

Reinhard Jeschull, Dr. Thomas Flaig

Validierung einer MATLAB-Toolkette

Notwendiges Übel oder Allheilmittel?

2. Juli 2019

Agenda



- ▶ Die Validas AG
- ▶ Grundlagen und Gründe für Validierung
- ▶ Beispiele von Fehlern
- ▶ Validierung einer MATLAB-Toolkette
- ▶ Fazit

Agenda



- ▶ Die Validas AG
- ▶ Grundlagen und Gründe für Validierung
- ▶ Beispiele von Fehlern
- ▶ Validierung einer MATLAB-Toolkette
- ▶ Fazit

Wer wir sind



Dr. Oscar Slotosch
(Vorstand)



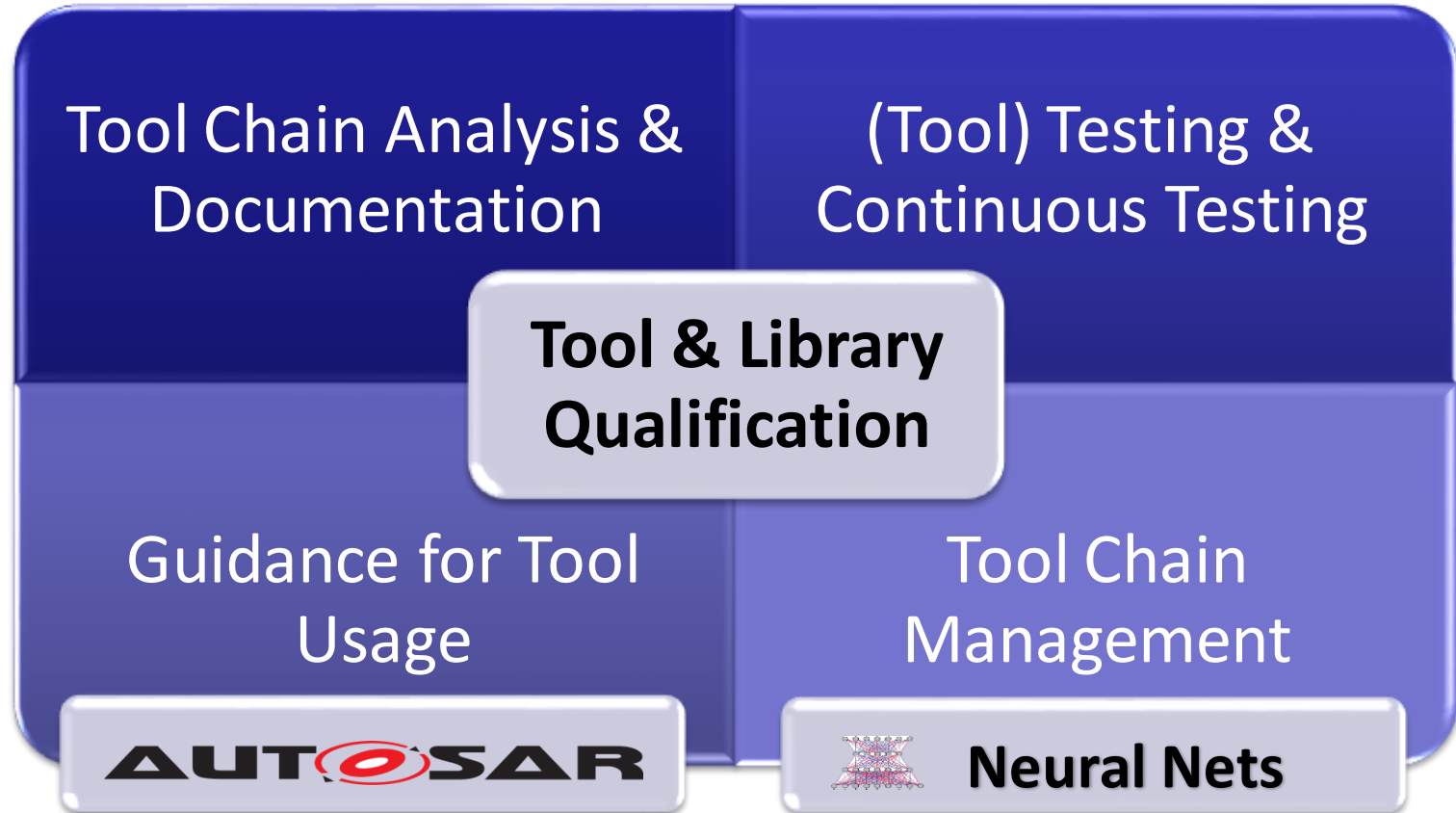
Dr. Peter Braun
(Vorstand)

