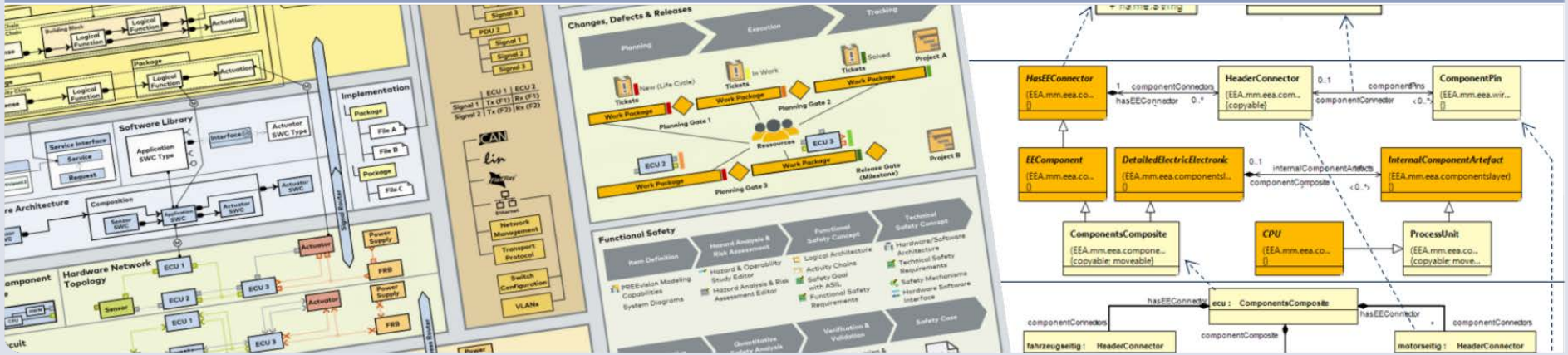
Karlsruher Institut für Technologie

Vorlesung Software Engineering (SE)

Wintersemester 2017/2018

Kapitel 1 – Einführung und Organisatorisches



Karlsruher Institut für Technologie

1. Einführung und Organisatorisches

Dozenten

Dozent

Dr.-Ing. **Clemens Reichmann**

Ort	Philipp-Reis-Str. 1, Karlsruhe
Telefon	0721 / 91430-200
Mail	Clemens.Reichmann@vector.com
Web	http://www.vector.com

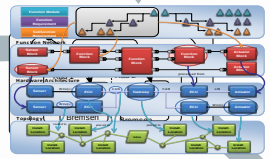
Betreuer

Dipl.-Ing. **Lukas Erlinghagen** (ITIV)

Ort	ITIV Gebäude 30.10 / Engesserstraße 5, Raum 126
Telefon	0721 / 608 - 47659
Mail	lukas.erlinghagen@kit.edu
Web	http://www.itiv.kit.edu

Studienmodell 13 – Systems Engineering

Idee



Systementwurf

Spezifikation - Modellierung - Simulation
Requirements Engineering
Echtzeitanforderungen

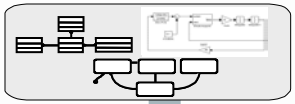
HW-Beschreibungssprachen

Digital (VHDL)
Analog (VHDL-AMS, MAST)

```
PROCESS (schlupf, state)
BEGIN
  CASE state IS
    WHEN freilauf =>
      IF schlupf > 0 THEN
        next_state <=
bremsen;
      ELSE
```

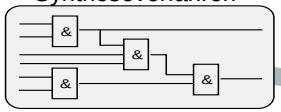
SW-Beschreibungssprachen

UML, Statecharts (Statemate),
Blockdiagramme (Matlab SIMULINK)



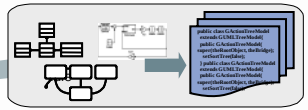
Entwurfs-Automatisierung

Laufzeitmodellierung
Syntheseverfahren



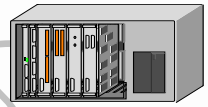
Code Generierung

Zielarchitekturen
Codegeneratoren



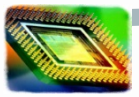
Rapid Prototyping

Hardwareplattformen
Codegenerierung
Echtzeitbetriebssysteme
Konfigurierbare Schnittstellen



