

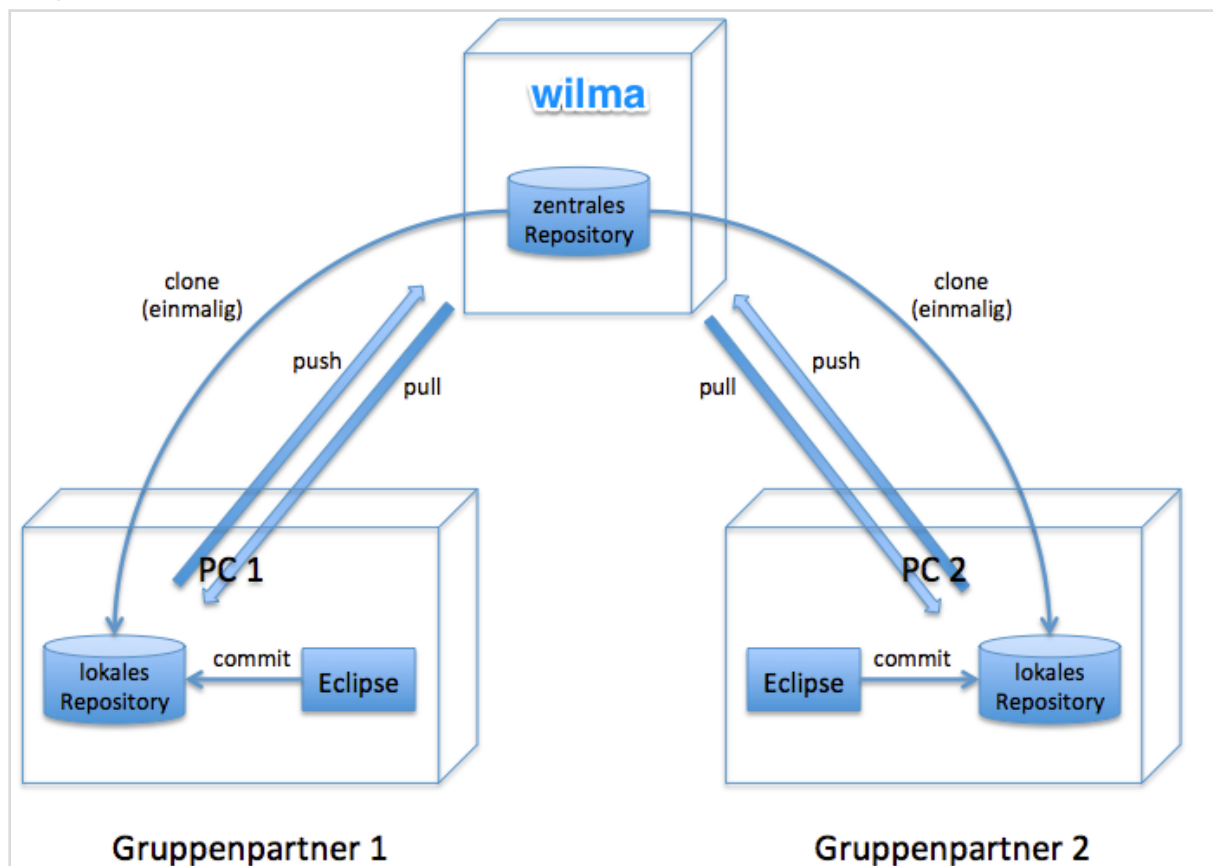
WS20: Arbeiten mit Git

Allgemeines

Zugrunde liegt der Gedanke, an einer zentralen Stelle ein Repository (engl. für "Datenbehälter") zu haben, in dem alle Programme vorliegen, die Sie im Laufe der Vorlesung PR1 erstellen. Auf jedem Rechner, auf dem Sie oder Ihre Gruppenpartner an den Programmen arbeiten möchten, legen Sie ein identisches Repository an. Diese lokalen Repositories können sehr einfach mit dem zentralen Repository synchronisiert werden, so dass Sie und Ihre Partner von überall aus auf den jeweils *aktuellen* Stand zugreifen können. (Das ist mit E-Mails, USB-Sticks etc. nicht so einfach.)

Git (<https://git-scm.com>) ist eine freie Software zur Versionsverwaltung, die ein solches Arbeiten unterstützt.