- ★ Seit April 2000 Technologiedienstleister für eingebettete Systeme
- ★ Kernkompetenzen: Modellbasierung, **Werkzeugqualifizierung**, Methodenberatung



- ★ Ca. 40 Mitarbeiter und Studenten aus unterschiedlichen Disziplinen
- ★ Skalierbarkeit durch Netzwerk aus Partnerfirmen und Freiberuflern

Woran wir arbeiten



Agenda



- ▶ Die Validas AG
- ▶ Grundlagen und Gründe für Validierung
- ▶ Beispiele von Fehlern
- ▶ Validierung einer MATLAB-Toolkette
- ▶ Fazit

Müssen Tools sicher sein?



Ja, wenn ...

- Safety Standards dies verlangen (Automobil, Eisenbahn, Luftfahrt, etc.)
 - § 4 I GPSG - EU Guideline 2001/95/EG fordert Einhaltung von „harmonized standard“ (de facto ISO26262 für Automobil, etc.)
- Tool-Anwender es verlangen
- Tool-Hersteller hat den Wunsch nach Qualität

Tool
Failure



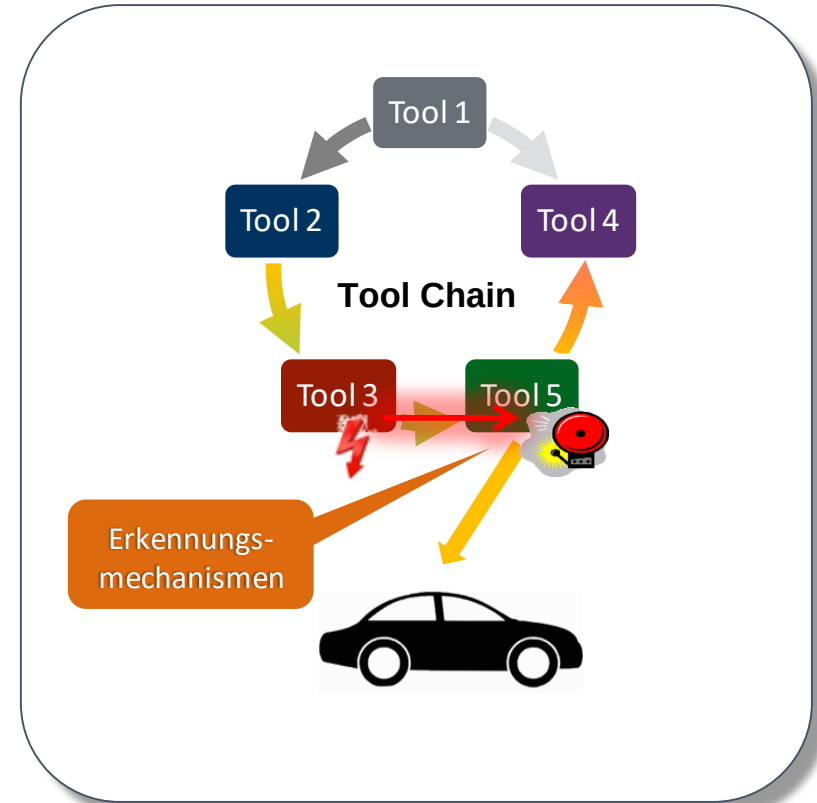
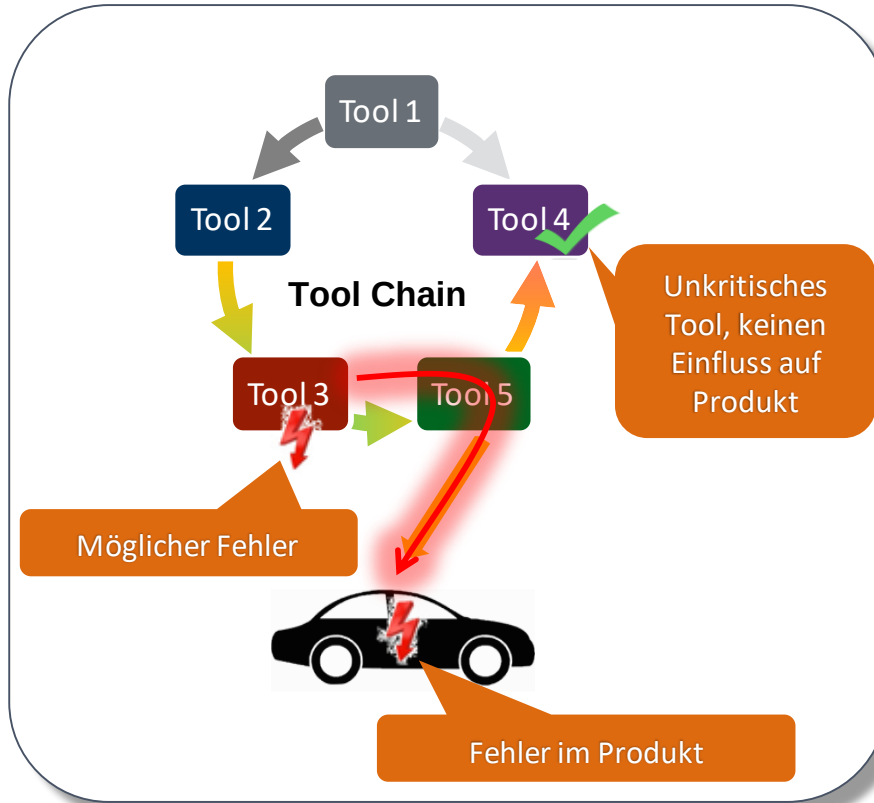
Product
Fault



