

Seminar Software Qualität

Vorbesprechung

4.07.2014

Agenda

- Teilnahme am Seminar
- Ablauf
- Themenüberblick
- Diskussion

Teilnahme am Seminar

Mail an juergens@in.tum.de

- Themenwünsche (sortiert nach Priorität)
- Motivation für Seminarteilnahme u. Themen (<1 Seite)
 - Stand im Studium, gewünschter Seminartyp (mit IN-Nummer)
 - Programmier-Erfahrung (Arbeit, Praktikum, Open-Source-Projekt, ...)
 - Motivation an Seminar

Zeitplan

- **Mail an uns bis Dienstag, 8.7.**
- Rückmeldung an Euch über Matching-System

Ablauf

Vorveranstaltung: Oktober

- Überblick Software Qualität
- Literaturrecherche
- Effektiv Präsentieren

Seminar: Dezember

- Block: 2 Tage mit 5-6 Vorträgen pro Tag
- Zeitraum: 15.12.-17.12.

Seminararbeit und Vortrag

Abgaben

- Gliederung: 31.10.2014
- Finale Version Seminararbeit: 2 Wochen vor Vortrag
- Folien Vortrag 3 Wochen vor Vortrag

Probenvortrag (verpflichtend): 2 Wochen vor Vortrag

Umfang

- Arbeit: max. 15 Seiten (wir stellen Vorlage bereit)
- Vortrag: 25min + 20min Diskussion

Praktischer Teil

- Anwendung des Seminarthemas in der Praxis
- Teil der Präsentation
 - Ergebnisse
 - Erfahrungen
 - Probleme & Grenzen
- Ziel: Analyse von 5 Open-Source Systemen mit unterschiedlichen Qualitätsanalysen

Bewertung

- Seminararbeit
- Ergebnisse Analyse
- Vortrag
- Teilnahme an Diskussion

Themen

1. Ungapped Clone Detection: Wieviel Copy & Paste Programmierung gibt es in Open-Source Systemen?
2. Gapped Clone Detection: Welche Inkonsistenzen gibt es in Klonen, und welche Auswirkungen haben sie?
3. Dynamische Analyse: Welcher Code gehört zu welchem Feature?
4. Test-Coverage: Welcher Code wird (nicht) getestet?
5. Bewertung von Wiederverwendung: Was wird verwendet und wie? Analyse und Vergleich der Bibliotheksnutzung in Open Source Systemen.
6. Dynamik von Wiederverwendung: Analyse der Bibliotheksnutzung über den Lebenszyklus eines Open Source Systems.

Themen

7. Verfügbarkeitsanalysen von Systemen
8. Gapped Clone Detection: Wie viele Klone gibt es in Open Source Unit Tests und worin unterscheiden sich die geklonten Tests?
9. Mutation Testing vs. Line Coverage: Wie gut sind Unit Tests in Open Source Projekten wirklich?

Diskussion