Workshopprogramm
Entwicklung zuverlässiger Software-Systeme

18. Juni 2009
93053 Regensburg, Galgenbergstr. 30,
Stanglmeierhörsaal im Maschinenbau-Gebäude

Begrüßung und Einführung
Hubert B. Keller

VDI/VDE-Gesellschaft
Mess- und Automatisierungstechnik

Laboratory for
Safe and Secure Systems
a discipline of software engineering

Veranstalter

GI-Fachgruppe Ada

VDI/VDE-GMA-FA Embedded Software

Hochschule Regensburg, Laboratory for Safe and Secure Systems

Ada Deutschland

Mit der freundlichen Unterstützung von AdaCore The GNAT Pro Company

Zuverlässigkeit und Sicherheit softwarebasierter Funktionen sind zentrale Themen der Zukunft

- µC-basierte Systeme
- Software dominiert
- Echtzeit-orientiert
- Innovationstreiber
- Merklicher Mehrwert-/ Kostenfaktor

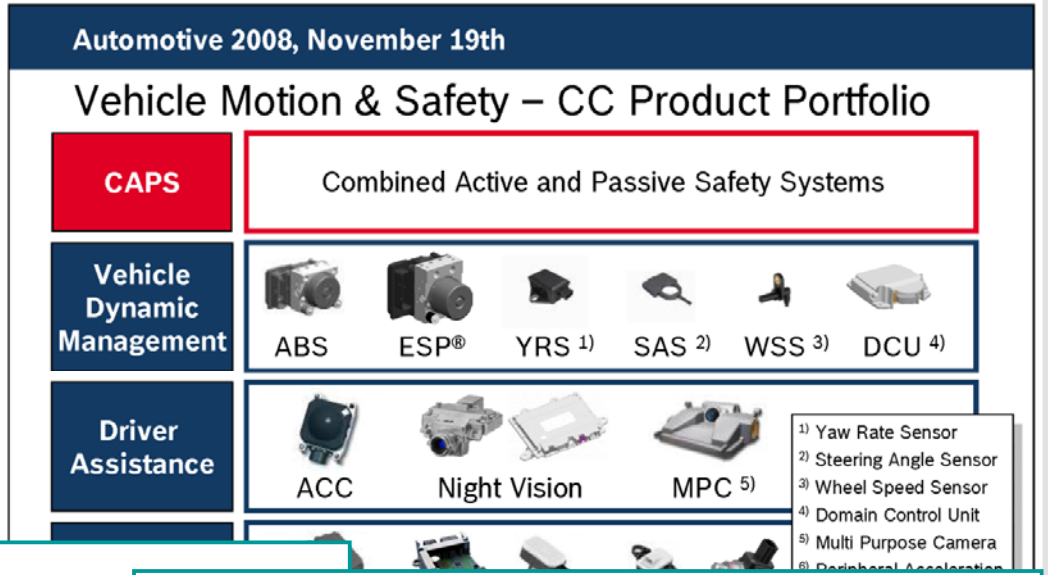
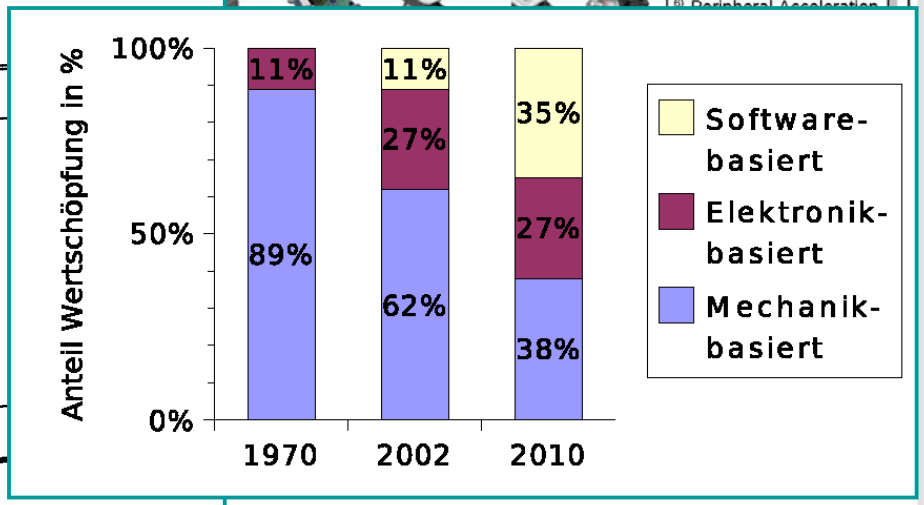


