

Übungen zur Vorlesung Softwaretechnologie

- Wintersemester 2013/14 -

Dr. Günter Kniesel

Übungsblatt 1

Zu bearbeiten bis: 27.10.2013

Bitte fangen Sie **frühzeitig** mit der Bearbeitung an, damit wir Ihnen bei Bedarf helfen können. Checken Sie die Lösungen zu den Aufgaben bitte in Ihr SVN-Repository ein, „Erklärungen“ bitte als Textdatei. Fragen zu Übungsaufgaben respektive zur Vorlesung können Sie auf der Mailingliste swt-tutoren@lists.iai.uni-bonn.de, bzw. swt-vorlesung@lists.iai.uni-bonn.de stellen.

Aufgabe 1. Erste Schritte mit Subversion (5 Punkte)

- Im Repository Ihrer Gruppe befindet sich ein Projekt namens *HelloTutor*. Checken Sie das Verzeichnis aus, ergänzen Sie das darin enthaltene Programm so, dass es Ihren Namen mit ausgibt und committen Sie anschließend Ihre Änderungen.
- Legen Sie nun ein neues Java-Projekt an und darin ein Programm, das alle durch 7 oder 13 teilbare Zahlen zwischen 10 und 40 (jeweils inklusive) ausgibt. Sichern Sie dieses in Ihrem SVN-Repository. Dazu können Sie den Befehl *Team->Share Project...* aus dem Kontextmenü des Projektes verwenden.

Aufgabe 2. Dateioperationen mit SVN (6 Punkte)

- Erzeugen sie in Eclipse ein neues Projekt „Dateien“ und legen Sie darin einen Ordner „Nummer 1“ mit einer Datei „Datei 1.txt“ an. Importieren Sie nun das Projekt ins SVN (*Team->Share project...*).
- Fügen Sie in dem Ordner eine neue Datei „Datei 2.txt“ hinzu. Vergleichen Sie die Symbole der beiden Textdateien im *Package Explorer*. Was fällt auf? Welche Bedeutung haben die jeweiligen Symbole? Welche Schritte müssen Sie unternehmen, damit sich „Datei 2.txt“ im gleichen Zustand wie „Datei 1.txt“ befindet?
- Benennen Sie nun „Datei 2.txt“ in „Datei 2a.txt“ um (Kontextmenü: *Refactor->Rename*). Was bemerken Sie? Was bedeutet dies für Dateioperationen bei SVN?
- Öffnen Sie nun den Projektordner mit einem Dateiverwaltungsprogramm (z.B. Finder oder Windows Explorer)¹. Benennen Sie nun hier, außerhalb von Eclipse, „Datei 1.txt“ in „XX.txt“ um. Wechseln Sie zu Eclipse und wählen Sie aus dem Kontextmenü des Projekts den Befehl „Refresh“ aus. Was bemerken Sie im Unterschied zu Aufgabenteil c) ? Welche Änderungen zeigt Ihnen die Synchronisations-Ansicht des Projekts (*Team->Synchronize*)?

¹ Den Pfad des Projekts finden Sie in den Eigenschaften des Projektes bei „Resource“ unter „Location“.

- e) Committed Sie das Projekt. Benennen Sie nun außerhalb Eclipses den Ordner „Nummer 1“ in „Nummer 2“ um und aktualisieren Sie die Dateiansicht in Eclipse. Was bemerken Sie nun?
- f) Wechseln Sie in die Synchronisations-Ansicht des Projekts. Was bemerken Sie nun? Was passiert bei einem Update? Was können Sie nun tun, um das Projekt zu „reparieren“?

Aufgabe 3. *Etiketten und Zweige* (15 Punkte)

Neben dem Hauptstamm eines mit SVN verwalteten Projektes (genannt „trunk“) gibt es noch die Konzepte der Etiketten („tags“) und Zweige („branches“).

- a) Erklären Sie, welchen Zweck *tags* und *branches* erfüllen und wie diese in SVN angelegt werden. Gibt es auf technischer Ebene im SVN Repository einen Unterschied zwischen *trunk*, *tags* und *branches*?
- b) Legen Sie von Ihrem Projekt in Aufgabe 1b einen Tag an und benennen Sie diesen sinnvoll.
- c) Erzeugen Sie einen Branch, checken Sie diesen aus und verändern Sie das verzweigte Projekt so, dass zusätzlich auch durch 5 teilbare Zahlen ausgegeben werden. Checken Sie das modifizierte Branch-Projekt wieder in das SVN Repository ein.
- d) Ergänzen Sie das Programm im *trunk* so, dass durch 6 teilbare Zahlen übersprungen werden und committen Sie das Projekt. Fügen Sie nun den *branch* aus Aufgabenteil c) wieder mit dem *trunk* zusammen („mergen“) und committen sie das Projekt erneut.
- e) Warum steigt beim Kopieren von Zweigen innerhalb des SVN Repositories der Speicherbedarf desselben – solange keine Modifikationen committet werden – nur minimal an?

