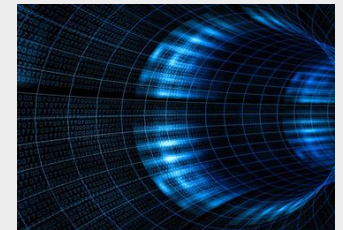


Kontinuierliche Integration auf Basis einer FMEA

Dr.-Ing. Markus A. Stulle
IFS Informationstechnik GmbH



- Qualität eines Softwareprodukts
- Systemtests und kontinuierliche Integration (KI)
- FMEA und Risikoanalyse mit Prometheus
- Systemtests im Testrahmenwerk Anubis der Serviceplattform Taurus®
- Ein bewährtes Verfahren
- Mobile Anwendungen





- **Anwendungsfälle** sind vollständig erfasst und werden fehlerfrei befriedigt
- Anwender erfahren positives **Nutzungserlebnis** („pride of ownership“)
- Produkt zeigt **Effizienz** in Betrieb und Wartung
- Produkt bietet und behält **Alleinstellungsmerkmale** im Wettbewerb!

Speed is the essence!

- **Fehlerfreie Implementierung**
nicht hinreichend zur Erzielung hoher Qualität
- Softwareprodukt:
Schnelle Bereitstellung von Funktionshub
und Fehlerbehebungen primäres Ziel!
- Qualitätssicherung im Zeitalter
agiler Methoden und hohem Innovationsdruck
ist ein Entwicklungsthema

- Implementierung einer Softwarekomponente **F** durch ausgezeichnete Funktionsaufrufe $f(\mathbf{x})$ testen
 - Testsituationen **T**: Funktionsargumente \mathbf{x} müssen Wertebereich möglichst gut abdecken
 - Entwicklung von **Attrappen** aufwändig, Vergleich von erwartetem und tatsächlichem Funktionsergebnis erfordert Wissen und Fähigkeiten des Entwicklers von **F**
- Im bewährten Verfahren erfolgt Entwicklung von Komponente und Test in **disjunkten** Gruppen



- **Anwendungsfall** resultiert aus der Anforderungsanalyse (Domäne)
- **Systemtestfall** erlaubt Substitution von Komponenten des Produkts (keine „black box“)
- Ausführung in reproduzierbarer **Testkonstruktion**
 - Prüfling, Plattform, Daten
 - Ort und Zeit
 - Bediener



- wird von **Prozesswerkzeug** gesteuert
- rüstet Testkonstruktion, führt Systemtestfall aus, wendet **Testkriterien** für Fehlerbilder an
- bietet **Nachbildungen**
 - für Rollen des Bedieners
 - für Komponenten der Zielplattform
 - für Teile der Umwelt (Störgrößen)
- **Beispiele:** Anubis, Echnaton.

- berücksichtigt die **nebenläufige** Programmausführung in Verbundanwendungen
- ermöglicht frühe **Machbarkeitsnachweise** und Einsicht in das Nutzungserlebnis
- unterstützt testgetriebene Entwicklungsparadigmen
- unabdingbare **Vorbedingung:** geeignete Architektur der Anwendung!



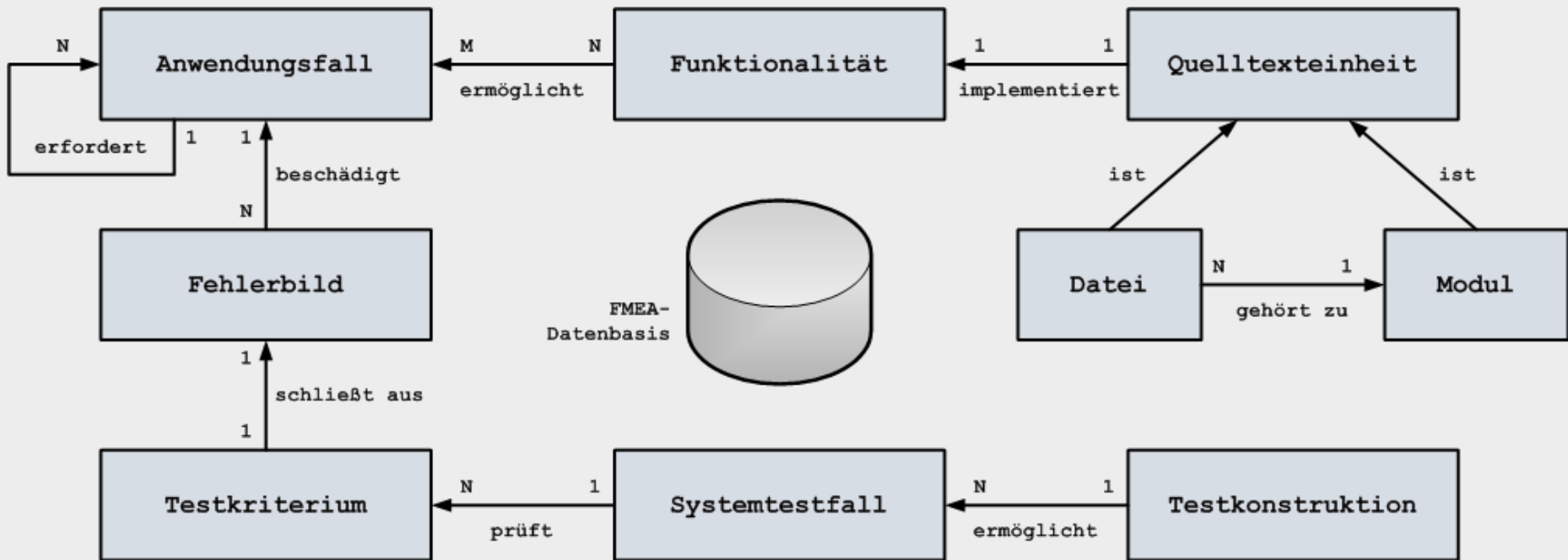
- **Ziel**
Ursache und Verursacher von Fehlern erkennen!
- **Verfahren**
Möglichst kleine Evolutionsschritte,
zeitnahes Einbuchen von Änderungen auf Stämmen
- **Merkmale**
 - Vor jedem Einbuchen: Produktion, Installation, **Systemtest**
 - Stets auslieferungsfähiger Stand des Produkts,
ohne Defekt, aber nicht unbedingt fehlerfrei.

- Definition eines **Integrationsprodukts**
- **FMEA-Datenbasis**
- Prozesswerkzeug
 - für Pflege der Datenbasis und **Risikoanalyse**
 - automatische **Produktion, Distribution** und **Installation** des Produkts („delivery“)
 - automatische Auswahl, Ausführung und Auswertung von **Systemtests** in der Zielumgebung



- **FMEA klassifiziert Fehlerbilder**
 - Kennzahlen für Schweregrad (S),
Wahrscheinlichkeit (W) und Verborgenheit (V)
 - Risiko-Prioritätszahl (RPZ) := $S*W*V$
wird in Kiviat-Diagrammen logarithmisch dargestellt.
- **FMEA-Datenbasis** eines Produkts
 - Entitäten Anwendungsfall, Funktionalität,
Quelltexteinheit (Fachprojekt),
Fehlerbild, Testkriterium, Systemtest.

