

Softwarequalitätssicherung

Übung 10

Aufgabe 1

In der Vorlesung haben Sie Versionsmanagement und Konfigurationsmanagement kennengelernt.

- a) Geben Sie jeweils eine Definition an,
- b) beschreiben Sie, wie Versionsmanagement bzw. Konfigurationsmanagement in einer Softwareentwicklung verwendet werden und
- c) wie sich Versionsmanagement bzw. Konfigurationsmanagement auf die Qualität eines SW-Produktes auswirken können.
- d) Nennen Sie Best Practices in Versionsmanagement und erläutern Sie diese beispielsweise unter Angaben von typischen Projektsituationen.

Aufgabe 2

Im Rahmen der Vorlesung haben wir uns sehr ausführlich mit Git beschäftigt. In dieser Aufgabe sollen Sie nun zeigen, dass Sie praktisch mit Git umgehen können. In der Übung sollten Sie live die Verwendung von Git nach dem folgenden Ablauf zeigen können:

- a) Anlegen eines Verzeichnisses <Name-Verzeichnis> mit drei sehr einfachen Java-Dateien <Name-File1>, <Name-File2> und „*NotInGitFile.txt*“. (Die Namen <...> können Sie frei wählen).
- b) Anlegen eines Git Repos im Verzeichnis <Name-Verzeichnis>
- c) Abfragen des Status des Git Repos
- d) Hinzufügen der beiden Dateien zu ihrem lokalen Git Repo
- e) Abfragen des Status (Sie sollten den Status immer wieder zwischendurch abfragen und diesen jeweils erläutern können)
- f) Die Datei mit dem Namen *NotInGitFile.txt* soll nicht in Git verwaltet werden.
- g) Zusätzlich sollten alle Dateien, welche auf „.pdf“ enden, nicht in ihrem Repo verwaltet werden.
- h) Modifizieren Sie nun <Name-File1>. Untersuchen Sie die Unterschiede zwischen der ursprünglichen Datei <Name-File1> und der geänderten <Name-File1> Datei, ohne die Änderungen zu stagen.

- i) Stagen Sie nun Ihre Änderungen und untersuchen Sie erneut die Unterschiede.
- j) Committed Sie Ihre Änderung mit einem passenden Kommentar.
- k) Lassen Sie sich nun das Log all Ihrer bisherigen Tätigkeiten anzeigen und erläutern Sie diesen.
- l) Führen Sie eine weitere Änderung an <Name-File1> durch, stagen und committed Sie diese.
- m) Lassen Sie sich das Log anzeigen.
- n) Checken Sie nun die initiale Version Ihres Repos aus und betrachten Sie die Datei <Name-File1> in einem Editor.
- o) Lassen Sie sich das komplette Log anzeigen. Worauf zeigt Master bzw. HEAD?
- p) Wechseln Sie nun wieder zur vorherigen Version zurück (die Version auf die Master zeigt).

Da wir in der Vorlesung noch nicht über Branches und Remote Repos gesprochen haben, ist der Rest dieser Übung für diesen Termin optional. Diejenigen, die schon Erfahrung mit Git haben, können die restlichen Aufgaben gerne bearbeiten.

- q) Erzeugen Sie nun einen Branch mit dem Namen „NewFeature“ und wechseln Sie zu dem Branch „NewFeature“.
- r) Führen Sie nun eine Änderung an beiden Dateien (<Name-File1>, <Name-File2>) durch und kopieren sie eine PDF-Datei in Ihr Arbeitsverzeichnis.
- s) Stagen und committed Sie die Änderungen mit einem passenden Kommentar.
- t) Wie sieht zu diesem Zeitpunkt die „Struktur“ ihre Git-Repos aus?
- u) Mergen Sie nun die Änderungen aus dem Branch „NewFeature“ in den Master Branch. Was müssen Sie dafür tun?
- v) Wie sieht nach einem erfolgreichen Merge die Struktur Ihres Git Repos aus?
- w) Legen Sie sich nun auf GitHub oder gitlab.rlp.net einen Account an (falls noch nicht vorhanden).
- x) Legen Sie ein Remote Repo an.
- y) Verknüpfen Sie das Remote Repo mit Ihrem lokalen Repo und legen Sie die Zuordnung des Master Branches fest.
- z) Pushen Sie Ihre lokales Repo auf das Remote Repo.
- aa) Ab diesem Punkt ist ihnen die weitere Bearbeitung der Aufgabe freigestellt. Sie könnten beispielsweise Änderungen lokal vornehmen oder ihr lokales Repo Eclipse bekannt machen oder auf dem Remote Repo Änderungen durchführen und den lokalen Status anzeigen lassen oder ... Sie sollten vor jeder Aktion immer das erwartete Verhalten mit dem tatsächlichen Verhalten vergleichen.