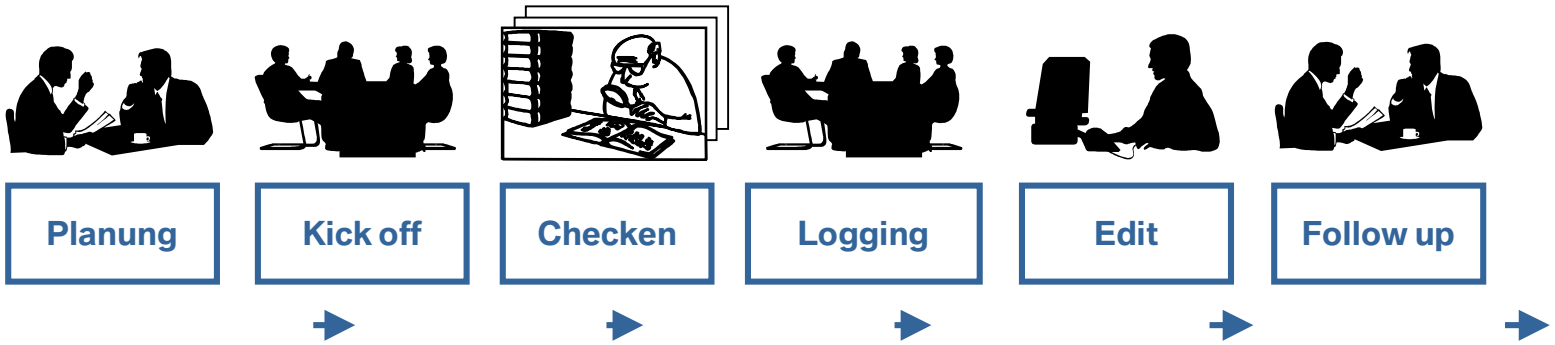
Dr. H. Treseler
BMW AG
28. Juni 2004

Produktorientierte Qualitätssicherung.

Inspektionen: Anwendung im Lastenheft-Prozess.



Produktorientierte Qualitätssicherung.
Inspektionen: Planung.



Einführung

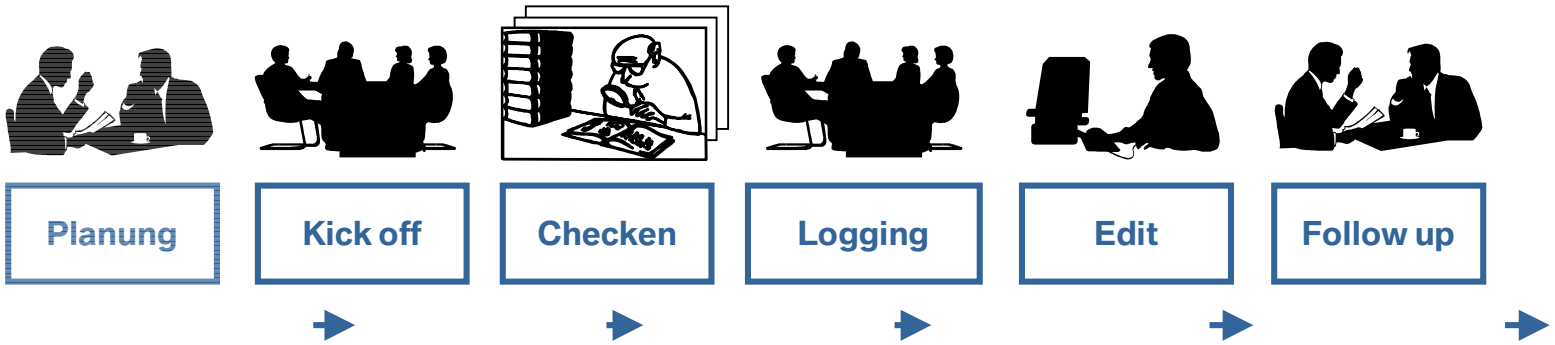
- Briefing (falls Erforderlich)

Planung

- Ziele
- Umfang (relevanter Dokumente, logische Seiten, Quell-Dokumente)
- Teilnehmer
- Vorgehensweise (Sichtweisen der Prüfer, Check-Rate, ...)
- Termine (Kick-off, Logging, Follow-up)

Quelle: BMW Inspections-Prozess,
nach Thomas Gilb: *Software Inspections*, 1993.