FMEA-Datenbasis



Anwendungsfall bearbeiten

Ein Anwendungsfall beschreibt ein Szenario, das im Betrieb eines Produkts in dessen Anwendungsdomäne auftreten kann. Anwendungsfälle werden im Lastenheft zusammengefasst und von Grundfunktionen befriedigt.

Bezeichner (Nomen)	Bezeichner (Verb)	Spezialisierer (Nomen)	Spezialisierer (Verb)
Fahrzeugsitzung	durchführen	Fahrzeugzugang	umschalten

Bezeichner (Nomen)	Bezeichner (Verb)	Spezialisierer (Nomen)	Spezialisierer (Verb)
Fahrzeugsitzung	durchführen	Fahrzeugzugang	umschalten

Beschreibung
 Fahrzeugzugang umschalten

Anwendungsfall-ID: A17135

Art des Anwendungsfalls: konkret

Wichtigkeit: 5 (= durchschnittlich)

Übergeordneter Anwendungsfall: Argonaut

Integrationsprodukte: Argonaut

Name: Befehlsverarbeitung im Fahrzeugzugangsgesät (VCI) scheitert

Beschreibung: Umschaltbefehl wird zwar erfolgreich an das Fahrzeugzugangsgesät (VCI) übermittelt, die Verarbeitung erfolgt jedoch nicht, oder bleibt ohne Auswirkung

Wahrscheinlichkeit = 4

10 - so gut wie sicher

9
8
7
6
5
4
3
2
1 - kaum vorstellbar

Schweregrad = 7

10 - Anwendung unbrauchbar

9
8
7
6
5
4
3
2
1 - ohne Auswirkung

Verborgtheit = 8

10 - nicht erkennbar

9
8
7
6
5
4
3
2
1 - sicher erkennbar

Wichtigkeit: 5 (= durchschnittlich)

Wichtigkeitsskala: 1 2 3 4 5 6 7 8 9 10

Funktionalitäten:

- ISOM/L-Programme für den Mandanten BMWREF
- Fachdienstproxies für die ISOM-Anbindung
- Ifh-Slave SGEmu stellt die Verbindung zwischen EDIABAS und IFS/Virgo/Application her.
- OEM-abhängiger Teil des Fahrzeugzugangs zu Virgo

- **Risiko** durch Änderungen an Quelltexteinheiten
 - Ermittle die möglichen Fehlerbilder und RPZ, vergleiche Alternativen!
- Welche **Systemtests** sind erforderlich?
 - Wähle so viele Testfälle aus, dass deren Testkriterien alle möglichen Fehlerbilder abdecken.
- Nutzen des **Abhängigkeitsgraphen**
 - Wenn abhängige Anwendungsfälle mittelbar beschädigt: Abbrechen der Systemtests.

1. **Thema** definieren

für Fehlerbehebung oder Funktionshub

- Alle Änderungen an Quelltexteinheiten (QTE) erfolgen innerhalb des Themas!

2. **Risiko** analysieren

- Summiere RPZ aller möglichen Fehlerbilder im Abhängigkeitsgraph der Anwendungsfälle

3. **Implementierung** durchführen

- Auf Bezug zum Thema im Quelltext achten!

4. Produktion der abhängigen Gebinde

- Fachprojekte erzeugen Funktionalitäten aus QTE

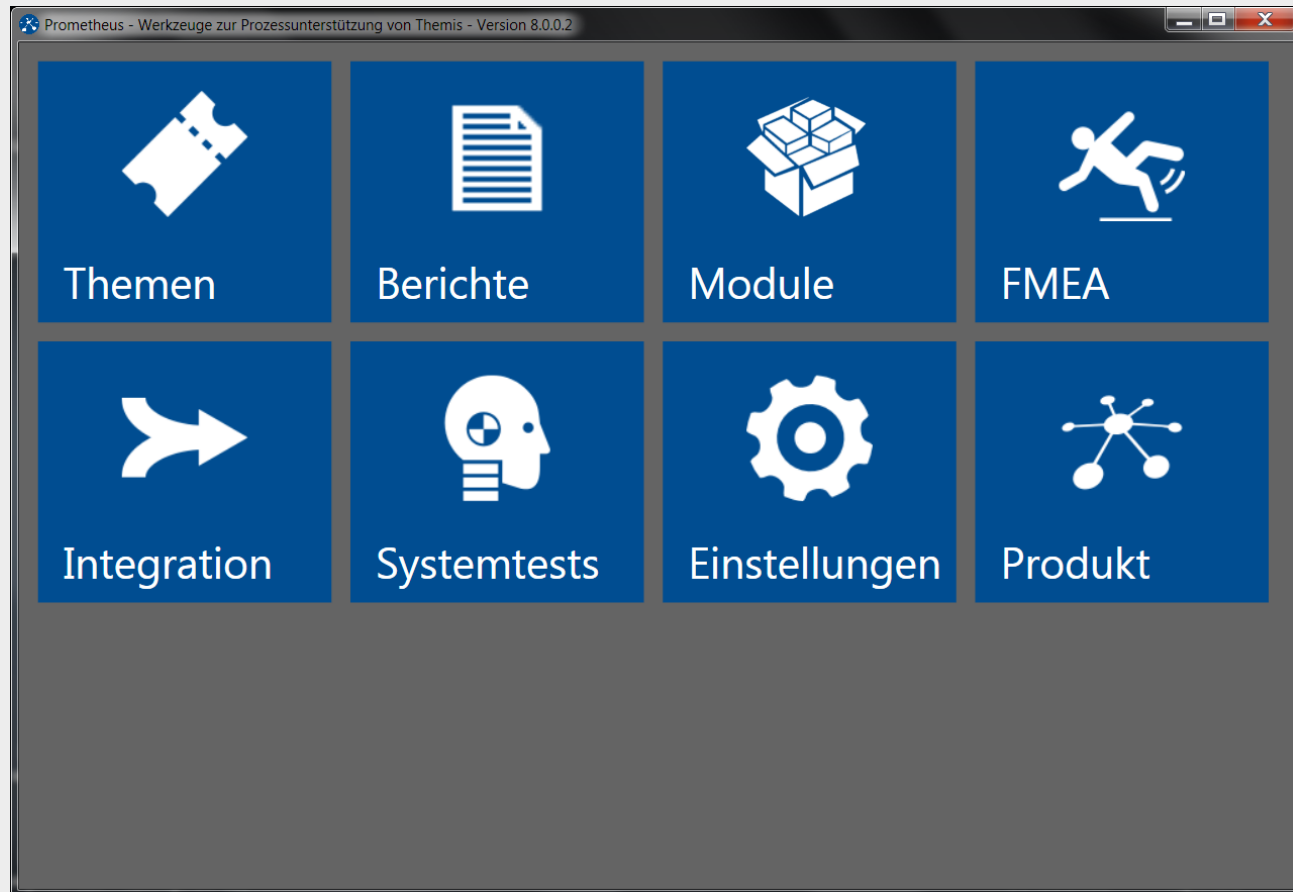
5. Prozesswerkzeug

- ermittelt die erforderlichen Systemtests
- installiert Produkte aus 4.
- rüstet Testkonstruktionen im Testrahmenwerk
- führt Tests aus und prüft Ergebnisse

6. Einbuchen der geänderten Quelltexteinheiten

Prozesswerkzeug Prometheus

- Verfolgen aller **Themen** des Geschäftsprozesses
- Ablösen von **E-Mail** im Unternehmen
- Pflegen der **FMEA-Datenbasis**
- Automatische Auswahl, Ausführung und Auswertung von **Systemtests**
- Unterstützt diskrete und kontinuierliche **Integration**



Prozesswerkzeug Prometheus

- **Abstraktion** von Vorgangsverfolgung, Versionsverwaltung und Dateiablage
- **Schneller Zugriff** auf Status und Elemente eines Themas essentiell für KI.