Tabelle 8: Anteil der Kosten für eingebettete Systeme an den Kosten des Endprodukts

Industriezweig	2003	2009*
Automobilindustrie	52%	56%
Luft- und Raumfahrt	52%	54%
Automatisierungstechnik	43%	48%
Telekommunikation	56%	58%
Unterhaltungselektronik und intelligente Häuser	60%	62%
Medizinische Geräte	50%	52%
Gewichteter Durchschnitt	**51%**	**53%**

Anmerkung: * Schätzung
Quelle: F.A.S.T.; TU München (2005)



(1) Die Kooperation von Forschungszentrum Karlsruhe GmbH und Universität Karlsruhe (TH)
HELMHOLTZ GEMEINSCHAFT
Forschungszentrum Karlsruhe in der Helmholtz-Gemeinschaft

IEC 61508 als Framework – Teil 7 Recommended Techniques

C.4 Entwicklungswerkzeuge und Programmiersprachen

C.4.1 Typstrenge Programmiersprachen

- Reduzieren der Fehlerwahrscheinlichkeit durch die Nutzung von Sprachen, die einen hohen Grad von Überprüfungen durch den Compiler erlauben
- Strenge Prüfungen, um sicher zu stellen, dass der korrekte Typ verwendet wird (auch für separat übersetzte Einheiten)
- Unterstützung weiterer Aspekte guten Software-Engineerings: z.B. einfach analysierbare Kontrollstrukturen (if.. then.. else, do.. while, etc.), die zu wohlstrukturierten Programmen führen
- Typische Beispiele typstrenger Sprachen sind Pascal, Ada und Modula 2.

…

- **C.4.3 Zertifizierte Werkzeuge und Übersetzer**
- Wenn möglich, sollten alle Werkzeuge zertifiziert werden, so dass ein gewisser Vertrauensgrad bezüglich der Outputs angenommen werden kann
- dies sollte durch eine unabhängige Institution gegenüber unabhängigen Kriterien/Standards erfolgen

Probleme softwarebasierter Funktionen

Software Engineering - Chaos Report der Standish Group
(Basis über 40.000 IT Projekte)

1994

- 16 % erfolgreiche Projekte (im Zeit- und Budget-Plan, mit geforderten Funktionen)
- 31 % abgebrochene Projekte
- 53 % irgendwie beendete Projekte

1996 → 27 % erfolgreiche Projekte

1998 → 26 % erfolgreiche Projekte

2000 → 26 % erfolgreiche Projekte

2002 → 32 % erfolgreich (Prozess, Methode, Tool)

2004 → 29% erfolgreich

2006 → 35% erfolgreich, 19% abgebrochen, 46% irgendwie beendet (teurer, später)

2008 → 32% erfolgreich, 24% abgebrochen, 44% irgendwie beendet (teurer, später)

Probleme softwarebasierter Funktionen

Fehler und Phasenzuordnung

HSE Out of Control – Analyse (2003)

- **44% had inadequate specification as their primary cause**
- 15% design and implementation
- 6% installation and commissioning
- 15% operation and maintenance
- 20% changes after commissioning