Realisierung

Elektronische Systeme
System on Chip (SoC)
Mikrooptische Systeme
Intelligente Sensoren
Prototypenentwicklung

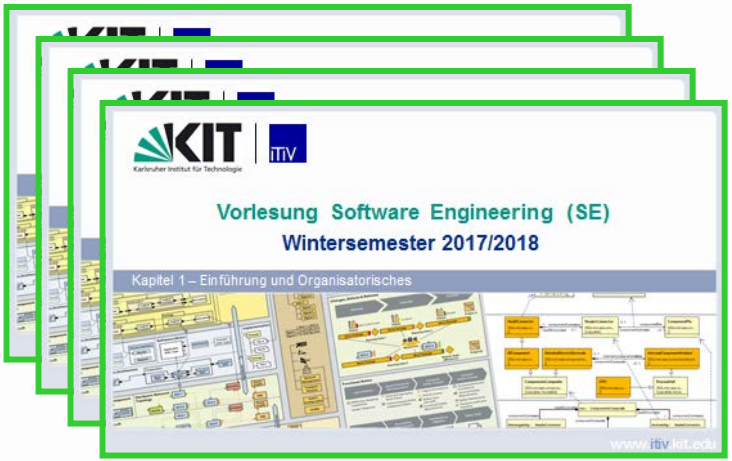


Technologietransfer

Telekommunikation – Medizintechnik
Automatisierung – Automobilelektronik
Kooperationsprojekte und Auftragsforschung
Schulungen für Industriekunden

