Advanced Software Engineering: Non-Functional Aspects in Software Engineering

Content:
Software quality is a term which describes how well a software conforms to its requirements and (users' expectation), such as, performance, safety, security, reliability, availability, and usability. With the increasing pervasiveness of software in business- and safety-critical applications, ensuring the quality of a software is gaining considerable importance. Examples of its applications are online banking, e-commerce, online booking and reservation, car and airplane control systems, remote surgery, high availability systems, such as e-mail servers, or adaptive systems, such as cloud computing controllers.

Software quality verification and forecasting are branches of software engineering concerned with the assurance of requirements satisfaction during both development and production phases. A broad range of techniques is involved in verification and forecasting. This seminar will focus on:

- Self-adaptive Systems.
- Highly Dynamic Systems: Systems which are not easily testable.
- Multi resource Environments.
- Systematic testing: methodologies for automatic test case generation, including model-based testing, random testing, mutation testing, search-based techniques, non-functional testing
- Software analysis: Floyd-Hoare axiomatic semantics, symbolic execution, abstract interpretation, anti-patterns detection
- Model-based prediction: workload prediction, software performance monitoring and prediction
- Statistical forecasting and machine learning: online failure prediction, proactive failure management
- Formal verification: software model-checking, probabilistic model-checking, theorem proving, assume-guarantee reasoning

Bachelor students who want to attend this seminar should be interested in critical systems design, formal methods, software performance, statistical forecasting, machine learning, as well as basic notions on logics, probability, testing, object-oriented programming, and a high interest in system analyses tools.

Topics: Topics will include, but are not limited to:

- Multicore CPU Performance Simulators
- Massive Scale Cloud Simulators
- Analyses for Systems with Emergent Properties like Decentralized IoT Systems
- Security Monitoring in DevOps Environments

Language: German or (optional) English

Organizational Matters: Seminar will be on bachelor level with block seminar presentations towards the end of the lecture period. Introduction during 2-3 sessions in the beginning of the term.

Supervisor: André van Hoorn andre.van-hoorn@informatik.uni-stuttgart.de, Steffen Becker steffen.becker@informatik.uni-stuttgart.de