The screenshot displays the Prometheus software interface. The main window title is 'Prometheus - Werkzeuge zur Prozessunterstützung von Themis - Version 8.0.0.2'. The central area shows details for 'Thema 14993' with the title 'Bordnetz 2020: Flash-Programmierung über D-CAN'. Below this is a table with the following data:

Integrationsprodukt	Typ	Themis-Meilenstein
Argonaut	Externe Wartungsanforderung	I125 - Implementierung abgeschlossen
Status	Priorität	Bearbeiter
ASSIGNED	Urgent	Christian Manhart
Version	Zielversion	Externer Vorgang
-	--	588-10418

Below the details are three columns: 'Weitere Themen', 'Verlauf', 'Aktuell', and 'Favoriten'. The 'Aktuell' column shows a list of tasks with their titles and timestamps. On the right side, there are four summary panels: 'Pinnwand' (a list of messages), 'Aufwände' (a chart showing cumulative hours over time), 'Entwicklungslinien' (a table showing production versions and counts), and 'Risikoanalyse' (a table showing various risk metrics).

Risikoanalyse mit Prometheus

- Ergebnis der **Risikoanalyse** als Kiviat-Diagramm
- Betroffene **Anwendungsfälle**
- Erforderliche **Systemtestfälle**

Prometheus - Werkzeuge zur Prozessunterstützung von Themis - Version 8.0.0.2

Thema 14993
 Risikoanalyse

Thema	14993
(1) Anzahl der Anwendungsfälle	3.3 ln(26)
(2) Ø Wichtigkeit der Anwendungsfälle	7
(3) Anzahl der Fehlerbilder	3.8 ln(45)
(4) Ø Risikoprioritätszahl	5.2
(5) Maximale Risikoprioritätszahl	7.9

Betroffene Anwendungsfälle | Betroffene Testfälle

Name	Wichtigkeit
Aktionsplan ausführen : Funktion freischalten	6
Aktionsplan ausführen : HDD-Aktualisierung durchführen	4
Aktionsplan ausführen : Individualdaten wiederherstellen	7
Aktionsplan ausführen : Maßnahmensteigerung durchführen	8
Aktionsplan ausführen : Programmierabbruch erkennen	9
Aktionsplan ausführen : Steuergerät codieren	8
Aktionsplan ausführen : Steuergerät programmieren	9
Aktionsplan ausführen : Steuergerät tauschen	7
Fahrzeug emulieren : Folgeemulation starten	6
Fahrzeug emulieren : Hybride Emulation beenden	5
Fahrzeug emulieren : Hybride Emulation durchführen	5
Fahrzeug emulieren : Hybride Emulation starten	5
Fahrzeugsitzung durchführen : Aktionsplan berechnen	9
Fahrzeugsitzung durchführen : Fehlerinformationen anzeigen	0

Risiken vergleichen | Systemtests ausführen | Risikobericht erstellen | Aktualisieren

Details

Thema

Bordnetz 2020: Flash-Programmierung über D-

Pinnwand

- Verena Ross 27 August
- Mario Schweigler 23 August
- Bettina von Staden 23 August
- Sarah Erdmann 22 August

Aufwände

229 Stunden

Entwicklungslinien

Produktionsversion	Anzahl
Nächste Produktion	12

FMEA-Werkzeug Prometheus

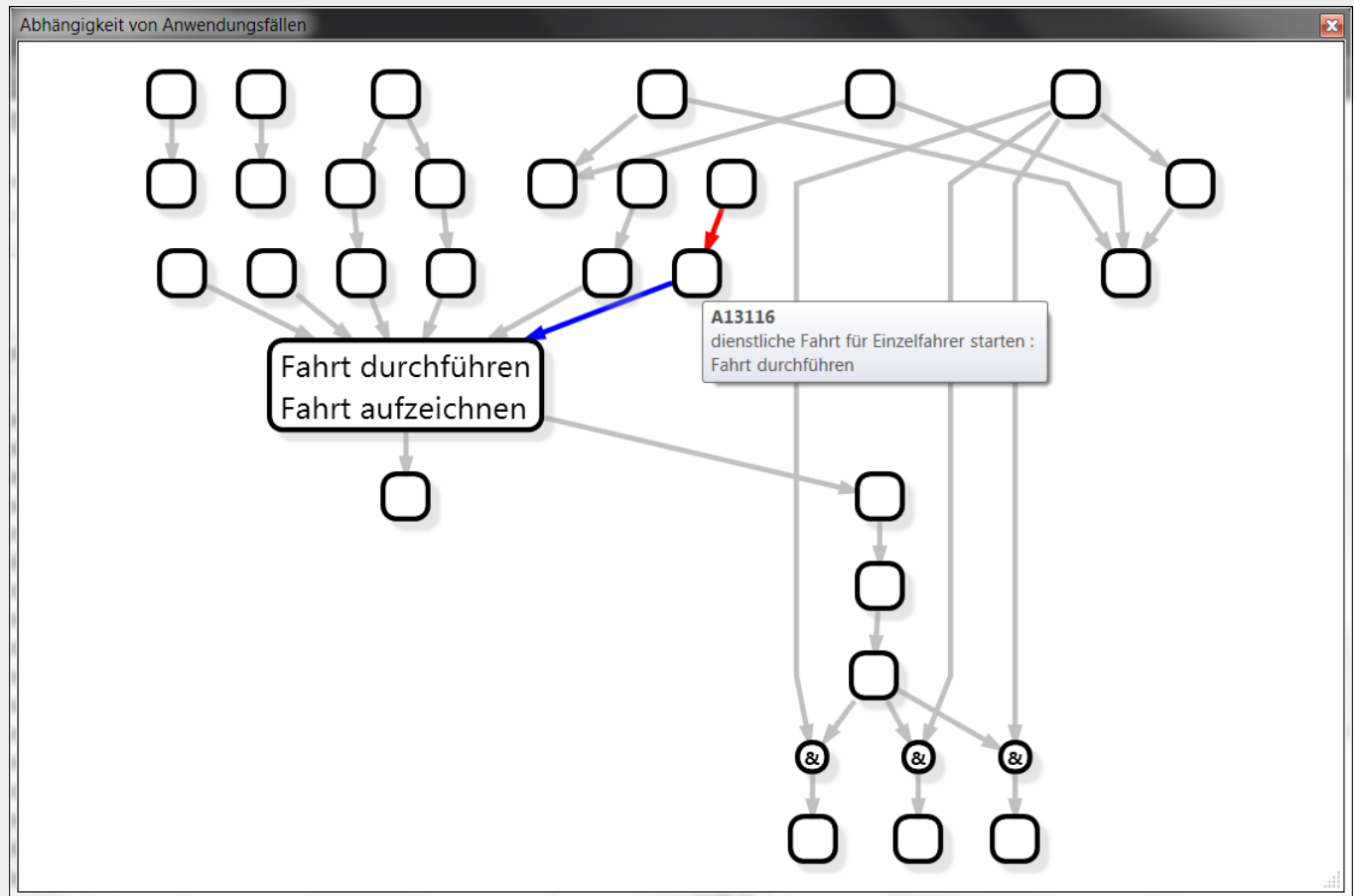
- Strukturierung der **FMEA-Datenbasis** in Domänen
- Schnelle Einsicht in **Abhängigkeiten**