Validierung von Toolketten



Tools sind nicht Teil des Produktes, haben aber Einfluss



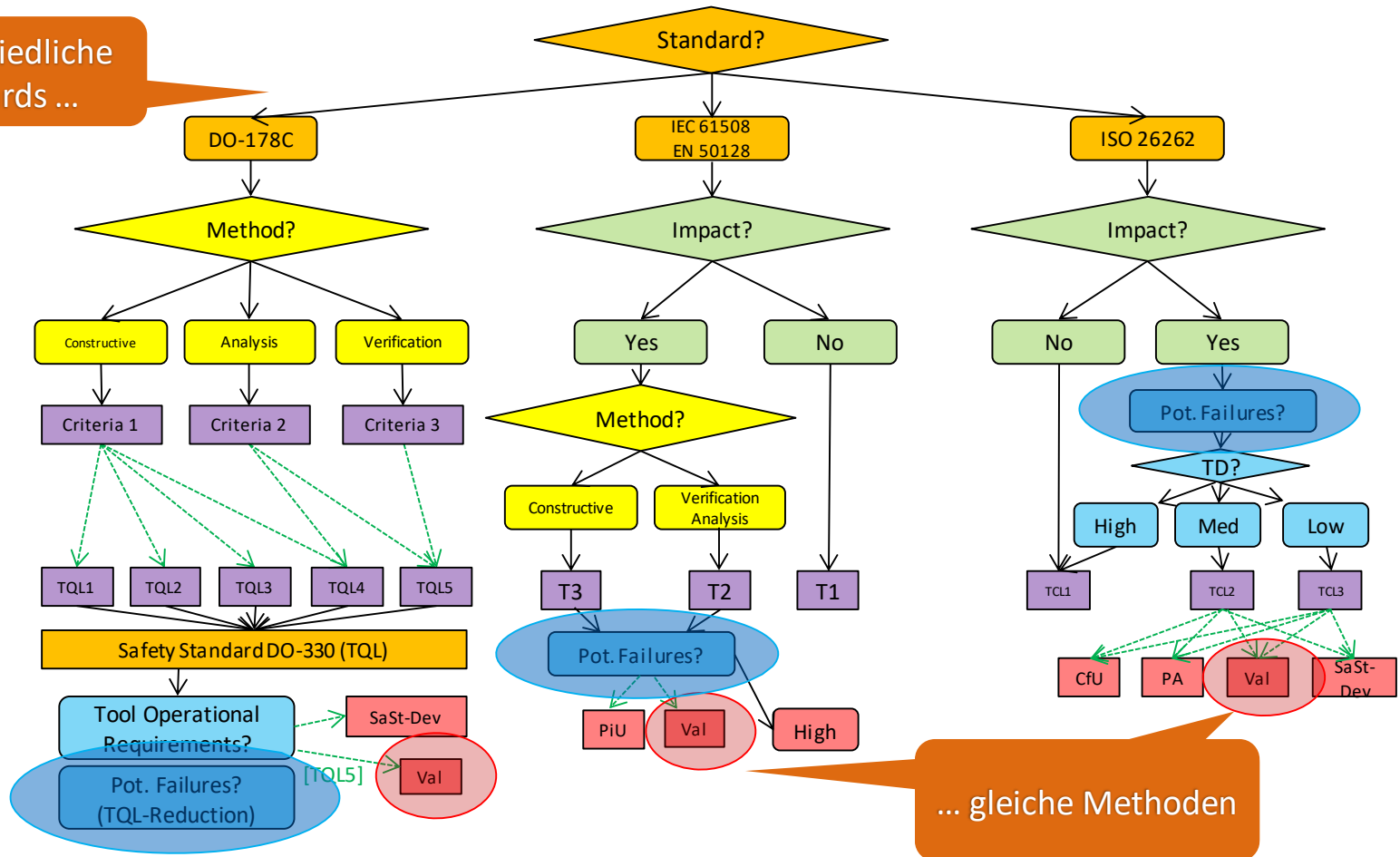
Validierungen in Standards



Unterschiedliche Standards ...

Classification/Evaluation

Qualification



Agenda



- ▶ Die Validas AG
- ▶ Grundlagen und Gründe für Validierung
- ▶ Beispiele von Fehlern
- ▶ Validierung einer MATLAB-Toolkette
- ▶ Fazit

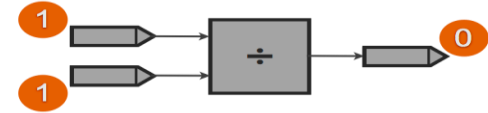
Beispiele von Fehlern



```
int8 x = -3;  
Out = abs(x);
```

Expected: **3**
Got: **-3**

Compilerbug bei
Absolutbetrag



Bug in Interpreter für If-Anweisung bei Template Engine

```
"C = $c"  
if $c then  
  "ok"  
else  
  "not ok"  
endif
```

C = true

Expected: **„ok“**
Got: **„not ok“**

Unerwartete Optimierung „zu Gunsten der Laufzeit“

```
int32_T qY;  
qY = ln1 + 1;  
if ((ln1 > 0) && (qY <= 0)) {  
  qY = MAX_int32_T;  
}  
Out1 = qY;
```

ln1 = 2147483647
(MAX_int32_T)

Expected: **2147483647**
Got: **-2147483648**

Fehler im Mathebibliothek

```
sqrt(x)
```

Expected: **Value**
Got: **Endless Loop**

Compiler optimiert Saturierung weg, da Overflow in C
„undefined behavior“

```
c:a.exe  
-2319.1619
```

Compiler patch level **x**

```
c:a.exe  
-24.384871
```

Compiler patch level **y**

Starke Abweichungen nach einem
Update (Patch-Level)

Agenda



- ▶ Die Validas AG
- ▶ Grundlagen und Gründe für Validierung
- ▶ Beispiele von Fehlern
- ▶ Validierung einer MATLAB-Toolkette
- ▶ Fazit

Die Validierungssuite



Ziele ✓:

- Erfüllung der Safety-Anforderungen (z.B. ISO26262)
- Entdeckung vieler durch Tools injizierter Produkt-Fehler
- Reduktion der Fehleranalyse in Produkttests
- Vermeidung von injizierten Produkt-Fehlern durch geeignete Gegenmaßnahmen
- Reduktion von Reibungsverlusten bei Entwicklung
- Bereitstellung validierter Werkzeuge für die modellbasierte SW-Entwicklung
- Erhöhung des Vertrauens in Tool(-kette)