Produktorientierte Qualitätssicherung.
Inspektionen: Kick off.



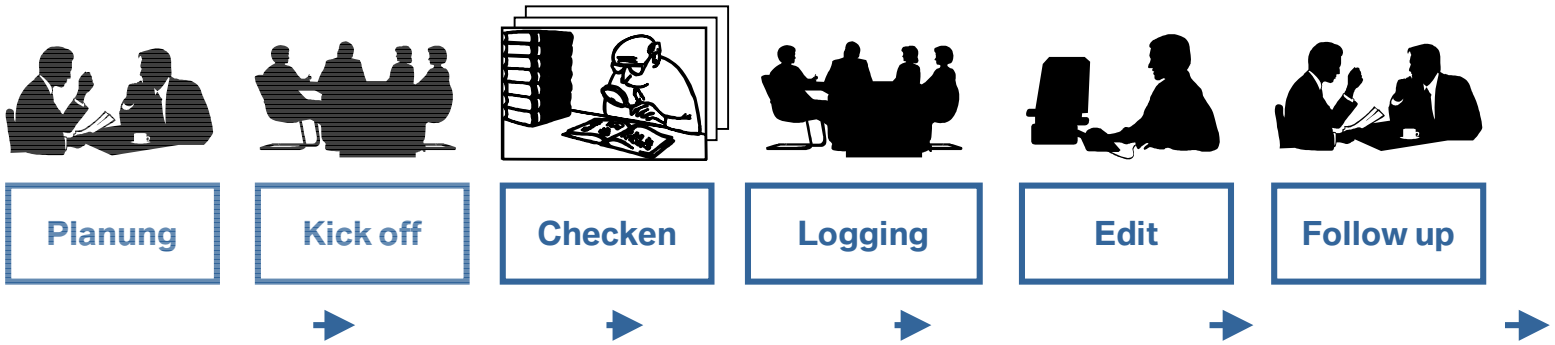
Durchführung
- Einführung (falls erforderlich)
- Erläuterung der Planung

Vereinbarung
- Sichtweisen (für Prüfer)
- Vorgehen im Team (Check-Rate, logische Seiten, ...)
- Termine (Kick-off, Logging, Follow-up)

Verteilung
- Dokumente (Quell-Dokumente, Regeln / Checklisten, Prozeduren)

Quelle: BMW Inspections-Prozess,
nach Thomas Gilb: *Software Inspections*, 1993.

Produktorientierte Qualitätssicherung.
Inspektionen: Checken.



Durchführung

- Individuelles Prüfen des Dokuments

Ziel

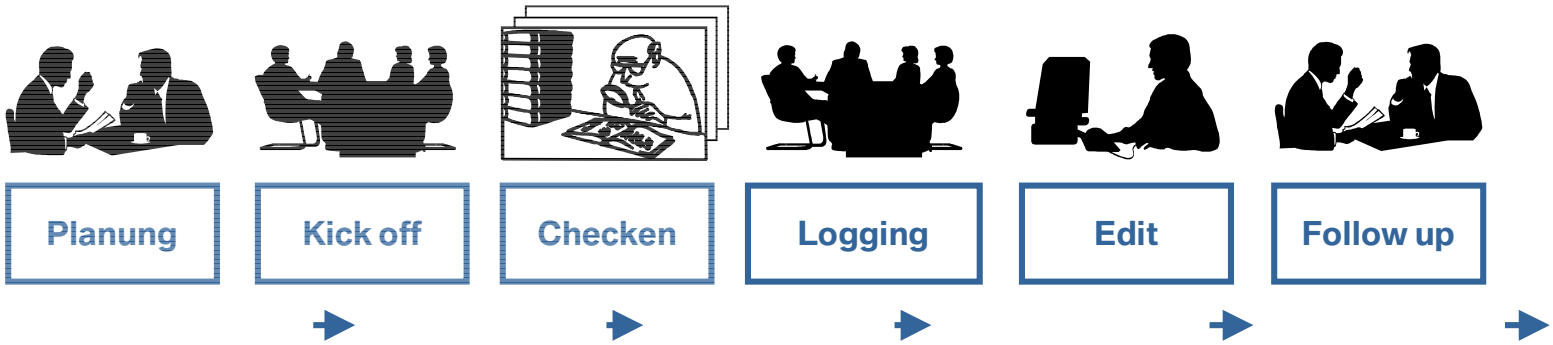
- Finden von
  - potentiell „teuren“ Fehlern
  - sinnvollen Prozessverbesserungen
  - Unklarheiten (Fragen an den Autor)

Spielregeln

- Wertung des Fehlerschadens liegt im Ermessen des Prüfers
- Vereinbartes Verhalten ist einzuhalten

Quelle: BMW Inspections-Prozess,
nach Thomas Gilb: *Software Inspections*, 1993.