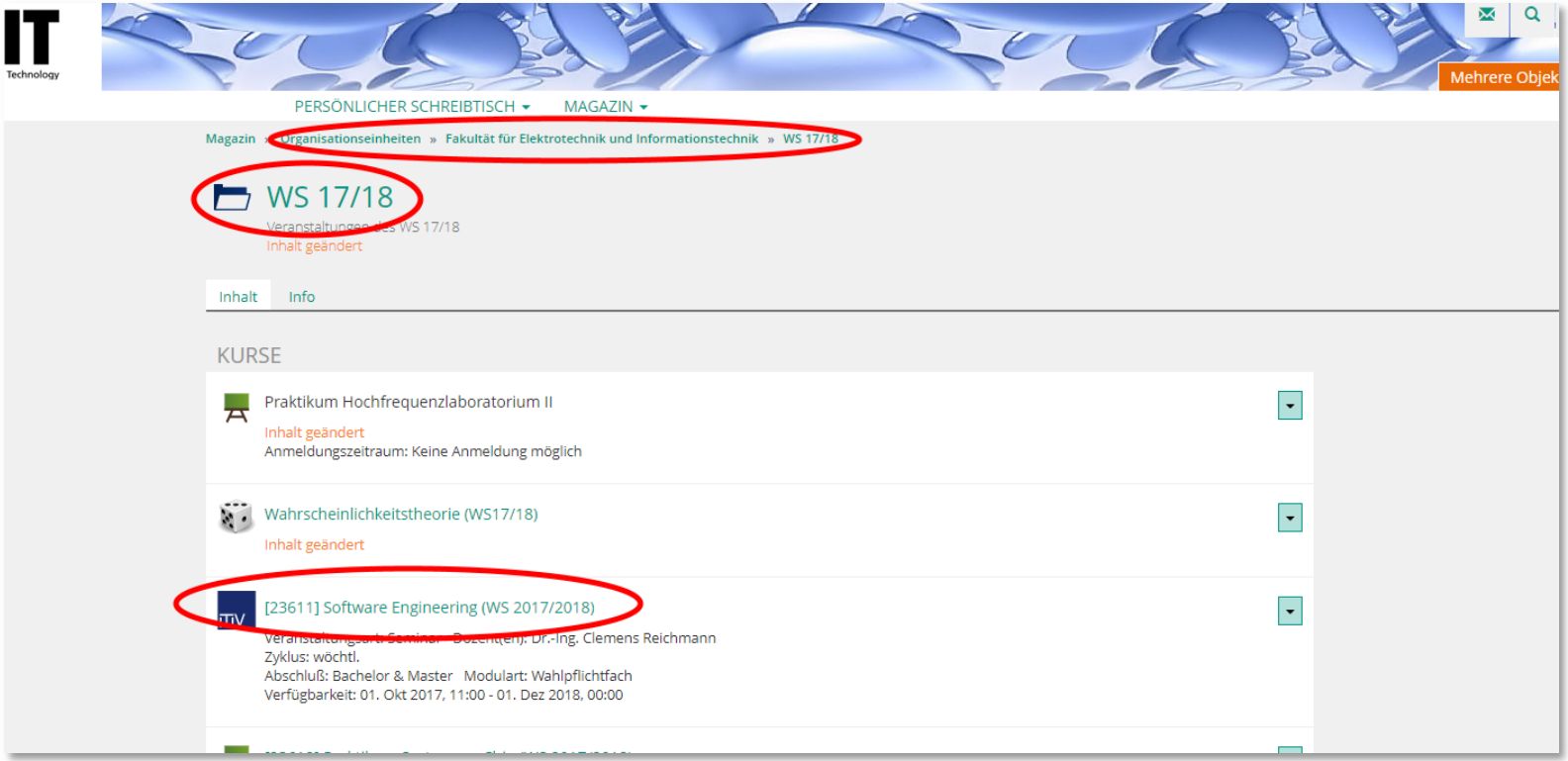
Link zum Kurs

- **Aktualisierung** des Kurses jeweils nach der Vorlesung
- Diese **Folien** stehen im **ILIAS** als PDF **zum Download** zur Verfügung:
  https://ilias.studium.kit.edu/goto.php?target=crs_722787&client_id=produktiv
  https://ilias.studium.kit.edu/goto.php?target=fold_729212&client_id=produktiv