Die Validierungssuite



Nicht-Ziele ✗

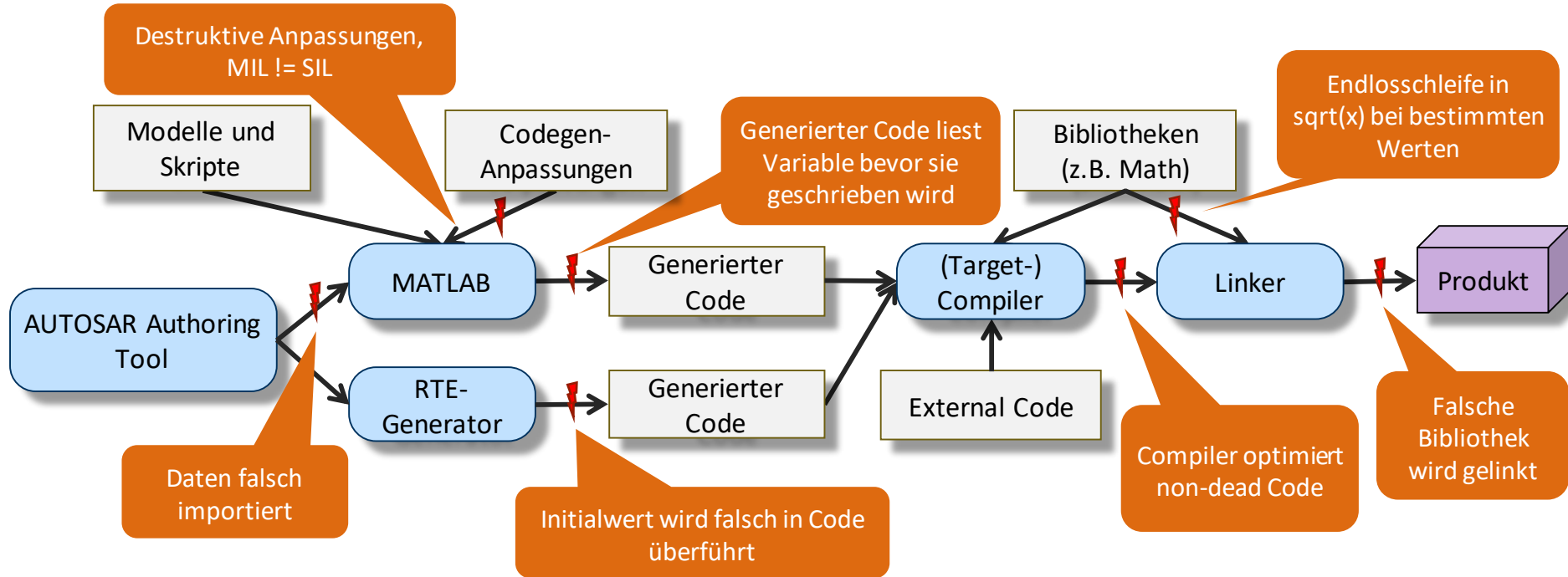
- Erspart explizit nicht die Absicherungsmaßnahmen für das entwickelte Produkt!

Die Validierungssuite prüft die Toolkette, aus der beliebig viele Produkte erzeugt werden können. Die Erfüllung der Anforderungen an das konkrete Produkt sind separat zu prüfen (Produkttest).