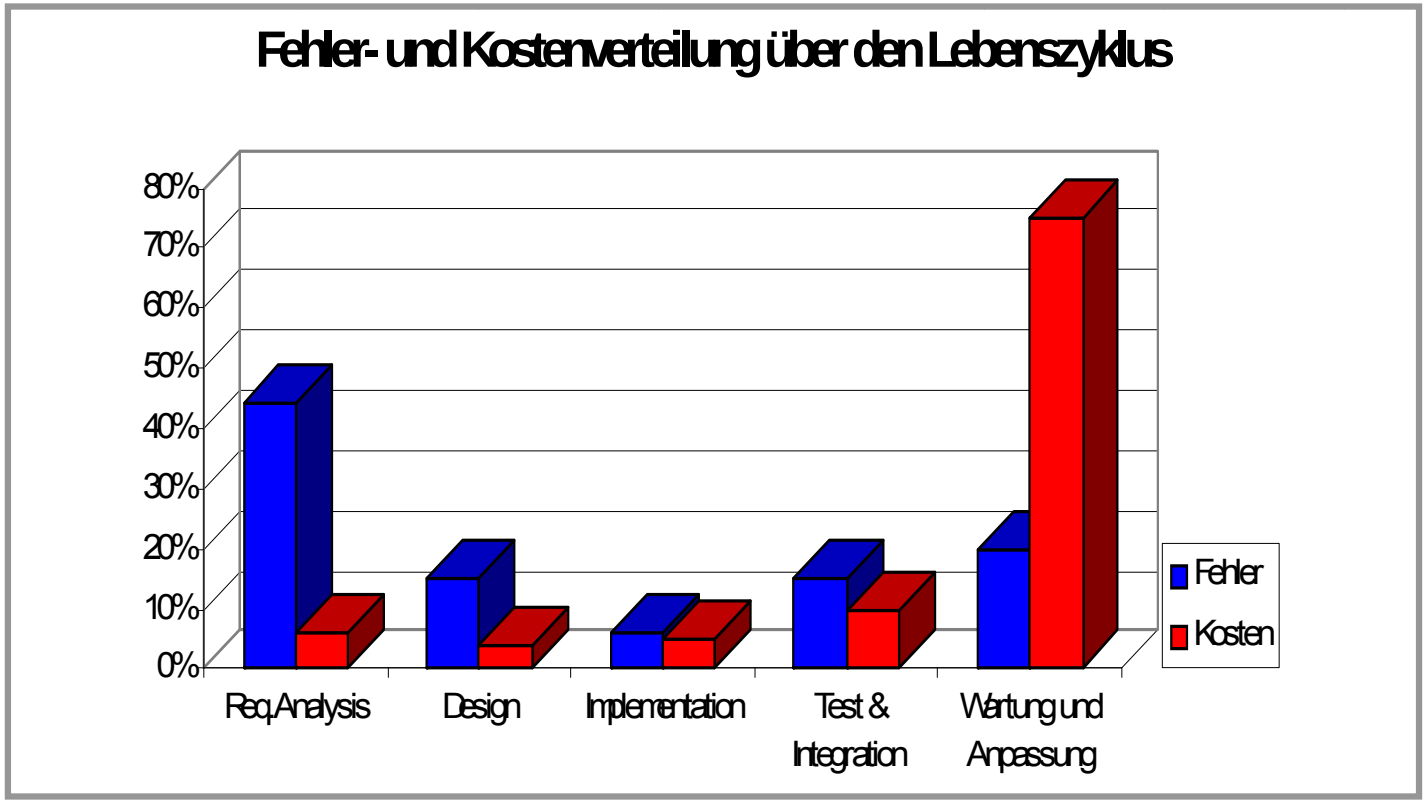
Analysephase wird deutlich unterschätzt
→ Mehr Fokus auf Analysemodell!
Aufwand Codierung weniger wichtig (MDA)!

(Out of control - Why control systems go wrong and how to prevent failure. Health and Safety Executive (2003), No 238)

Probleme softwarebasierter Funktionen

Kosten-/ Fehlerverteilung

Reduzierte Analyse- und Modellierungsphase produzieren massiv Fehler



50%-80% der Kosten entstehen zeitverzögert in der Wartung!