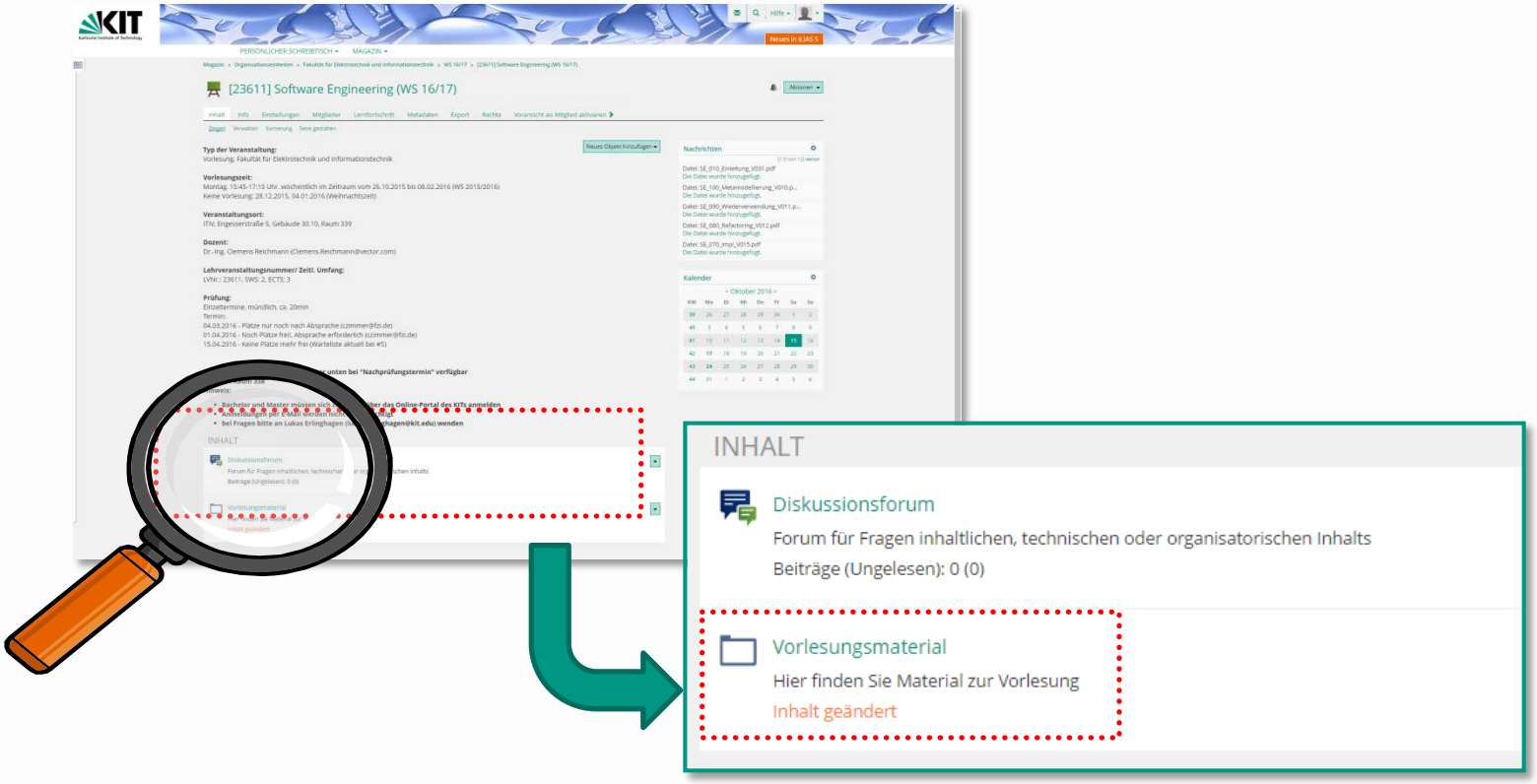


Passwort: softw@re

Kurszugang


Technology
Mehrere Objek
PERSÖNLICHER SCHREIBTISCH
MAGAZIN
Magazin » Organisationseinheiten » Fakultät für Elektrotechnik und Informationstechnik » WS 17/18
WS 17/18
Veranstaltungen des WS 17/18
Inhalt geändert
Inhalt
Info
KURSE
Praktikum Hochfrequenzlaboratorium II
Inhalt geändert
Anmeldungszeitraum: Keine Anmeldung möglich
Wahrscheinlichkeitstheorie (WS17/18)
Inhalt geändert
[23611] Software Engineering (WS 2017/2018)
Zyklus: wöchtl.
Abschluß: Bachelor & Master Modulart: Wahlpflichtfach
Verfügbarkeit: 01. Okt 2017, 11:00 - 01. Dez 2018, 00:00

Navigation


[23611] Software Engineering (WS 16/17)
INHALT
Diskussionsforum
Forum für Fragen inhaltlichen, technischen oder organisatorischen Inhalts
Beiträge (Ungelesen): 0 (0)
Vorlesungsmaterial
Hier finden Sie Material zur Vorlesung
Inhalt geändert

Montag, 15:45 – 17:15 Uhr

SR 1.067
Geb. 20.30

entfällt

Vorlesung	Termin
1	**16.10. 2017**
2	23.10.2017
3	30.10.2017
4	06.11.2017
5	13.11.2017
6	20.11.2017
7	27.11.2017
8	04.12.2017
9	11.12.2017
10	18.12.2017

Vorlesung	Termin
11	08.01.2018
12	15.01.2018
13	22.01.2018
14	29.01.2018
15	05.02.2018

Oktober 2017

KW	MO	DI	MI	DO	FR	SA	SO
39							1
40	2	3	4	5	6	7	8
41	9	10	11	12	13	14	15
42	16	17	18	19	20	21	22
43	23	24	25	26	27	28	29
44	30	31					

November 2017

KW	MO	DI	MI	DO	FR	SA	SO
44			1	2	3	4	5
45	6	7	8	9	10	11	12
46	13	14	15	16	17	18	19
47	20	21	22	23	24	25	26
48	27	28	29	30			

Dezember 2017

KW	MO	DI	MI	DO	FR	SA	SO
48					1	2	3
49	4	5	6	7	8	9	10
50	11	12	13	14	15	16	17
51	18	19	20	21	22	23	24
52	25	26	27	28	29	30	31