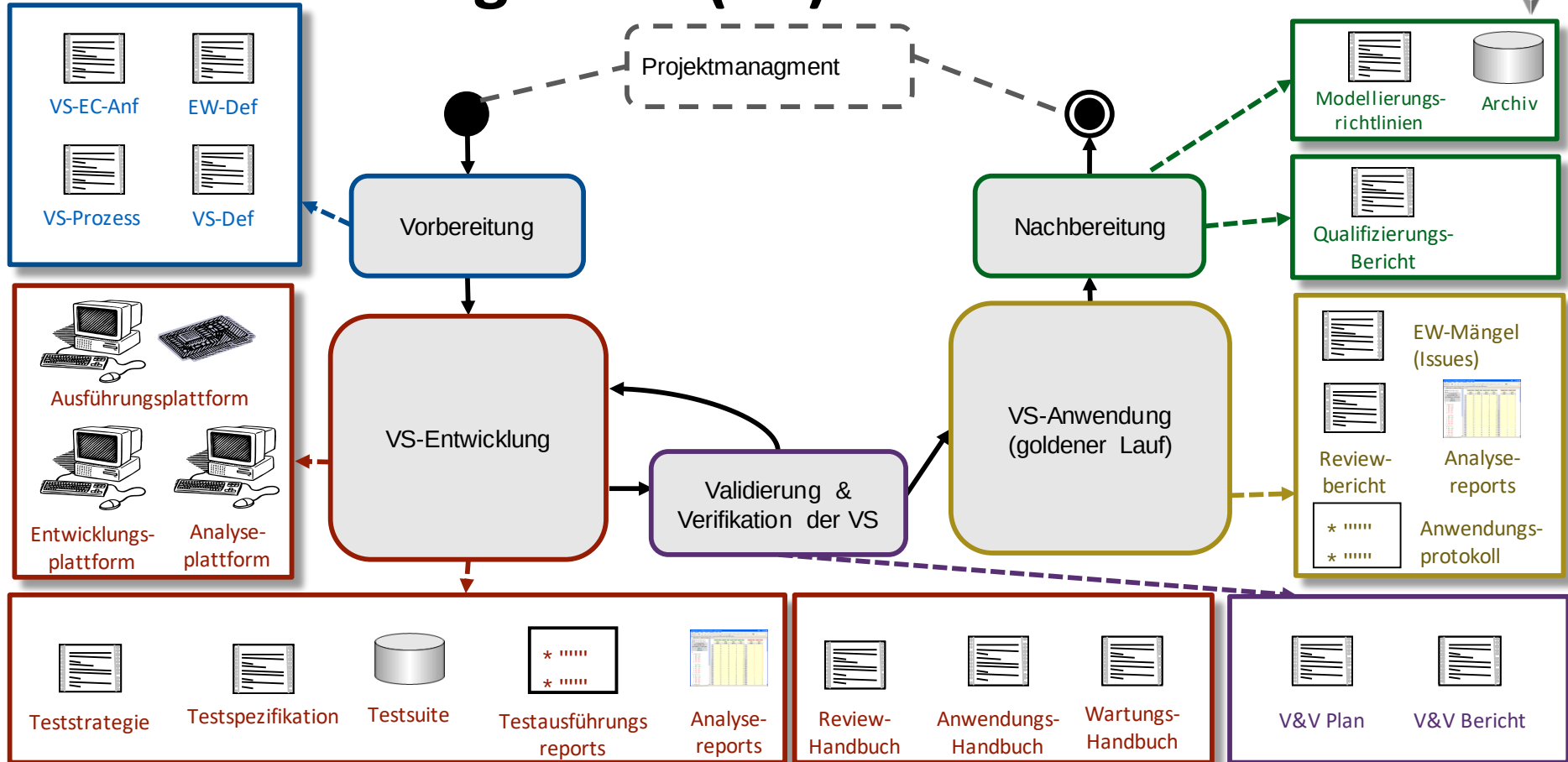
Absicherung Toolkette != Absicherung Produkt

- Validierung ist notwendiges Indiz für das Vertrauen in das Werkzeug, aber keine Garantie für Fehlerfreiheit des entwickelten Produkts
- Beispiel: Numerische Stabilität des Produkts (Targetverhalten) nicht durch Validierung garantiert!