Produktorientierte Qualitätssicherung.
Inspektionen: Planung.



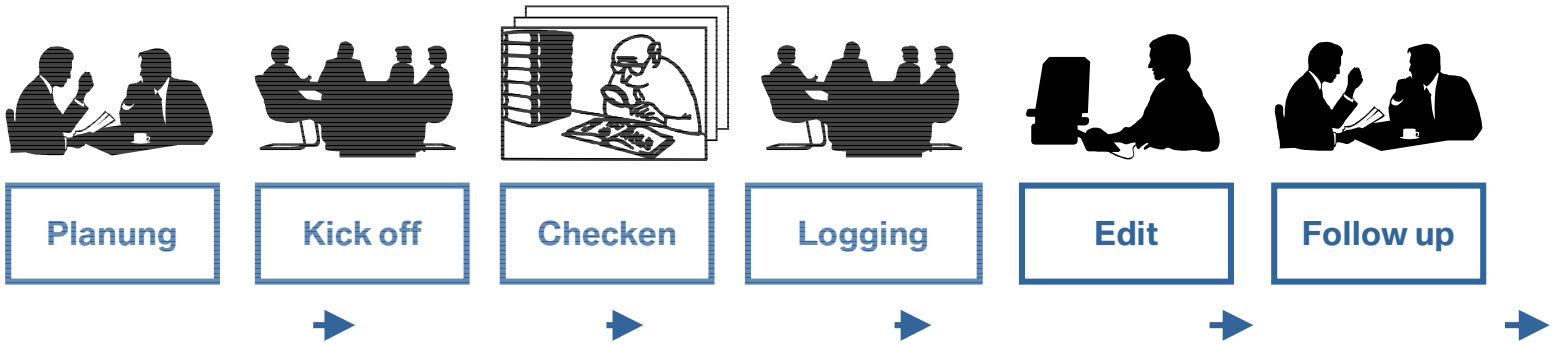
Durchführung

- Systematischen Protokollieren der Ergebnisse

#	Dok. Ref. [...]	Seite / Kap.	Zeile / Stmt No.	Issue Klasse	N V ?	Check-list oder Regel Ref.	Kurze Beschreibung des Issues Benutze Regel-Kodes wo möglich (z. B. TEMPLATE, EXTRA, KONSISTENZ)	# des Auftretens	Autor Notizen beim Editieren
1	1	2	5	SM **Maj** Min	N V ?	GR_DOC25	VOLLSTANDIGKEIT	1	
2	1	2	9	SM **Maj** Min	N V ?	GR_DOC22	KONSISTENZ	1	

Quelle: BMW Inspections-Prozess,
nach Thomas Gilb: *Software Inspections*, 1993.

Produktorientierte Qualitätssicherung.
Inspektionen: Bearbeiten.



Durchführung

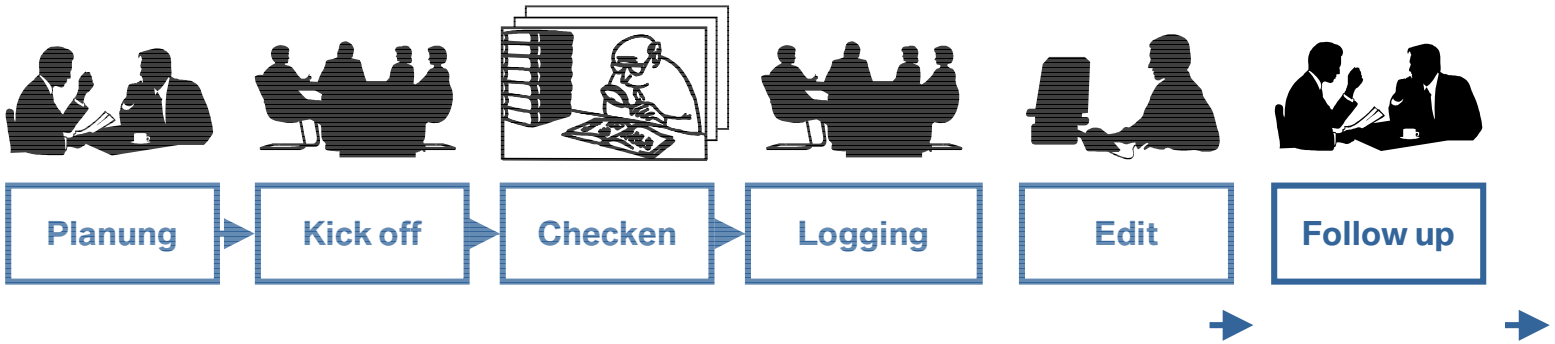
- Bearbeitung der berichteten Probleme und Korrektur der Fehler

