Einstiegsliteratur Proseminar „Werkzeuge und Methoden der Software-Analyse“

Block „Methoden“:

- **M1: Abstrakte Interpretation: Grundlagen**
  Patrick Cousot:
  Abstract Interpretation: Achievements and Perspectives.
  Proceedings of the SSGRR 2000 Computer & eBusiness International Conference, 2000

- **M2: Abstrakte Interpretation: Domains**
  Stefan Bygde:
  Abstract Interpretation and Abstract Domains – with special attention to the congruence domain.
  Mälardalen University Master Thesis, 2006

- **M3: Predicate Abstraction**
  Kapitel 15: Jhala R., Podelski A., Rybalchenko A. (2018) Predicate Abstraction for Program Verification. *Aus*: Clarke E., Henzinger T., Veith H., Bloem R. (eds) Handbook of Model Checking. Springer, Cham

- **M4: Counterexample-Guided Abstraction Refinement**
  Kapitel 13: Dams D., Grumberg O. (2018) Abstraction and Abstraction Refinement. *Aus*: Clarke E., Henzinger T., Veith H., Bloem R. (eds) Handbook of Model Checking. Springer, Cham

- **M5: Fuzzing**
  Alexandre Rebert, Sang Kil Cha, Thanassis Avgerinos, Jonathan Foote, David Warren, Gustavo Grieco, David Brumley:
  Optimizing Seed Selection for Fuzzing.
  USENIX Security Symposium 2014: 861-875

- **M6: Delta-Debugging**
  Andreas Zeller:
  Yesterday, My Program Worked. Today, It Does Not. Why?
  ESEC / SIGSOFT FSE 1999: 253-267

- **M7: Explicit-State Model Checking**
  Kapitel 5: Holzmann G.J. (2018) Explicit-State Model Checking. *Aus*: Clarke E., Henzinger T., Veith H., Bloem R. (eds) Handbook of Model Checking. Springer, Cham

- **M8: Bounded Model Checking**
  Edmund M. Clarke, Armin Biere, Richard Raimi, Yunshan Zhu:
  Bounded Model Checking Using Satisfiability Solving. Formal Methods in System Design 19(1): 7-34 (2001)

- **M9: Combinatorial Testing**
  Changhai Nie, Hareton Leung:
  A survey of combinatorial testing.
  ACM Comput. Surv. 43(2): 11:1-11:29 (2011)

- **M10: Advanced Coverage Criteria**
  Teile aus: J. Edvardsson. A survey on automatic test data generation. In Proceedings of the Second Conference on Computer Science and Engineering in Linköping, pages 21–28. ECSEL, October 1999
  und
  https://www.st.cs.uni-saarland.de/edu/testingdebugging10/slides/04-AdvancedCoverageCriteria.pdf

- **M11: Softwarequalitäts-Standards**
  MISRA-C:2004 Standard

- **M12: Symbolic Model Checking**
  Sagar Chaki, Arie Gurfinkel:
  BDD-Based Symbolic Model Checking.
  Handbook of Model Checking 2018: 219-245

Block „Werkzeuge“:

- **W1: Valgrind**
  Nicholas Nethercote, Julian Seward:
  Valgrind: a framework for heavyweight dynamic binary instrumentation.
  PLDI 2007: 89-100

- **W2: Clang Static Analyzers / Sanitizers**
  Konstantin Serebryany, Derek Bruening, Alexander Potapenko, Dmitriy Vyukov:
  AddressSanitizer: A Fast Address Sanity Checker. USENIX Annual Technical Conference 2012: 309-318
  und evtl. weitere Papervon K. Serebryany zu MemorySanitizer und ThreadSanitizer

- **W3: SPIN**
  Gerard J. Holzmann:
  The Model Checker SPIN.
  IEEE Trans. Software Eng. 23(5): 279-295 (1997)

- **W4: SonarQube**
  Javier García-Munoz, Marisol García-Valls, Julio Escribano-Barreno:
  Improved Metrics Handling in SonarQube for Software Quality Monitoring.
  DCAI 2016: 463-470

- **W5: PMD**
  Markus Aderhold, Artjom Kochtchi:
  Tailoring PMD to Secure Coding.

Technical Report TUD-CS-2013-0245, TU Darmstadt, 2013

- **W6: Frama-C**
  Florent Kirchner, Nikolai Kosmatov, Virgile Prevosto, Julien Signoles, Boris Yakobowski:
  Frama-C: A software analysis perspective.
  Formal Asp. Comput. 27(3): 573-609 (2015)

- **W7: KeY**
  Wolfgang Ahrendt, Sarah Grebing:
  Using the KeY Prover.
  Deductive Software Verification 2016: 495-539

- **W8: Polyspace**
  Dr. Alain Deutsch, Klaus Wissing.
  New test approach for embedded applications.
  Software Engineering 2007: 127-136

- **W9: SMACK**
  Zvonimir Rakamaric, Michael Emmi:
  SMACK: Decoupling Source Language Details from Verifier Implementations.
  CAV 2014: 106-113