Die folgende Abbildung zeigt die Synchronisationsmechanismen, die für die Arbeit mit Eclipse und Git benötigt werden:

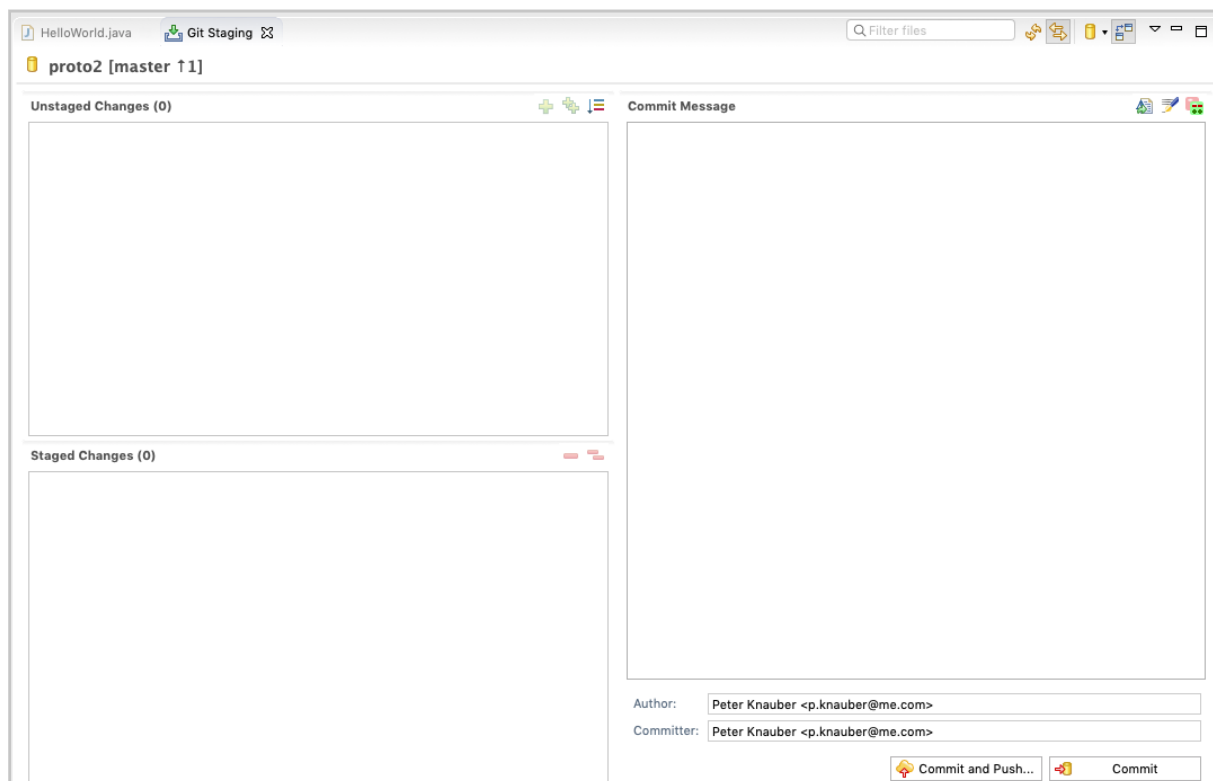


Für die Vorlesung PR1 wurde für Ihre Übungsgruppe ein zentrales Repository mit einem geeigneten Java-Projekt auf einem Rechner *wilma* vorbereitet.

Auf jedem Rechner, an dem Sie oder einer Ihrer Gruppenpartner arbeiten möchten legen Sie zunächst einen Klon (clone) des zentralen Repositories an. Jetzt kann die eigentliche Arbeit losgehen.

Normaler Arbeitsablauf

1. Sprechen Sie sich immer mit Ihren Gruppenpartnern ab: Sie sollten nicht gleichzeitig an derselben Datei arbeiten, weil sonst nur Änderungen eines Partners (wer zuletzt *pusht*) in der Endversion enthalten sind!
2. Bevor Sie mit Ihrer Arbeit beginnen, besorgen Sie sich die aktuellste Version aller Dateien aus dem zentralen Repository: Kontextmenü des Projekts → Team → Pull.
3. Wenn Sie fertig sind, sich an Ihrem Rechner abmelden wollen oder einen stabilen Zwischenstand erreicht haben, schieben Sie Ihre Änderungen in das zentrale Repository:
a) Kontextmenü des Projekts → Team → Commit: Es öffnet sich die sogenannte Staging View:



- Links oben sehen Sie die Dateien, die Sie geändert haben.
- Durch Klicken auf das "+" darüber markieren Sie diejenigen Dateien, die Sie ins zentrale Repository auf *wilma* übertragen möchten.
- Geben Sie im rechten Teilfenster eine kurze aber aussagekräftige Beschreibung des aktuellen Zwischenstands an, zum Beispiel:
 - Programm ist vollständig, aber noch nicht getestet
 - Problem: negative Zahlen funktionieren nicht
 - fertige Version

b) Klicken Sie *Commit und Push...*, damit Ihre Änderungen in Ihr *lokales* Repository (*commit*) und ins *zentrale* Repository auf *wilma* (*push*) übernommen werden.