Januar 2018

KW	MO	DI	MI	DO	FR	SA	SO
1	1	2	3	4	5	6	7
2	8	9	10	11	12	13	14
3	15	16	17	18	19	20	21
4	22	23	24	25	26	27	28
5	29	30	31				

Februar 2018

KW	MO	DI	MI	DO	FR	SA	SO
5				1	2	3	4
6	5	6	7	8	9	10	11
7	12	13	14	15	16	17	18
8	19	20	21	22	23	24	25
9	26	27	28				

Ablauf

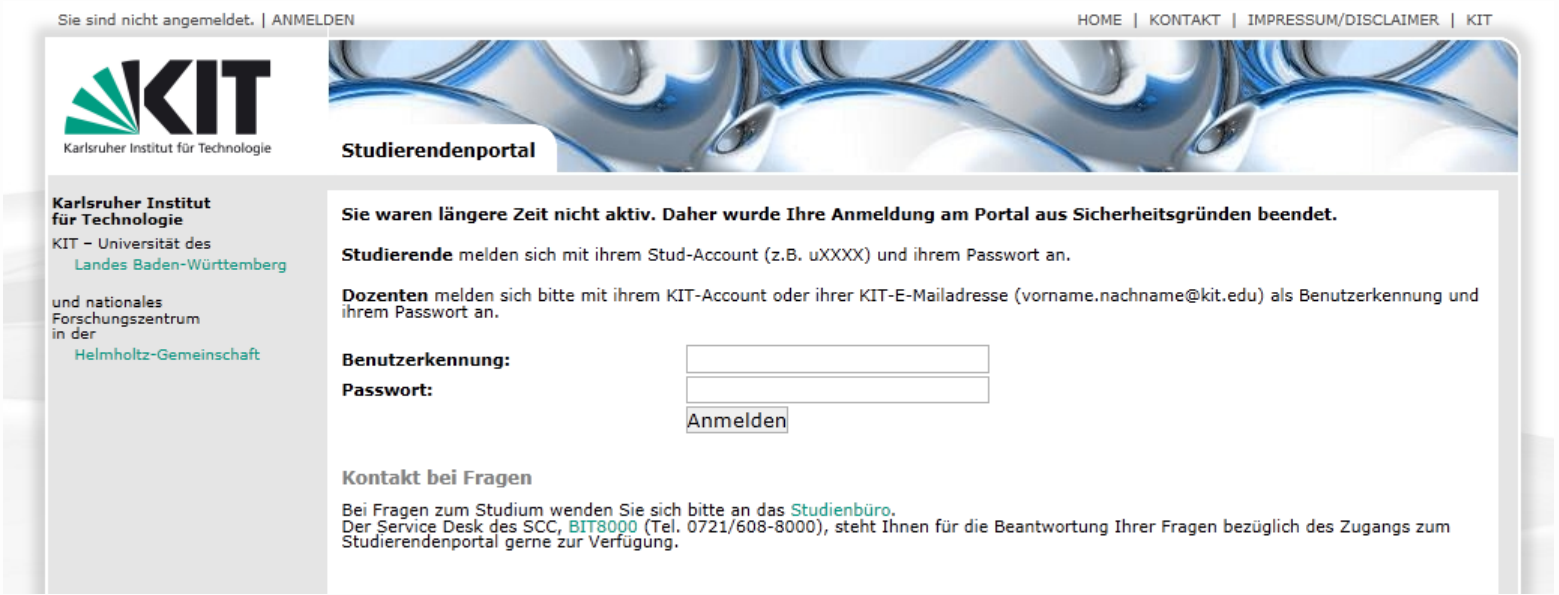
- Leistung: schriftlich
- Dauer: 120 min
- Termin: **Donnerstag, 15.03.2017**
  14:00 – 16:00
- Ort: Geb. 30.46, Hörsaal Neue Chemie
- Anmeldung: Online im Campus-Portal
- Mitzubringen: **Studentenausweis**
- Hilfsmittel: 10 Seiten mit eigener Handschrift

Online-Prüfungsanmeldung notwendig! (https://studium.kit.edu/)

Hinweis

Es ist eine **Online-Prüfungsanmeldung** notwendig!

https://studium.kit.edu/



- Inhalt — Fragen **vorab** per E-Mail an Clemens.Reichmann@vector.com
- Termin — Freitag, 09.03.2018 16:00-18:00 Uhr
- Ort — Vector-Informatik, Philipp-Reis-Str.1, Karlsruhe

→ Haben Sie Interesse an einer solchen Veranstaltung?