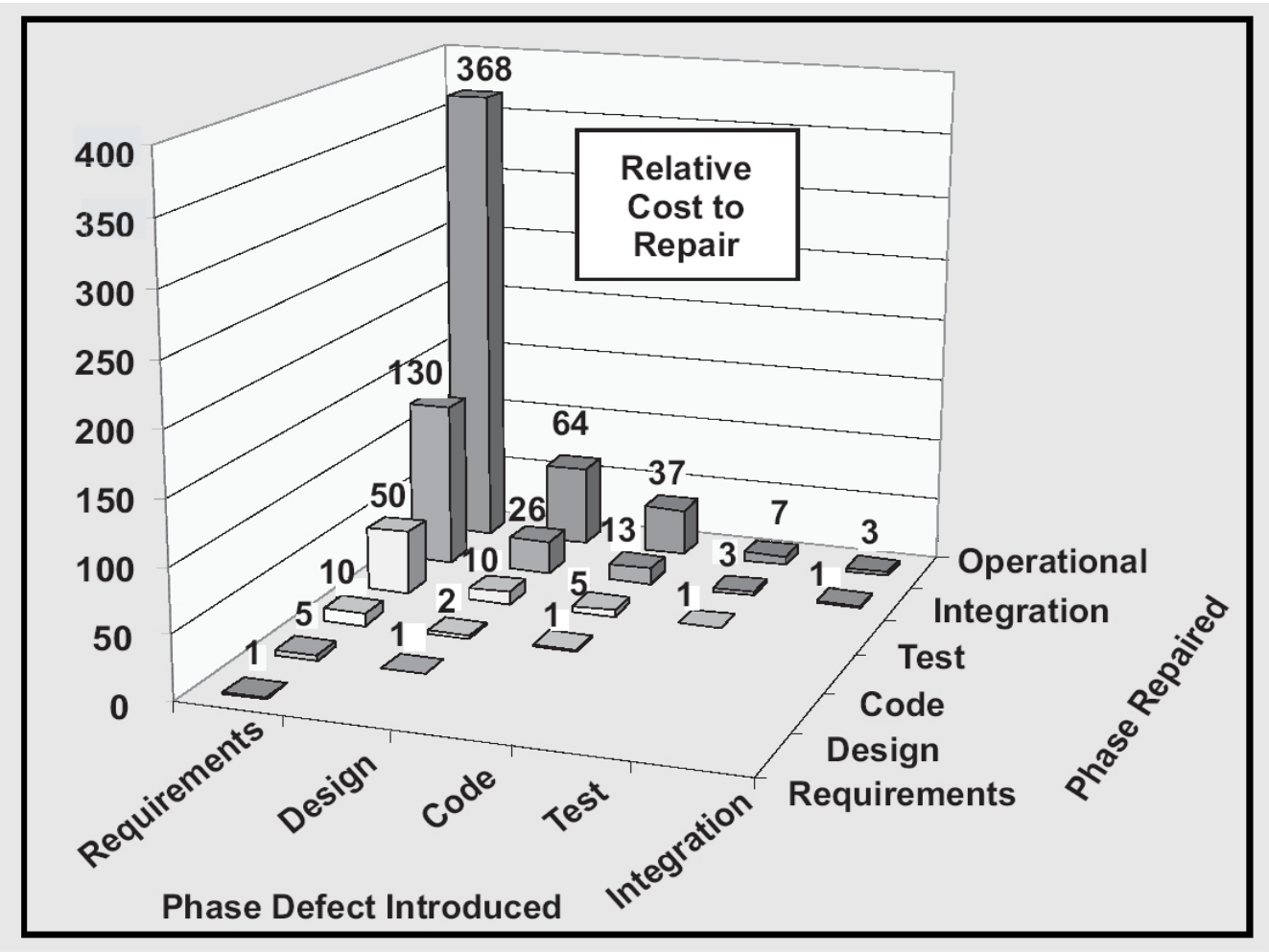
Probleme softwarebasierter Funktionen

Zeitverzögerte Kostenexplosion

(CrossTalk - Journal of Defense Software-Engineering, Basis: NASA-Studie)

(→ vgl. HSE-Analyse 44% der Fehler durch mangelhafte Spezifikation!)