The screenshot shows the Prometheus FMEA tool interface. The main window is titled 'Prometheus - Werkzeuge zur Prozessunterstützung von Themis - Version 8.1.2.2'. The interface is divided into several sections:

- FMEA Header:** Includes a home icon, the title 'FMEA', and a search/filter section.
- Filtern/Suchen:** Shows 'Anwendungsdomäne' set to 'City Lynx' and 'Anwendungsfall' as an empty field. The filter is set to 'Zeige alle Anwendungsfälle'.
- Anwendungsfälle (278) / Testfälle (0):** A table listing use cases with their names and importance scores. The entry 'Fahrt durchführen : Fahrt aufzeichnen' is highlighted with a blue background and has an importance score of 10.
- Details:** Shows the name 'Fahrt durchführen : Fahrt aufzeichnen' and description 'Eine mit City Lynx gestartete Fahrt wird aufgezeichnet.' The use case ID is 'A10207'. It includes a 'Vorschau der Abhängigkeiten des Anwendungsfalls' section with several buttons for actions like 'Fahrt durchführen', 'Fahrt beenden', 'Fahrt unterbrechen', and 'geplante Fahrt für...'. There are also buttons for 'dienstliche Fahrt für...' and 'Geplante Fahrt für...'.
- erforderliche Anwendungsfälle (6):** A section showing dependencies, with sub-sections for 'Fehlerbilder (0)', 'Produkte/Domänen (4)', 'Module (0)', and 'Testfälle (0)'. A tree view shows a hierarchy of domains: Argonaut (BMW, BMWREF, Bosch, ePlanet, ISOM Didact), Bulldog (Broadbill, City Lynx, City Lynx, Einzelfahrer, City Lynx, Gemeinschaftsfahrzeug, City Lynx, Fahrgemeinschaft), and Prometheus.

Abhängigkeitsgraph Prometheus

- Reproduzierbarer und bedienbarer **Graph**
- Darstellung von **Konjunktionen**
- Direktes Bearbeiten der **Abhängigkeiten** von Anwendungsfällen

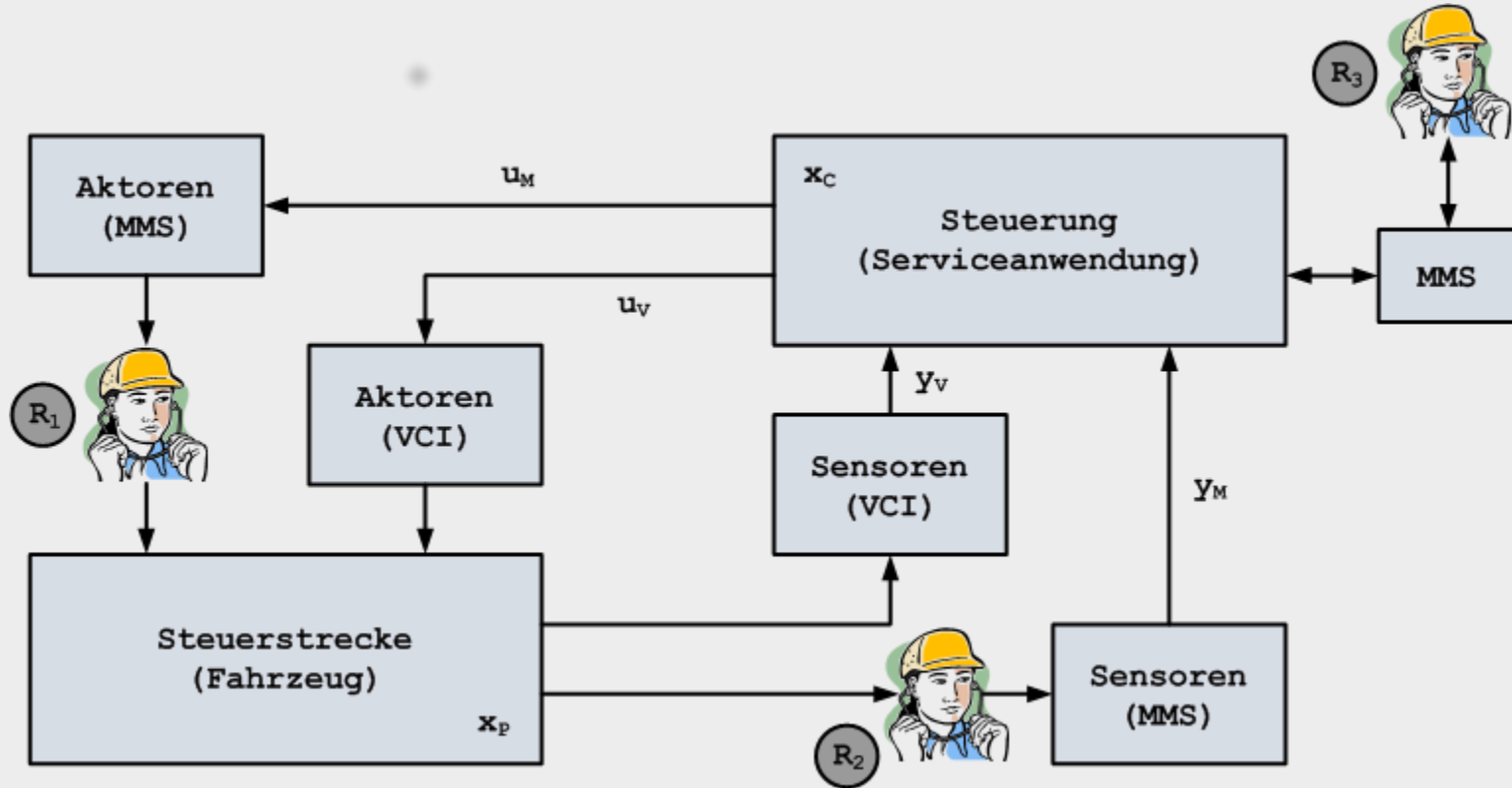


- Basis der BMW-Serviceanwendung ISTA/P, Baureihen **E31** (1989) bis **F34** (Q3/2013)
- Plattform ist mehrmandantenfähig, Servicesprache **ISOM/L**
- Taurus® Cloud /
Taurus® Smart Client
ifs-it.de/produkte/taurus/
- ca. 2 Mio. Zeilen Quelltext