1. **Zielsetzung** der Vorlesung
2. **Software Engineering**
   Grundlagen wie Begriffe, Prozesse, Vorgehensmodell, Methode, …
3. **Anforderungsmanagement**
   Anforderungen erfassen und managen
   Werkzeuge/Methodik: SysML, Elektrik/Elektronik-Architektur
4. ~~**Projektmanagement**~~
   ~~wie man ein Projekt organisiert, überwacht, …~~
5. **Softwareentwurf**
   modularer Entwurf, objektorientierter Entwurf, UML2

6. Entwurfsmuster

typische Lösungen

7. Implementierung und Werkzeuge

rund um die Programmierung

8. Refactoring

Qualität beim Programmieren/Modellieren

9. Software-**Wiederverwendung**

Wiederverwenden von SW
(aus Bibliothek, Framework, Altbestand, …)

Empfehlungen

– Sommerville, Ian: **Software Engineering.**

  10. Auflage, *Addison-Wesley*, 2015

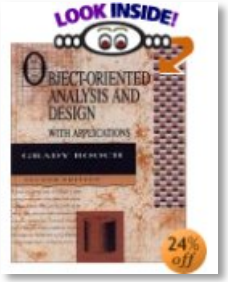
– Balzert, Helmut: **Lehrbuch der Softwaretechnik.**

  Teil 1: Software-Entwicklung;
  Teil 2: Software-Management, Qualitätssicherung, Unternehmensmodellierung).

  2. Auflage, *Spektrum Akademischer Verlag*, 2008/2011.

weitere…

– Wirfs-Brock; Wiklerson; Wiener:
Designing Object-Oriented Software.
Prent. Hall, 1990



– Booch, Grady: **Object-Oriented Analysis and Design with Applications.**
Addison-Wesley Professional, 1993

– Ivar Jacobson et. al.: **Das UML-Benutzerhandbuch.**
2. Auflage, *Addison-Wesley*, März 2007
(**ISBN: 3827325706**)

(weitere…)

- **Erfolgreich im Team**, Christoph V. Hang, dtv Verlag, 1998
- **Projekte zum Erfolg führen**, Heinz Schelle, dtv Verlag, 1999
- **The Push to Make Software Engineering Respectable**. Describes the essentials of professionalism for software engineers. (G. Pour, M. L. Griss and M. Lutz, IEEE Computer, 33 (5), May 2000)
- **Special Issue on Software Process Diversity**. This special issue includes a number of interesting and useful articles on different types of process for developing software. It also includes articles covering process maturity and the CMM model (Chapter 25). (IEEE Software, 17 (4), July 2000)

- **eXtreme programming - Development through Dialog**. A short introduction that addresses common concerns about extreme programming. (R. C. Martin, IEEE Software, July 2000)
- **A Software Development Process for Small Projects**. Suggests a software process that has been adapted for small companies. (M. L. Russ and J. D. McGregor, IEEE Software, 17 (4), September 2000)
- **Wisdom: A Software Engineering Method for Small Software Development Companies**. A discussion of a 'lightweight' method that brings some elements of systematic software engineering to small organisations who cannot afford complex methods. (N. J. Nunes and J. F. Cunha, IEEE Software, 17 (4), September 2000)

(weitere…)

- **Weaving Together Requirements and Architectures**. A short discussion of how requirements engineering and architectural design can be integrated in an evolutionary development process. (B. Nuseibeh, IEEE Computer, March 2001)
- **Get Ready for Agile Methods, with Care**. A good discussion of the pros and cons of agile methods such as extreme programming by a leading software engineering practitioner and researcher. (B. Boehm, IEEE Computer, 35 (4), January 2002)
- **Software System Engineering: A Tutorial**. There are few articles published on this important topic. This is a good overview and introduction. (R. H. Thayer, IEEE Computer, 35 (4), April 2002)

(weitere…)

- **The Agile Methods Fray**. A follow-up to Boehm's January article presented as a discussion between Boehm and DeMarco. (T. DeMarco and B. Boehm, IEEE Computer, 35 (6), June 2002)
- **A Reference Model for Requirements and Specifications** (C. A. Gunter, E. L. Gunter, M. Jackson and P. Zave, IEEE Software, 17 (2), March 2000) Requirements Management: The search for Nirvana (D. J. Reifer, IEEE Software, 17 (3), May 2000)
- **Requirements that Handle IKIWISI, COTS and Rapid Change**. An interesting short article on how requirements engineering has to evolve to cope with different types of software development. (B. Boehm, IEEE Computer, 33 (7), July 2000)

(weitere…)