Beispiel einer einfachen Toolkette



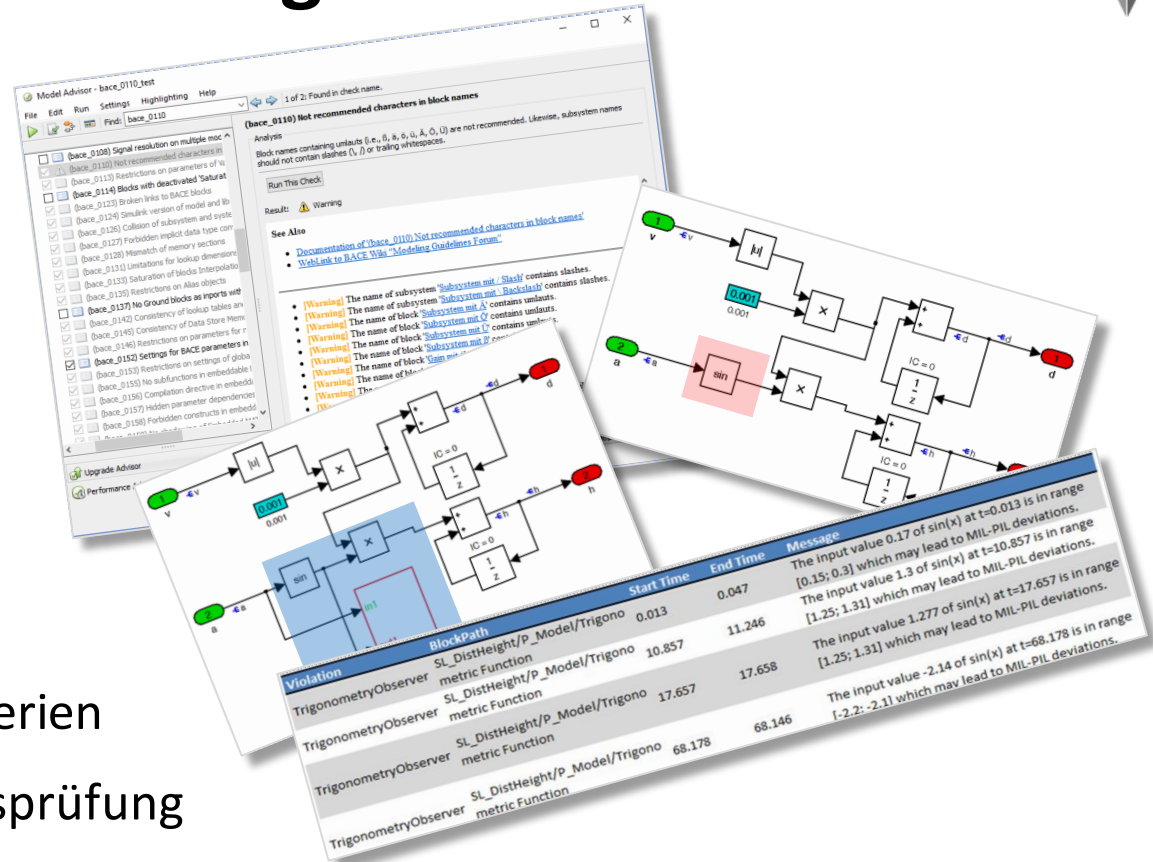
Die Validierungssuite (VS)





Ergebnisse der Validierungssuite

- ▶ Teilqualifizierungsbericht
- ▶ Dokumentierte Findings
- ▶ Vermeidungsstrategien
 - Richtlinien (statisch)
 - Manuelle Reviews
- ▶ Erkennungsstrategien
 - Manuelle Reviews
 - Automatisierte Prüfkriterien
 - dynamische Simulationsprüfung
- ▶ Vorschläge für Anpassungen an Toolkette



Agenda



- ▶ Die Validas AG
- ▶ Grundlagen und Gründe für Validierung
- ▶ Beispiele von Fehlern
- ▶ Validierung einer MATLAB-Toolkette
- ▶ Fazit

Fazit



► **Notwendiges Übel? => Nein**

↗ Validierung bietet eine standardkonforme Absicherung der Entwicklungswerkzeugkette

► **Allheilmittel? => Jein**

↗ Durchführung des Validierungssuite-Prozesses mehrfach vom TÜV Nord abgesichert

↗ Entdeckung vieler durch Tools injizierter Produkt-Fehler

↗ Vermeidung von injizierten Produkt-Fehlern durch geeignete Gegenmaßnahmen

⇒ Erspart explizit nicht die Absicherungsmaßnahmen für das entwickelte Produkt

⇒ Voraussetzung: Toolkette wird, wie definiert, eingesetzt

⇒ Entwicklungsbegleitende Tests zur Absicherung bestimmter Fehlerquellen (-> numerische Stabilität) weiterhin notwendig



Die Validas AG ist Ihr Innovationspartner für die Einführung und Evaluierung innovativer Methoden, Technologien und Werkzeuge.

Die Expertise der Validas umfasst Modellbasierung (UML/SysML, MATLAB Simulink/Stateflow, AUTOSAR), Funktionssicherheit und Werkzeugqualifizierung (ISO 26262, ISO 61508, DO 178B/C) sowie Softwarequalität (Testautomatisierung, modellbasierte Tests, Code-Analysen).

Zu unseren Kunden zählen renommierte Firmen wie Airbus Defence and Space, Audi, BMW, BOSCH, Carmeq, Continental, DENSO, ESG, ETAS, Infineon, Lear, MAN, TechConnect, Texas Instruments, Valeo, Wind River.

Validas AG
Arnulfstraße 27
80335 München
Fon +49 (0) 89-53 88 669-0
Fax +49 (0) 89-53 88 669-10
info@validas.de
www.validas.de