Aufgabe 4. *SVN Properties* (4 Punkte)

In Subversion lassen sich Dateien und Verzeichnisse um Meta-Informationen, die so genannten Eigenschaften („Properties“) ergänzen. Geben Sie für die folgenden Eigenschaften wieder jeweils deren Bedeutung und einen beispielhaften Wert an:

- a) `svn:ignore` (für Verzeichnisse)
- b) `svn:mime-type` (für Dateien)
- c) `svn:keywords` (für Dateien)
- d) `svn:externals` (für Verzeichnisse)

Hinweis: Verwenden Sie zur Bearbeitung der Aufgaben 3 und 4 auch weiterführende Literatur bzw. Web-Suchanfragen.

Aufgabe 5. Erste Schritte mit GIT (12 Punkte)

Nach Kennenlernen von SVN, wollen wir uns mit Git beschäftigen und nachvollziehen, was daran anders / besser ist.

Installieren Sie dazu einen Git-Client auf ihrem System, und bearbeiten Sie dann die folgenden Aufgaben. Die Detailerläuterungen in der Aufgabenstellung beziehen sich auf die Arbeit mit SmartGit. Hinweise zur Installation von SmartGit erhalten Sie unter <https://sewiki.iai.uni-bonn.de/teaching/lectures/se/2013/git>

In dem Repository [git://github.com/swt201112/uebungsblatt_1.git](https://github.com/swt201112/uebungsblatt_1.git) steht ein Text der aus drei Abschnitten besteht. Jeder von Ihnen soll *parallel zu den anderen Gruppenmitgliedern* auf seinem eigenen Rechner genau einen Abschnitt bearbeiten. Ordnen Sie sich selbst die Abschnitte zu, indem Sie ihre Vornamen alphabetisch sortieren. Der 1. Vorname bearbeitet den 1. Abschnitt, der 2. Vorname den zweiten Abschnitt, und so weiter. Wenn Sie mehr als 3 Teammitglieder sind, geht es zyklisch weiter, d.h. es gibt mehrere Leute, die den gleichen Abschnitt bearbeiten.

- a) Holen Sie sich unser Repository auf ihren lokalen Rechner, indem sie ein Clone von [git://github.com/swt201112/uebungsblatt_1.git](https://github.com/swt201112/uebungsblatt_1.git) durchführen.
- b) Erstellen Sie nun für sich einen eigenen Branch, wechseln Sie zu dem neuen Branch und führen Sie darin folgende Änderungen durch:
 1. Korrigieren Sie die enthaltenen Rechtschreibfehler in der Datei *briand-kellogg.txt* entsprechend Ihrer vereinbarten Aufgabenteilung pro Abschnitt.
 2. Benennen Sie die Datei *briand-kellogg.txt* in *kellogg-briand.txt* um.
 3. Fügen Sie in der Datei *gedicht.txt* eine neue Strophe hinzu
 4. Comitten Sie die Änderungen in ihr lokales Repository.
- c) Übergeben Sie ihren lokalen Git-Ordner (gepackt) an alle ihre Teamkollegen (E-Mail, USB-Stick, ...)
- d) Mergen Sie die Repositories ihrer Kollegen auf ihren eigenen Branch.
 - Dazu müssen Sie zuerst die Repositories unter Remotes/Manage Remotes als Remote Repository hinzufügen und einen Pull durchführen.
 - Eventuell auftretende Konflikte lösen Sie (in SmartGit) auf, indem sie auf die entsprechende Datei rechtsklicken und „Show Changes“ auswählen. In dem sich öffnenden Fenster manuell die Änderungen durchführen und speichern. Dann ein „Stage“ durchführen und anschließend wieder in Ihr lokales Repository committen.
- e) Vereinbaren Sie, wer in ihrem Team als "Integrator" für die finale Version verantwortlich ist. Schicken Sie ihren Git-Ordner mit dem Zustand nach d) an ihren Integrator.
- f) Der Integrator hat die Aufgabe alle ihre verschiedenen Branches zu einem zusammenzuführen und anschließend seinen Git-Ordner als ZIP-Datei gepackt als Gruppenabgabe in das SVN hochzuladen.