

Subversion

Zu Beginn der Arbeitsphase: **Update** (um auf dem aktuellen Stand aufzubauen)

Nach Abschluss einer sinnvollen Einheit: **Update** (um Konflikte früh zu bemerken), dann **Commit** (damit andere auf deiner Arbeit aufbauen können)

Jeder Commit sollte eine minimale sinnvolle Einheit bilden.

Einerseits tust du niemandem einen Gefallen, wenn du ein kleines Refactoring durchführst und die Veränderungen in mehreren Schritten eincheckst, so dass zwischendurch nicht kompilierbare Revisionen erzeugt werden.

Andererseits tust du auch niemandem einen Gefallen, wenn du einen ganzen Tag auf deinen Änderungen sitzt und am Abend eine riesige Menge Code eincheckst. Niemand hat dann Lust, sich mit deinen Änderungen auseinanderzusetzen und das Finden von Fehlerursachen wird stark erschwert.

Die Entscheidung, wann eine Änderung eingchecked wird, erfordert also Feingefühl.

Wenn du eine Änderung programmiert hast, der Code sauber kompiliert und alle Tests durchlaufen, dann sollte dich in den meisten Fällen nichts von einem Commit abhalten.

Stelle dir zur Orientierung die Frage, ob du deine Veränderungen in einem kurzen Satz zusammenfassen kannst. „Verschiedene Verbesserungen am Netzwerkcode“ ist kein Zeichen für eine logische Einheit. „Verwende short statt int für Port-Nummern in TCPConnectionAdapter“ deutet eher auf ein sinnvolles Changeset hin. Wenn du deine Veränderungen nicht zusammenfassen kannst, hast du wahrscheinlich zu viel auf einmal gemacht.

Tipp: Deine Zusammenfassung kannst du dann auch gleich als Commit-Message verwenden. Deine Teammitglieder werden dir dankbar sein.

Repository-Einrichtung

Die Verbindung zwischen lokalem Projekt und Repository muss auf jedem verwendeten Client einmalig eingerichtet werden. Mit Subclipse ist das gar nicht schwer. Hier die Adresse unseres Repositories:

```
svn+ssh://rzrepository.informatik.uni-hamburg.de/informatik/agis/swt/repositories/SEP_Team2
```