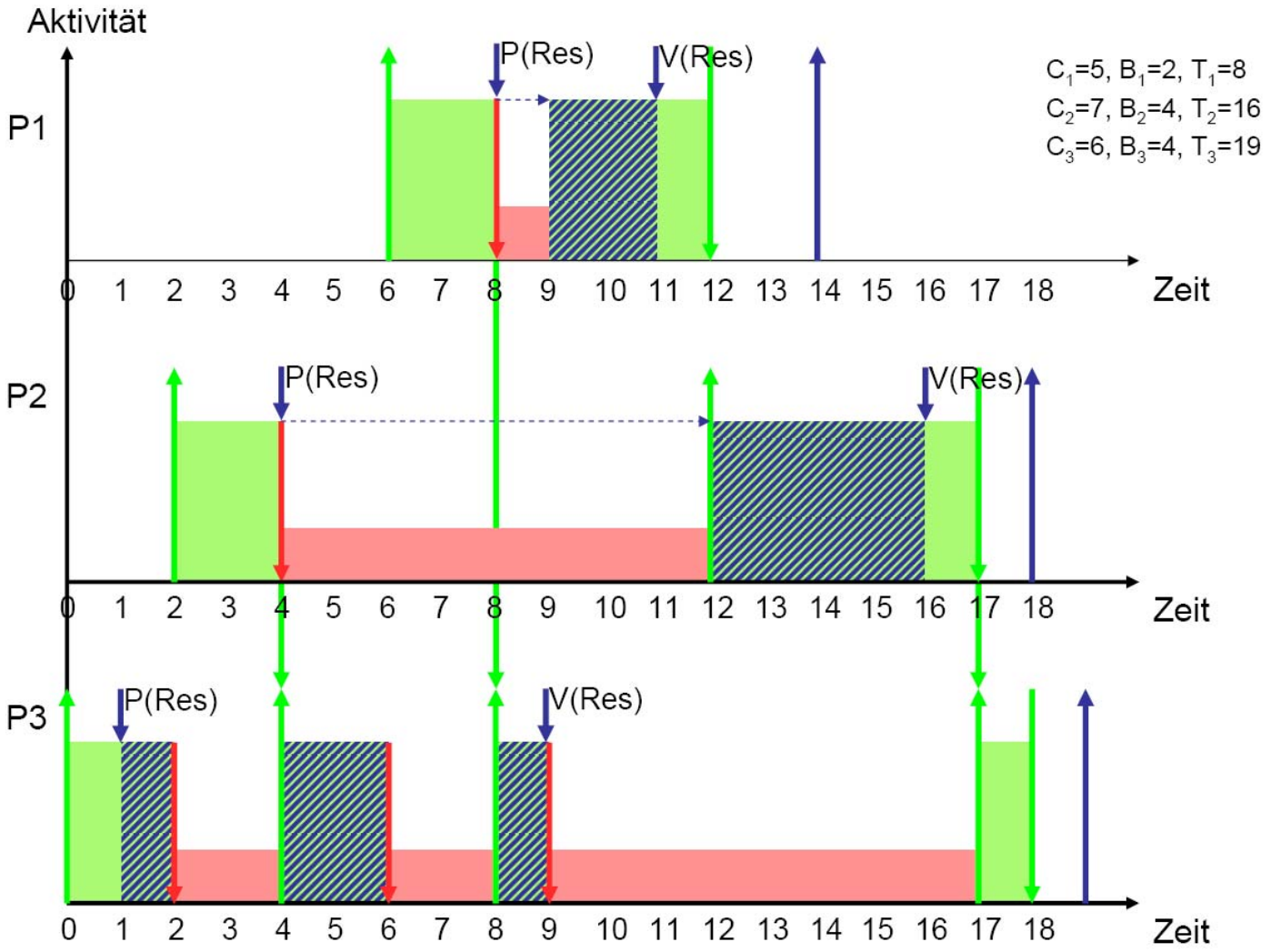
Relative Fehlerbehebungskosten abhängig von der Phase



Realzeitsysteme – Testen reicht nicht!

Zeitanomalien:
Test zeigt Ausführbarkeit

(Beweis aufwendig.
Ist unnötig, klappt schon!
Ist ja getestet!)