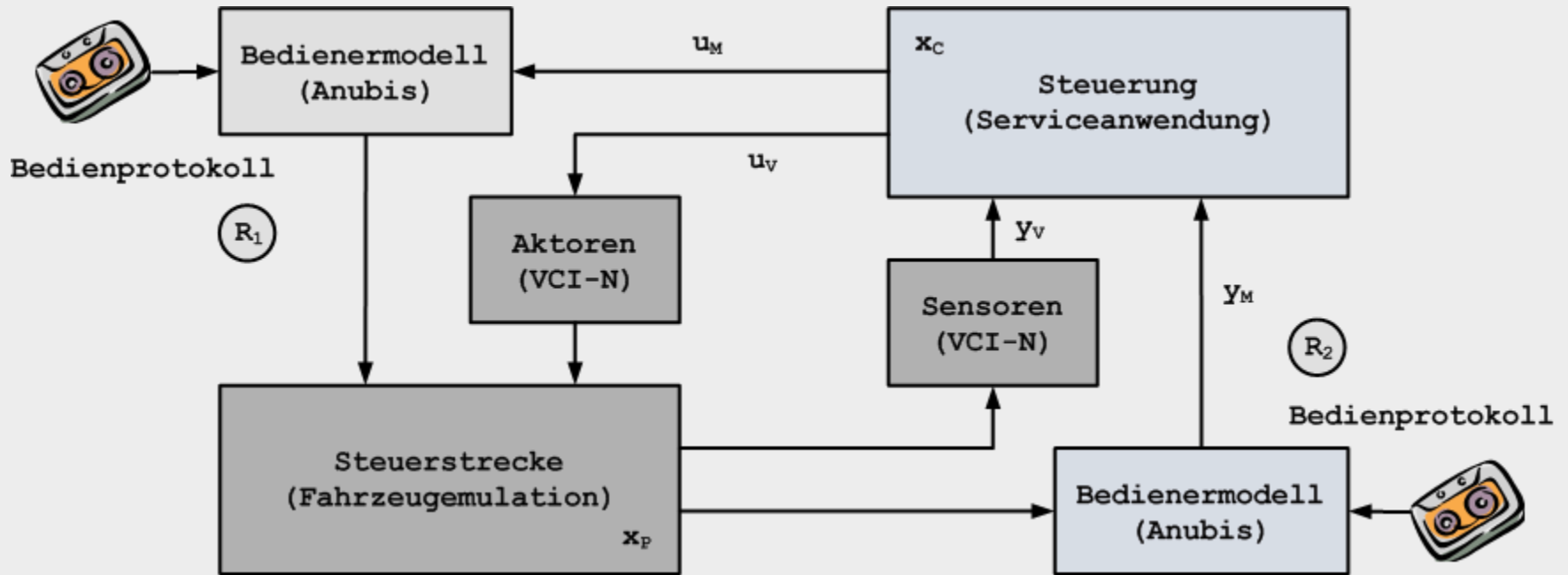


- Rolle **R1** - Arbeiten als **Aktor**
 - Stellgrößen der Anwendung werden von der MMS übermittelt, z.B. als Textanzeige „Lenkrad in Geradeausstellung drehen!“.
- Rolle **R2** - Arbeiten als **Sensor**
 - Vom Bediener ermittelte diskrete Messgrößen des Fahrzeugs werden ebenfalls über die MMS mitgeteilt, z.B. durch Drücken eines Knopfes.
- Rolle **R3** - Arbeiten als **Selektor**
 - Auswählen unter alternativ ausführbaren Teilfunktionen

Regelkreisbetrachtung



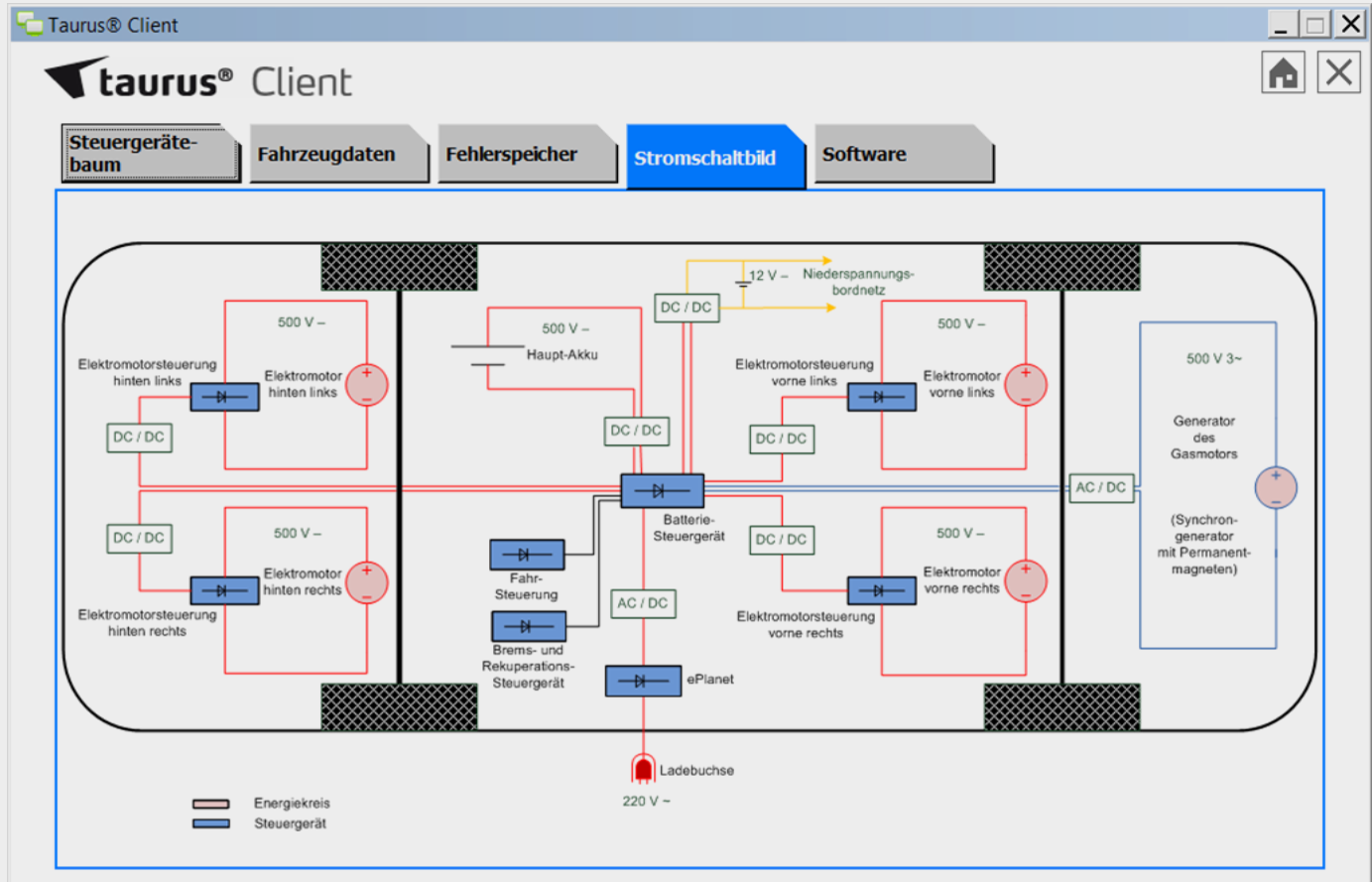
Regelkreis im Testrahmenwerk Anubis



- **DEDS-Modelle** für Steuergeräte aus „Reinraumzugang“
 - Modellierungsform Petri-Netz (Nebenläufigkeit)
 - Emulation von Fahrzeugbordnetz, Speicher und Sensoren
- **Hybride Emulation**
 - Integration von CAN-Hardware
ermöglicht schrittweisen Übergang zum realen Fahrzeug
- Emulationsbeschreibungen aus **Taurus®**-Sitzungen
 - Aufbau von Fahrzeugbibliotheken für Systemtests