Spielregeln

- Autor darf werten, ob Probleme Fehler sind oder nicht
- Autor muss Wertung in der Dokumentation festhalten
- Autor muss erkannte Fehler korrigieren

Quelle: BMW Inspections-Prozess,
nach Thomas Gilb: *Software Inspections*, 1993.

Produktorientierte Qualitätssicherung.
Inspektionen: Follow-up.



Durchführung

- Stichprobenartige Überprüfung der Korrekturen
- Aufnahme der Metriken
  - Gesamt-Zeitaufwand
  - Anzahl der (von Autor akzeptierten) Fehler
- Follow-Up-Fragebogen
  - zur Verbesserung des Inspektions-Prozesses
- Individuelles Prüfen des Dokuments

Quelle: BMW Inspections-Prozess,
nach Thomas Gilb: *Software Inspections*, 1993.

Produktorientierte Qualitätssicherung.
Methoden.

- Konstruktive Maßnahmen:
  - Erstellung von Vorgaben an Software-Entwicklungsprozesse,
  - Beispiel: ISO 12207, IEC 61508.

- Analytische Maßnahmen:
  - Überprüfung der Software-Entwicklungsprozesse („Software-Assessments), v. A. bei Zulieferern,
  - ISO 15504.

Prozessorientierte Qualitätssicherung.
Standards zur Software-Entwicklung.

Reifegrad-Standards / Qualitätsstandards:

- CMMI: Bewertung durch Level, sehr generisch, Beschreibung was zu geschehen hat, aber nicht wie es zu geschehen hat
- ISO 15504: Level direkt vergleichbar mit CMMI

Entwicklungsstandards:

- ISO 12207 AMD 1: international, sehr generisch, Detaillierung erforderlich
- V-Modell 97: national, konkret, anleitend, beschreibt Dokumente
- VDA AK 13: gute Definitionen, informativ, unverbindlich

