

Übungen zu Softwaretechnik: Programmierung und Software-Entwicklung

Aufgabe 12-1

Verifizieren und Validieren

Präsenz

- a) Verifikations- und Validierungstechniken können statisch oder dynamisch stattfinden. Beschreiben Sie den Unterschied zwischen statischen und dynamischen Techniken.
- b) Gebräuchliche Entwicklungsumgebungen wie z.B. IntelliJ, Dreamweaver oder Eclipse bieten umfangreiche Unterstützung für Code Navigation an. Welche Unterstützungsmöglichkeiten fallen Ihnen ein und/oder würden Ihnen sinnvoll erscheinen? Für welche der oben beschriebenen Techniken kann Code Navigation hilfreich sein?
- c) GitLab ist eine webbasiertes Projektmanagement Tool und bietet für vorhandene Projekte die Möglichkeit Milestones festzulegen. Was beschreibt ein Milestone in GitLab, was sollte er enthalten und in welchem Zusammenhang können Milestones in Scrum hilfreich sein? Was könnten in diesem Zusammenhang Issues sein?
- d) In der Vorlesung haben Sie bereits von Git gehört. Git ist ein System zur „verteilten Versionsverwaltung“. Was ist unter dem Begriff „verteilte Versionsverwaltung“ zu verstehen? Außerdem werden für die Verwendung von DVCS (Distributed Version Control Systems) projektabhängige „Workflows“ vorgeschlagen. Warum ist es sinnvoll einen einheitlichen Workflow innerhalb eines Projektes zu verwenden?
- e) GitHub ist ein Online-Dienst, der unter Verwendung von Git die Verwaltung von open-source Software unterstützt. Für die Verwendung der von GitHub bereitgestellten Versionsverwaltung wird der „GitHub Workflow“ vorgeschlagen, dabei werden sechs Arbeitsschritte definiert:
 1. Create a branch (from master)
 2. Add commits (to your branch)
 3. Open a pull request
 4. Discuss and review your code
 5. Deploy
 6. Merge

Was bezwecken die angegebenen Arbeitsschritte?

- f) Was versteht man in diesem Zusammenhang unter „continuous integration“?

- a) Was versteht man grob unter der Verwendung von formalen Techniken zur Verifikation von Programmen und was ist deren Ergebnis?
- b) Erläutern Sie Pro und Contra der Verwendung von formalen Techniken zur Verifikation von Programmen.

Aufgabe 12-3**Testen***Präsenz*

- a) Was versteht man in der Softwareentwicklung unter Testabdeckung und warum kann selbst bei einer vollständigen Testabdeckung die Korrektheit eines Systems nicht garantiert werden?
- b) Um ungewünschtes Verhalten des Systems weitestgehend zu vermeiden ist es üblich, dass neben dem eigentlichen Programmcode auch Tests geschrieben werden. Welche Möglichkeiten der Klassifikation von Tests können Sie sich vorstellen und was beschreiben die folgenden Testarten?
 - Unit Test
 - Component testing
 - Inetgration Test
 - Akzeptanztests

Welche weiteren Testarten & Strategien fallen Ihnen ein?

- c) In Zusammenhang mit Testen von Software fallen immer wieder Begriffe wie „Mock“, „Stub“, „Fake Object“ und „Dummy Object“. Geben Sie eine knappe Definition der Begriffe an und erläutern sie knapp welchen Zweck sie erfüllen.