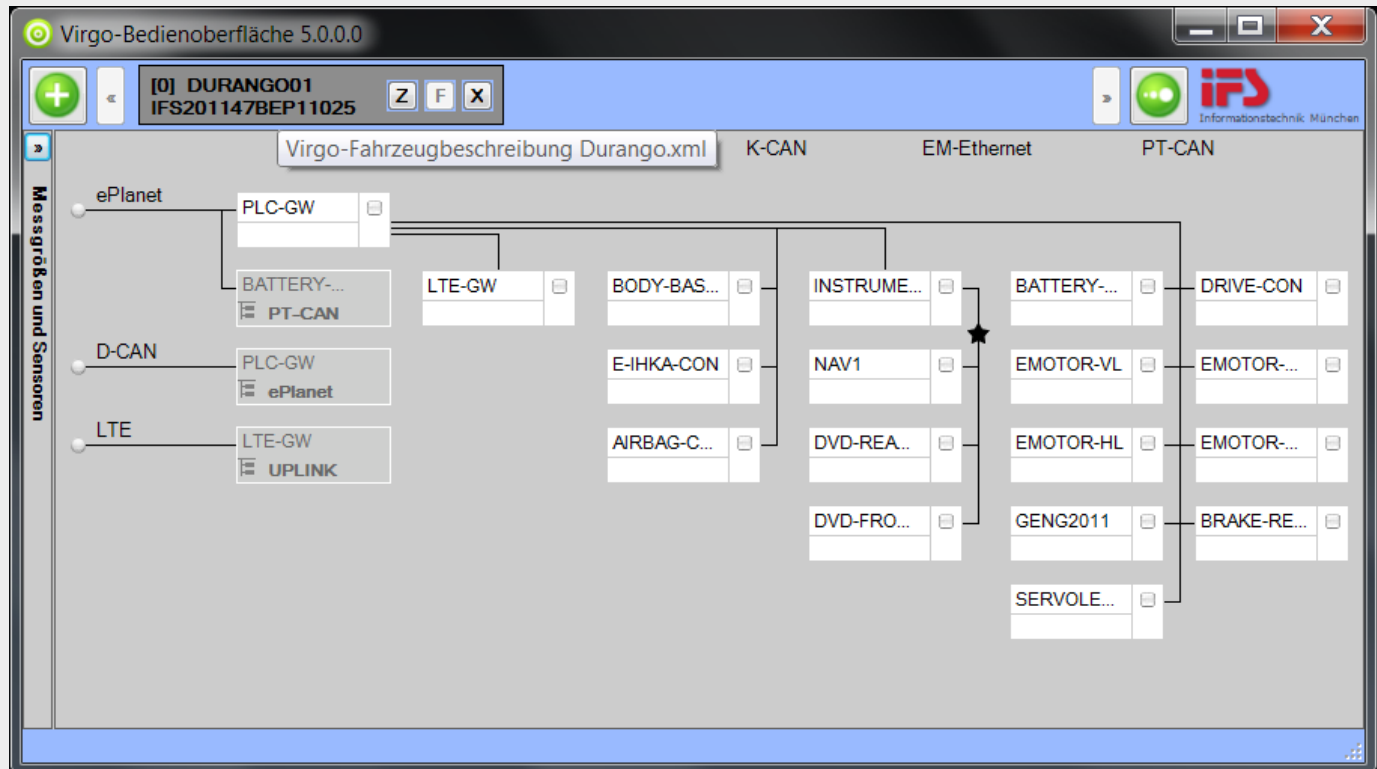
Plug-In Hybrid Durango

- **18 Steuergeräte**, Diagnoseprotokoll UDS
- **Fahrzeugzugang** über ePlanet®, LTE und K-Leitung (ISO 9141)
- **Serviceanwendung** auf Basis ASAM/MCD
- Herausforderung: Softwareaktualisierung „anywhere“!



Emulation Durango

- **Virgo-BO** verwendet Virgo wie Anubis über Web-Service-Schnittstelle
- **Interaktives Rüsten** von Testkonstruktionen, z.B. Einbringen von SG-Fehlverhalten



Emulation Durango, DEDS-Modell

- **Nebenläufigkeit:** Schalten von T_6 markiert E_1 und F_3
- **Verhalten** von Transitionen über Web-Service zur Laufzeit steuerbar
- Testrahmenwerk Anubis kann auch komplexe **Servicesituationen** konstruieren

The screenshot shows the INSTRUMENTS (96) application window. The main area displays a Petri net diagram with places (circles) and transitions (rectangles). Below the diagram, there are two lists: 'Stellen' (Places) and 'Transitionen' (Transitions).

Stellen

F1	Fahrmodus
F2	Fahrmodus mit Programmierfehleranzeige
F3	eingeschränkter Fahrmodus
D	Diagnosemodus

Transitionen

T6	Datenübertragung abgeschlossen
T7	Zeitüberschreitung bei Datenübertragung
T8	für Diagnosemodus autorisieren
T9	Flash-Daten konsistent

A dialog box titled 'Transition für Diagnosemodus autorisieren (T8)' is open in the foreground. It contains the following information:

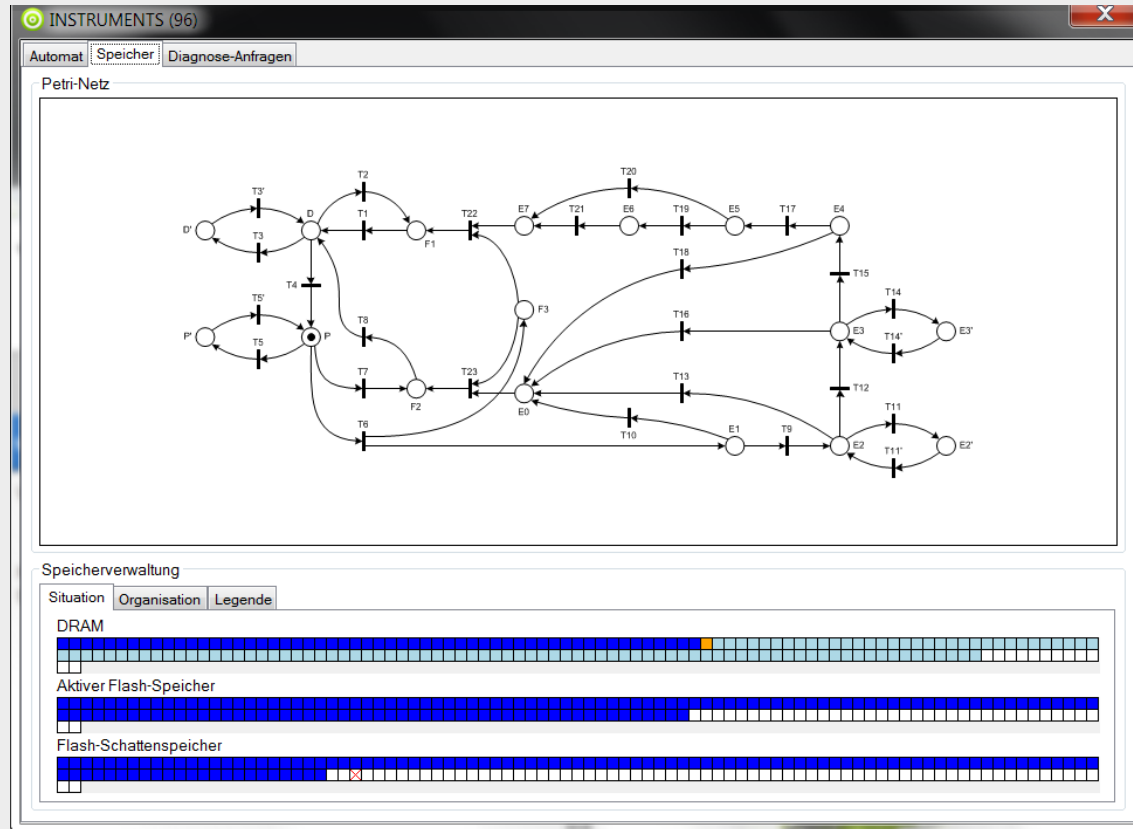
Antwortverhalten:
 wie in der Emulationsbeschreibung konfiguriert
 abweichend