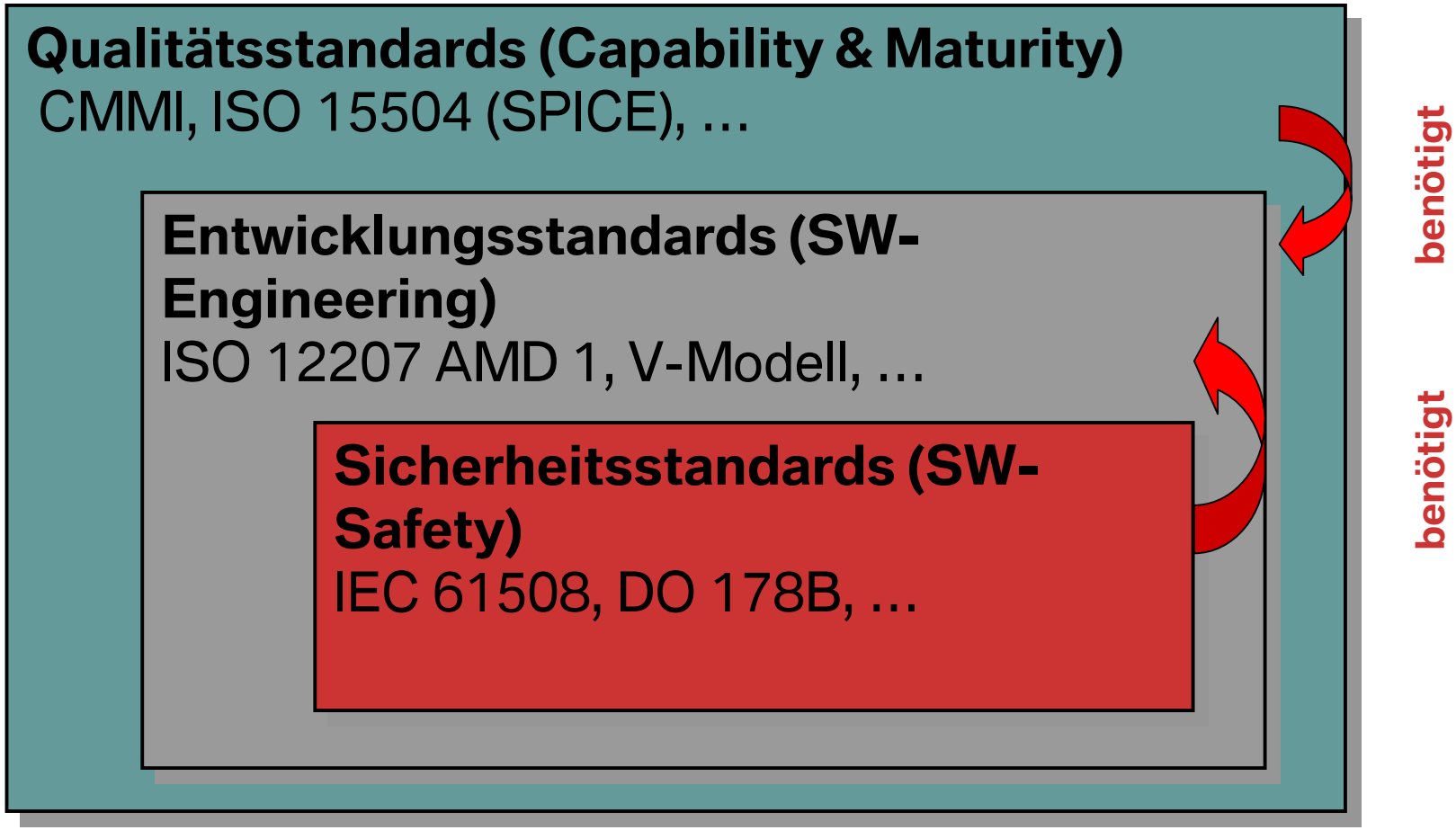
Sicherheitsstandards:

- IEC 61508: weite Verbreitung, Detaillierung erforderlich
- MISRA Guidelines: Spezifika des Automobilsektors
- RTCA DO-178B: konkret, präziser als IEC 61508, luftfahrtspezifisch
- EN 50128: pragmatisch, benutzerfreundlich
- Def Stan 00-55: klares Rollenkonzept