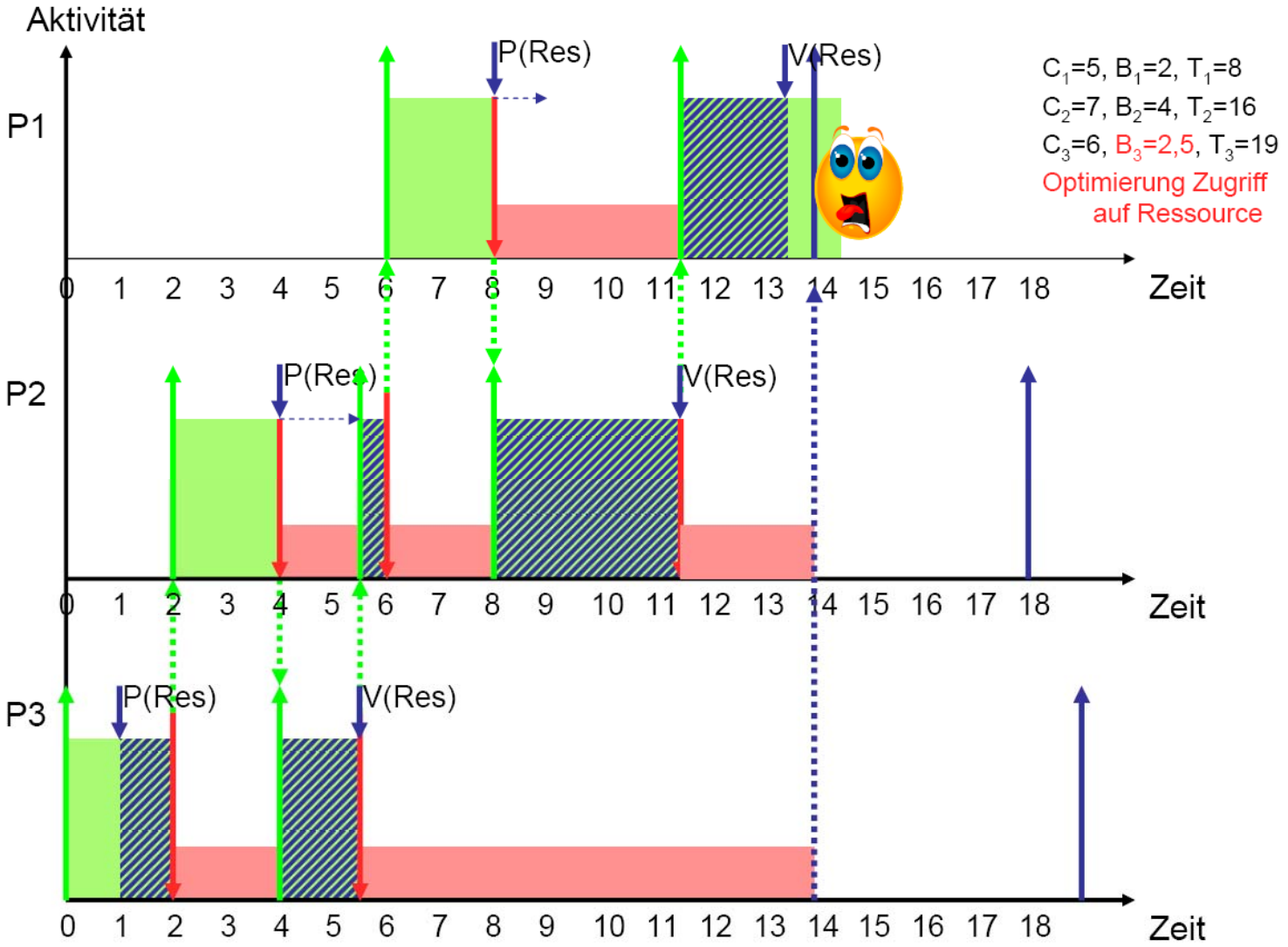
Realzeitsysteme

Zeitanomalien:
Test zeigte vorher Ausführbarkeit!

Zugriff auf Ressource durch Prozess P3 von B=4 auf B=2,5 optimiert

Optimierung von Blockadezeiten führt dann zu Zeitfehler!

Test war kein Beweis!
(Schnellerer Prozessor analog!)