Diagnose-Anfrage	negativ	keine Antwort
RETURN_TO_DIAGNOSTIC_MODE	<input checked="" type="checkbox"/>	<input type="checkbox"/>

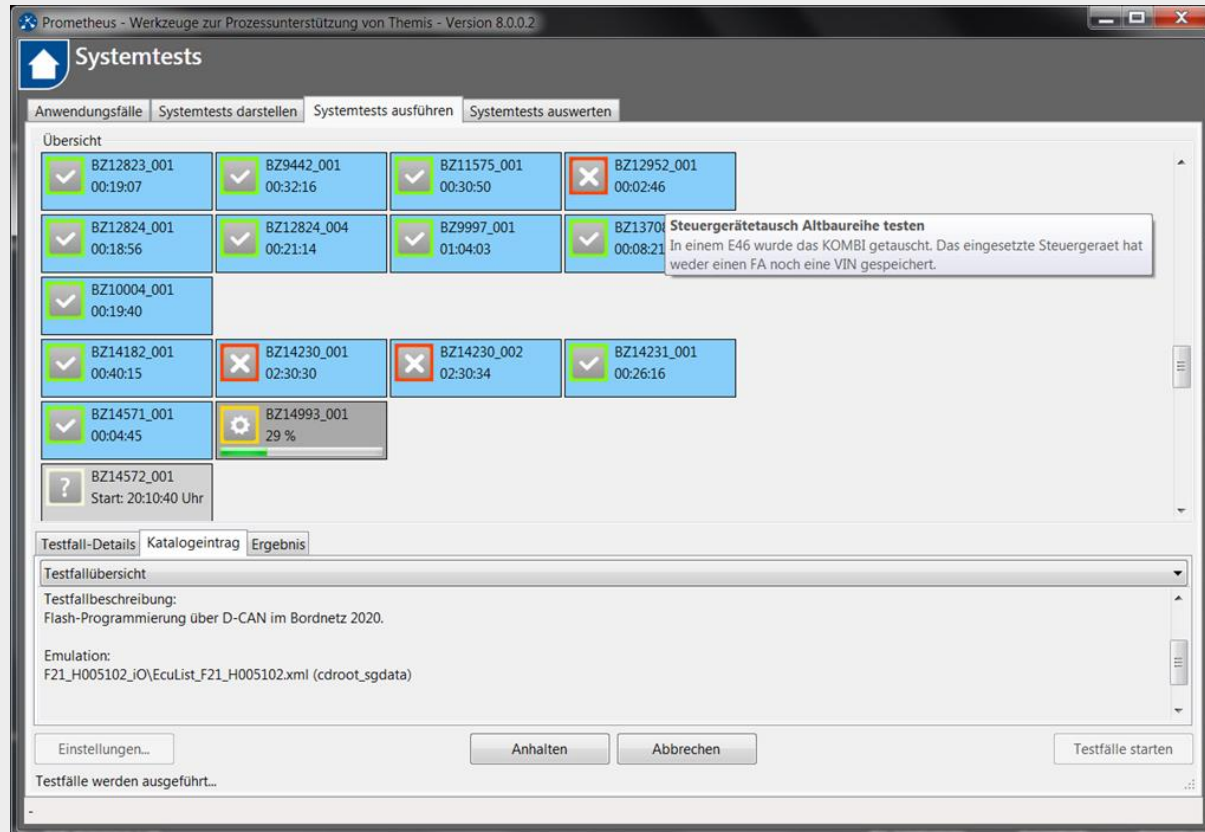
Buttons: Abbrechen, OK

Emulation Durango, SG-Speicher

- Servicesituationen:
SG-Speichermodell
 ermöglicht Nachbildung
 defekter Speicherzellen
- Systemdynamik:
Zeitverhalten des
 Speichers konfigurierbar