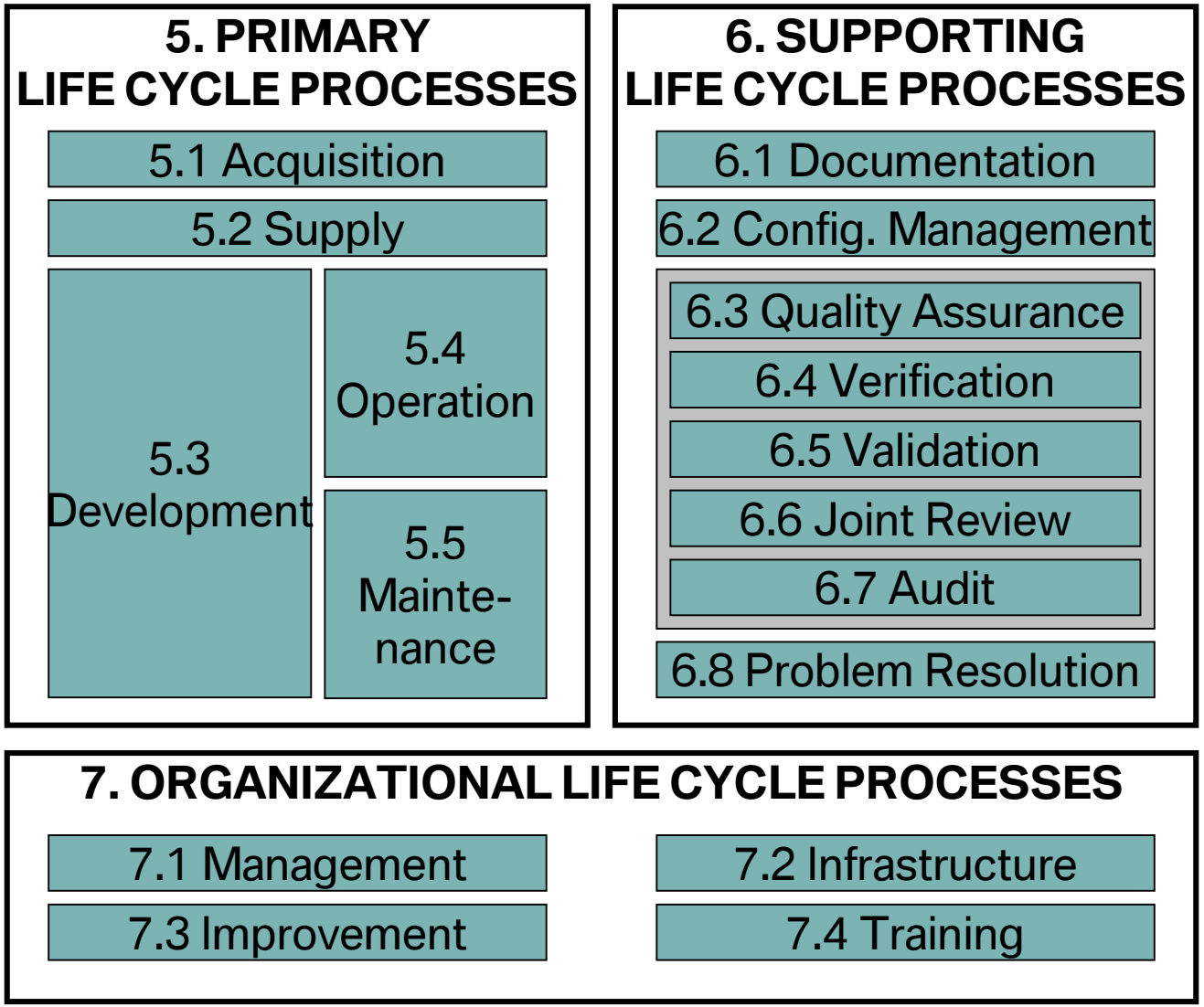
Dr. H. Treseler
BMW AG

Prozessorientierte Qualitätssicherung.

Beziehungen der Standards.



Prozessorientierte Qualitätssicherung.
ISO 12207 „Software-Lebenszyklus“.

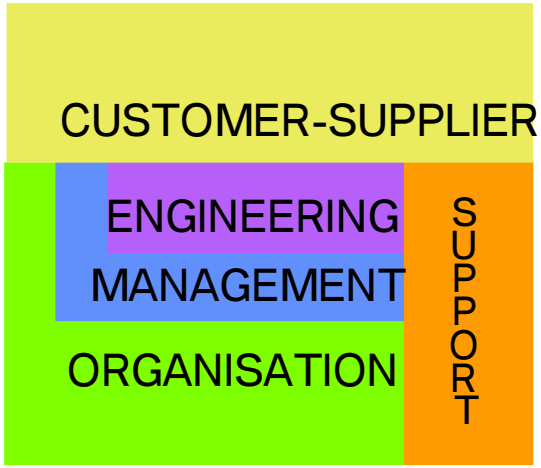


Prozessorientierte Qualitätssicherung.
Bewertung von Entwicklungsprozessen.

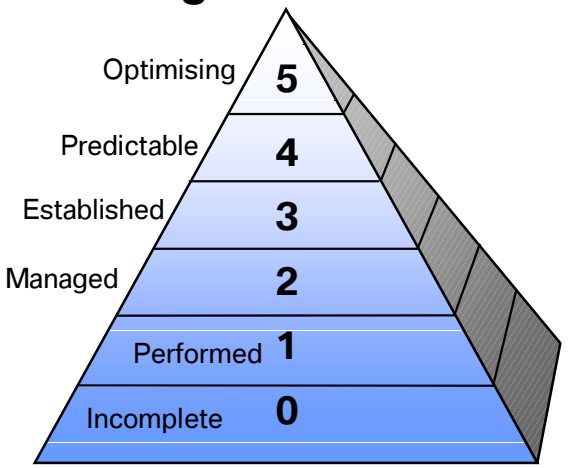
- Software-Entwicklungsstandards sind die Voraussetzung:
  - SPICE (ISO/IEC 15504) als Referenzmodell
  - IEC 61508 für sicherheitskritische Software