Übersicht

***Key Note:* John Barnes: The Spark approach to high integrity software**

Sitzungen, Mi, 17. Juni

16.30 Uhr GI FG Ada, Mitgliederversammlung
17.30 Uhr FV Ada-Deutschland, Mitgliederversammlung
18.30 Uhr GMA FA Embedded Software, Sitzung

Alle Sitzungen am 17.6.2009 finden statt bei:
Hochschule Regensburg, Fakultät Elektro- und Informationstechnik, Hörsaal S-110 (Raum im Kellergeschoss), Seybothstrasse 2, 93025 Regensburg*

Abendveranstaltung, Mi, 17. Juni

20.00 Uhr Cafe & Restaurant Vitus, Philippe Lebeau,
93047 Regensburg, Hinter der Grieb 8, T+49 941 52646
Dinner Speech: Peter Siwon, MicroConsult, "Intuitionsfallen in Projekten - Erfahrungen aus dem Projektalltag"

Sitzung, Fr, 19. Juni

9.00 Uhr FB Sicherheit, AK Begriffe
Die Sitzung am 19.6.2009 findet statt bei:
IT Inkubator Ostbayern GmbH
Bruderwöhrdstraße 15b, 93055 Regensburg
Tel.: 09 41/60 48 89-0

Programm, Do. 18. Juni

- 8.00 Registrierung (93053 Regensburg, Galgenbergstr. 30, Stanglmeierhörsaal im Maschinenbau-Gebäude)
- 8.30 Begrüßung
  Grußwort Prof. Dr. Josef Eckstein, Präsident der Hochschule Regensburg
- 8.45-9.40 John Barnes: The Spark approach to high integrity software
- 9.40-10.20 Daniel Kästner, AbsInt GmbH: Astrée: Nachweis der Abwesenheit von Laufzeitfehlern
- *10.20-10.50* *Pause (Ausstellung)*
- 10.50-11.10 Harald Hauff, Universität Passau: Erfüllung von funktionalen und nichtfunktionalen Anforderungen in eingebetteten Systemen durch modellbasierte Software-Entwicklung
- 11.10-11.50 Zamira Daw, Hochschule Mannheim: Methode zur Entwicklung sicherheitskritischer Systeme mittels deterministischer UML-Modelle
- 11.50-12.30 Armin Beer, Siemens AG: Model-based testing and verification of dependable systems
- *12.30-13.30* *Mittagspause*
- 13.30-13.50 Gunter Blache, ETAS GmbH: Model based development & automatic code generation for safety critical systems with ASCET
- 13.50-14.10 Michael Erskine, LFK-Lenkflugkörpersysteme GmbH: Sicherheitskritische Software mit dem V-Modell
- 14.10-14.30 Stefan Puchner, TU München: Safety-Cases für zertifizierte Fahrzeugfunktionen
- 14.30-14.50 Sandro Schulze, University of Magdeburg: IT Security in Automotive Software Development
- 14.50-15.10 Michael Niemetz, Continental Automotive GmbH: Quirks and Challenges in the Design and Verification of Efficient, High-Load Real-Time Software Systems
- *15.10-15.45* *Pause (Ausstellung)*
- 15.45-16.25 M. Steindl, Regensburg University of Applied Sciences: Migration of Safely Embedded Software to FPGA Based Architectural Concepts
- 16.25-16.45 Matthias Götz, Forschungszentrum Karlsruhe: Analyse und Entwicklung einer µ-Controller-basierten Echtzeitsteuerung für Sensorsysteme mit Ada-Raven
- 16.45-17.25 Sven Söhnlein, Universität Erlangen-Nürnberg: Zuverlässigkeitsbewertung einer Getriebesteuerungs-Software durch Auswertung der Betriebserfahrung
- 17.30-17.45 Abschluss

Ausstellung

Workshopprogramm
Entwicklung zuverlässiger Software-Systeme

18. Juni 2009
93053 Regensburg, Galgenbergstr. 30,
Stanglmeierhörsaal im Maschinenbau-Gebäude

VDI/VDE-Gesellschaft
Mess- und Automatisierungstechnik

Laboratory for
Safe and Secure Systems
a discipline of software engineering