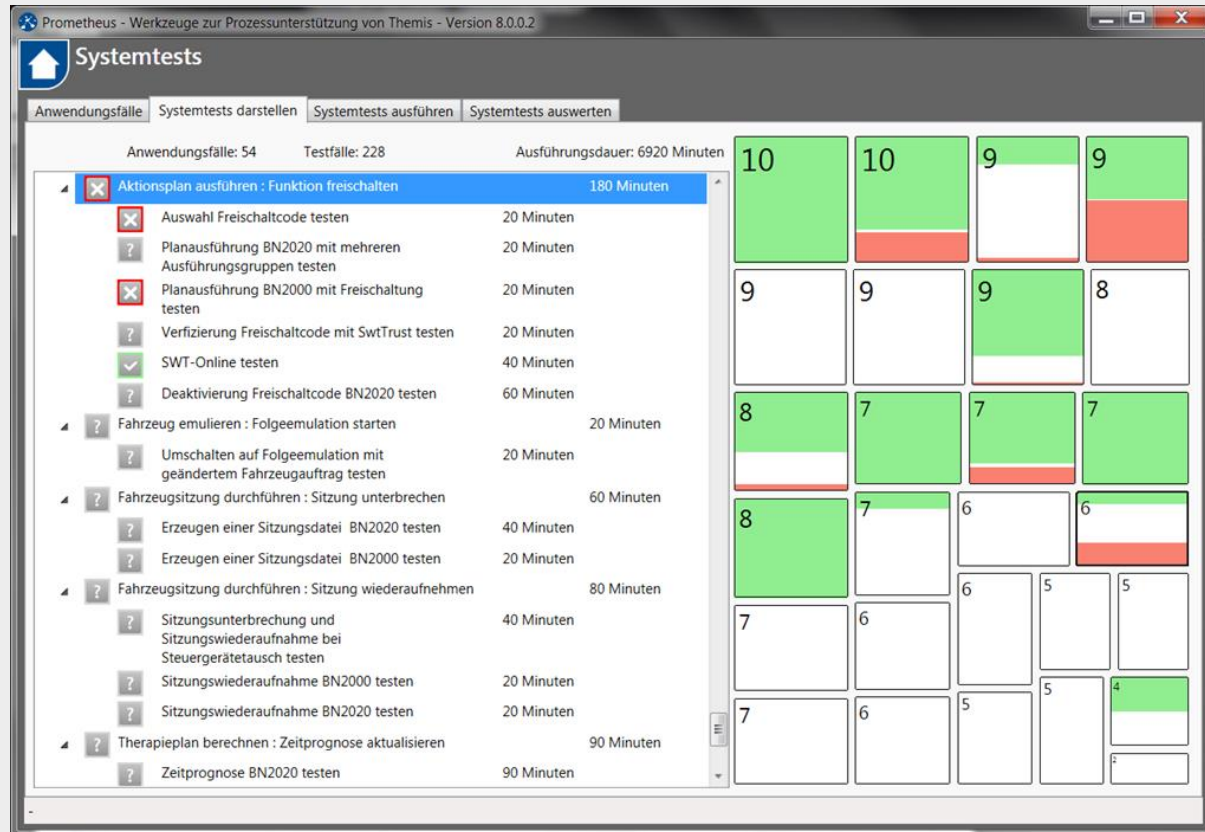


- **Visualisierung** der automatisch ausgeführten Systemtests
- **Systemtestfall** wird von Kachel repräsentiert
- Live-Darstellung des **Fortschritts** der Testausführung



Prometheus und Anubis

- Live-Darstellung der Systemtestergebnisse als **Baumkarte**

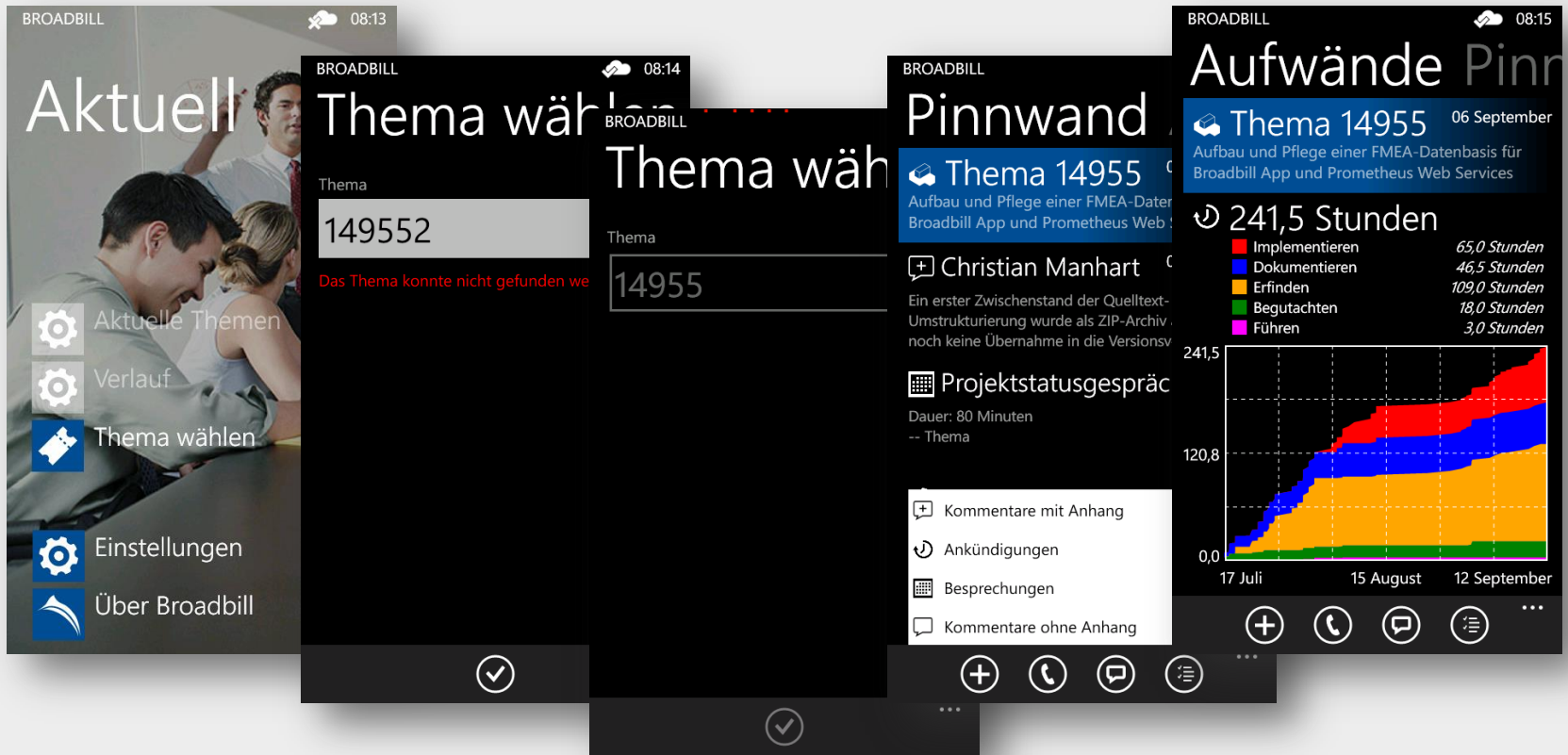


- **FMEA** von Anfang an verfolgen
 - Ergebnisse der Anforderungsanalyse in Datenbasis erfassen
 - Architektur des Produkts an Systemtestfähigkeit ausrichten
 - Quelltexteinheiten nicht technisch organisieren, sondern in Fachprojekten strukturieren.
- Vor **Änderungen** am Produkt
 - FMEA-Datenbasis aktualisieren, Risiken analysieren
 - risikoärmstes Thema realisieren
 - Kontinuierliche Integration von Ergebnissen.

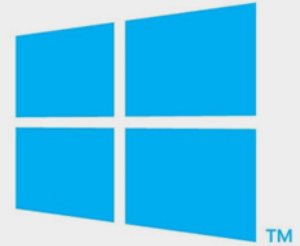


- **MVVM**-Architektur
 - Präsentation (V), Präsentationsmodell (VM), Datenmodell (M) und Datenspeicher (DS), Geschäftslogik (BL)
- Enge **Kopplung** V/VM
 - durch Daten- und Befehlsanbindung, kein „code behind“
 - ermöglicht Fernsteuern VM oder Seitennavigation über öffentliche Schnittstellen (Testrahmenwerk **Echnaton**)
- **Rolle** des Bedieners einfach nachbildbar
 - aber nicht dessen Nutzungserlebnis!

Beispiel Broadbill App



- Plattform Microsoft Windows Phone, verwendet Prometheus Web Services (PWS), ca. 23.000 Zeilen Quelltext.
- Quelltexteinheiten
 - ursprünglich: **1 Projekt**, 3 Quelltextmodule
 - nach Neustrukturierung für FMEA:
40 Fachprojekte, 12 Schnittstellen-Projekte, 52 Quelltextmodule
- **Aufwand** für Neustrukturierung: 14 Personentage.



Vielen Dank für Ihre Aufmerksamkeit

- ▶ info@ifs-it.de, Tel.: 089 – 45 09 83 - 0
- ▶ kontinuierliche-integration.de, continuous-integration.org



© Michael Nagy, Presseamt München

Software-QS-Tag
November 2013

Excellence in Automotive Computing.

Seite 37 7. November 2013

Thema: **14791**

ifs

Informationstechnik München