- **Developing Groupware for Requirements Negotiation**: Lessons Learned. A description of a support systems based around Boehm's win-win approach for teams who are negotating system requirements (B. Boehm, P. Grunbacher, R. O. Briggs, IEEE Software, 18 (3), May/June 2001)
- **Where do Requirements Come From**. A short description of a new approach to requirements discovery using stimuli to creative thinking. (N. Maiden and A. Gizikis, IEEE Software, 18 (5), September/October 2001)
- **Correctness by Construction: Developing a Commercial Secure System**. Discusses the use of formal specification in developing a secure system (A. Hall and R. Chapman, IEEE Software, 19 (1), January/February 2002)

(weitere…)

- **What do you mean by COTS**. The use of COTS systems, discussed in Chapter 14, is likely to be increasingly important. This is an interesting clarification of the sometimes-confusing terminology in this area. (D. Carney and F. Long, IEEE Software, 17 (2), March 2000) Redesigning Air Traffic Control: An exercise in software design (D. Jackson and J. Chapin, IEEE Software, 17 (3), May 2000)
- **Real-time Distributed Object Computing**: An Emerging Field. A special section discussing the use of distributed objects in real-time systems (E. Shokri and P. Sheu, IEEE Computer, 33 (6), June 2000)
- **Diversity in Reuse Processes**. Describes four different successful approaches to achieve software reuse. (M. Morisio, C. Tully and M. Ezran, IEEE Software, 17 (4), July 2000)

- **Component-based Enterprise Frameworks**. A special issue of the CACM with a range of papers on frameworks. (Comm ACM, 43 (10), October 2000)
- **Tracking Usability Issues**: To Bug or Not to Bug. A discussion of how usability problems that arise in a system can be tracked and integrated with other problem reporting systems. There are conflicting viewpoints on whether these issues should be classed as system bugs (C. Wilson and K. Coyne, ACM Interactions, May/June 2001, 8 (3))
- **COTS-based Systems Top 10 List**. A list of hypotheses about COTS-based systems which highlights why this approach may not be as cheap nor as effective as enthusiasts suggest (V. R. Basili and B. Boehm. IEEE Computer, 34 (5), May 2001)

(weitere…)

- **Accelerating Development with Agent Components**. One of a number of articles on very rapid software development (software engineering on internet time) which proposes the use of agents for more rpaid delivery. The problems of verification and validation of agent-based systems are not covered (and these are big problems). (M. Griss and G. Pour, IEEE Computer, May 2001, 34 (5))
- **Middleware Challenges Ahead**. A good summary of the features that next-generation middleware will have to provide to support new types of distributed application such as ubiquitous computer systems. (K. Geihs, IEEE Computer, June 2001, 34 (6))

(weitere…)

- **Comm. ACM. Special Issue on Aspect-oriented Programming**. This special issue includes an introduction to aspect-oriented programming - an approach to program design that may have a major impact on future software development (Comm. ACM, 44 (10), October 2001)
- **Will UML 2.0 be agile or awkward**. A plea for the revision of UML expected in 2002 to be lean rather than flabby. (COmm. ACM, 45 (1), January 2002)
- **The Case for Reflective Middleware**. Discusses a new approaches to the construction of middleware for distributed systems. (F. Kon, F. Costa, G. Blair and R. H. Campbell. Comm. ACM, 45 (6), June 2002)

- **Can you Trust Software Capability Evaluations**. Discusses practical problems with using the CMM model to assess contractor capability. (E. O' Connell and H. Saiedian, IEEE Computer, 33 (2), February 2000) Applying CMM Project Planning practices to Diverse Environments. Discusses how the Capability Maturity Model can be adapted to cope with development environments that are different from the large, long-lifetime project developments for which the model was designed. (D. L. Johnson and J. G. Brodman, IEEE Software, 17 (4), July 2000)

- **Extreme Collaboration**. Discusses an approach used at NASA's Jet Propulsion Laboratory for rapid aerospace design. This involves a design team working together in a large room with lots of public displays. (G. Mark. Comm. ACM, 45 (6), June 2002)

– **How Internet Companies Negotiate Software Quality**. An article discussing the difficulties of applying conventional approaches to software quality management when there are very severe schedule pressures and a need for rapid delivery. (R. Baskerville et al., IEEE Computer, May 2001, 34 (5))
There have been major revisions of capability maturity models discussed in Chapter 25 with a new integrated model suite published by the Software Engineeirng Institute. Full information can be accessed at http://www.sei.cmu.edu/cmmi/ but I hope to find something more accessible soon.

Fragen?