- Das Referenz-Modell ist zweidimensional:
  - Prozess-Dimension (nach ISO/IEC 12207): enthält Prozess in Gruppen
  - Reifegrad-Dimension: erlaubt die Bewertung der Reife der Prozesse unabhängig voneinander

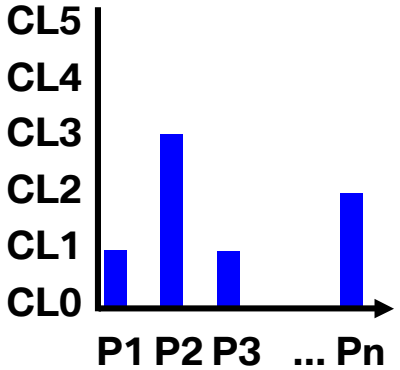
Prozess-Dimension



Reifegrad-Dimension



Prozess-Profile



Dr. H. Treseler
BMW AG
28. Juni 2004

Prozessorientierte Qualitätssicherung.

SW-Assessments: Bewertung nach Prozessattributen (Beispiel).

	Reifegrad	1	2		3		4		5	
	Prozessattribut	PA 1.1	PA 2.1	PA 2.2	PA 3.1	PA 3.2	PA 4.1	PA 4.2	PA 5.1	PA 5.2
CUS.1.3	**Lieferantenüberwachung**	L	L	L						
ENG.1.1	**Systemanforderungsanal. und –entw.**	L	L	L						
ENG.1.2	**Software-Anforderungsanalyse**	F	F	F	P	N				
ENG.1.3	**Software-Entwurf**	L	L	L						
ENG.1.4	**Software-Erstellung**	P	P	N						
ENG.1.5	**Software-Integration**	L	L	P						
ENG.1.6	**Software-Test**	F	F	L						
ENG.1.7	**Systemintegration und –test**	P	P	P						
MAN.2	**Projektmanagement**	L	L	L						
SUP.2	**Konfigurationsverwaltung**	P	P	N						
SUP.3	**Qualitätssicherung**	P	P	N						

Reifegrad:
0: unvollständig
1: ausgeführt
2: beschrieben
3: etabliert
4: vorhersehbar
5: optimiert

PA 1.1: Prozessdurchführung
PA 2.1: Prozessplanung
PA 2.2: Resultate/Arbeitserzeugnisse
PA 3.1: Prozessbeschreibung
PA 3.2: Prozessressourcen
PA 4.1: Metriken
PA 4.2: Prozesssteuerung
PA 5.1: Prozessänderung
PA 5.2: Prozessverbesserung

F: vollständig (fully)
L: weitgehend (largely)
P: teilweise (partially)
N: nicht ausreichend
[ ]: nicht zutreffend

SW-Qualitätssicherung.
Zusammenfassung.

- Ein übergreifendes Konzept der Qualitätssicherung mit Aktivitäten in jeder Entwicklungsphase ist erforderlich:
  - Auswahl von analytischen und konstruktiven Maßnahmen,
  - Auswahl von prozessorientierten und produktorientierten Maßnahmen.
- Um eine Kosteneinsparung zu erreichen, muss die Qualitätssicherung in frühen Entwicklungsphasen verstärkt werden (= konstruktive Maßnahmen).
- Miteinbeziehung von SW-Lieferanten in die Qualitätssicherung.

SW-Qualitätssicherung.
Literatur.

– Wallmüller, Ernest: *Software-Qualitätssicherung in der Praxis*. München [u.a.]: Hanser, 1990, Neuauflage 2001.
– CMMI *"Capability Maturity Model ® Integration (CMMI^{SM})"* Version 1.1", CMU SEI, 2002.
– Weiterführendes zum CMM: Thaller, G.E.: *Qualitätsoptimierung der Softwareentwicklung - Das Capability Maturity Model.* Braunschweig/Wiesbaden: Vieweg 1993.
– ISO 12207 *"Information Technology - Software lifecycle processes"*, 1st Edition, Aug. 1995.
– V-Modell *„Entwicklungsstandard für IT-Systeme des Bundes, Vorgehensmodell"*, Stand 1997.
– Martin Pol et al.: *Management und Optimierung des Testprozesses.* dpunkt.verlag, Heidelberg, 2002.
– ISO/IEC TR 15504 *"Information technology -- Software process assessment"*, 1998-2003.

Vielen Dank für Ihre Aufmerksamkeit.