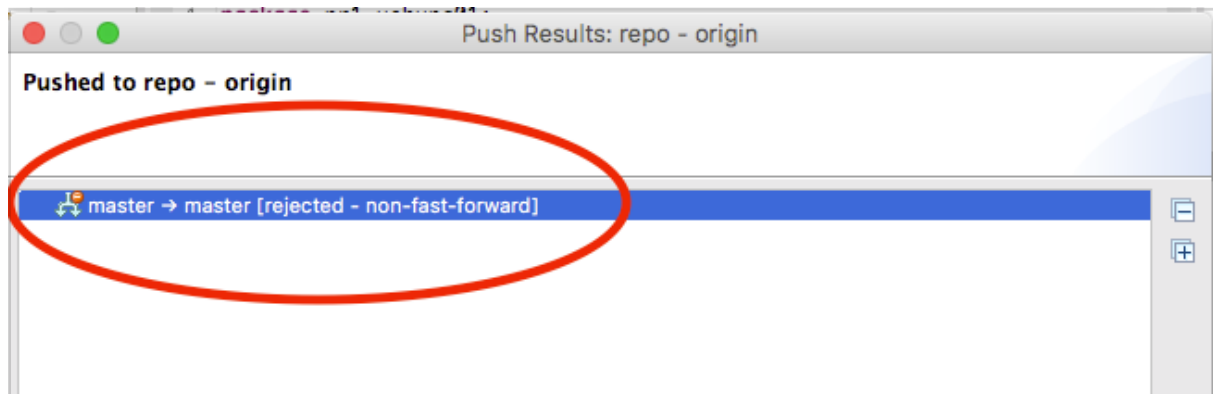
Mögliche Probleme und ihre Beseitigung

Falls Sie Schritt 3 vergessen, können Sie und Ihre Gruppenpartner nicht von anderen Rechnern aus auf Ihre Änderungen zugreifen.

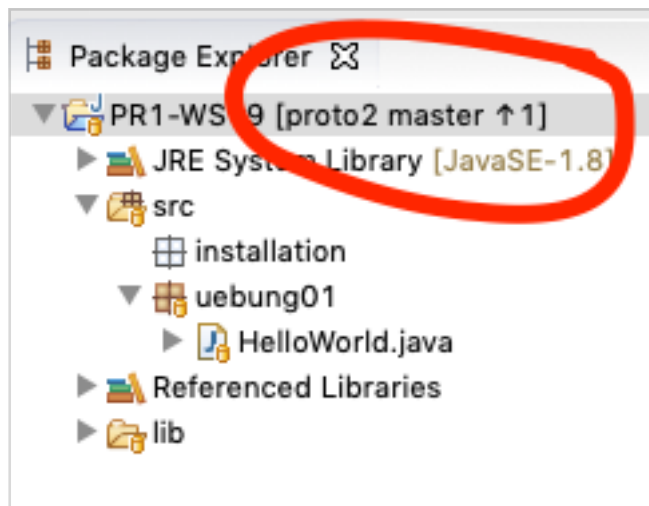
- Sie müssen zurück zu diesem Rechner und die Änderungen *pushen*. Erst dann können sie auf einem anderen Rechner *gepullt* werden.

Falls Sie Schritt 2 vergessen haben oder falls die Absprache mit Ihrem Gruppenpartner nicht funktioniert hat, kann es zu sogenannten "Merge-Konflikten" kommen, weil parallel zueinander verschiedene Änderungen an der gleichen Datei vorgenommen wurden. Ein Konflikt tritt auf, wenn Sie ein *Push* versuchen, aber eine neue Datei oder eine neuere Version einer existierenden Datei auf *wilma* vorliegt als in Ihrem lokalen Repository. Git wird Sie darüber informieren und Sie müssen den Konflikt dann auflösen: Git kann ja nicht wissen, ob die "richtige" Version im zentralen Repository auf *wilma* (und bei Ihrem Partner) oder in Ihrem lokalen Repository liegt.

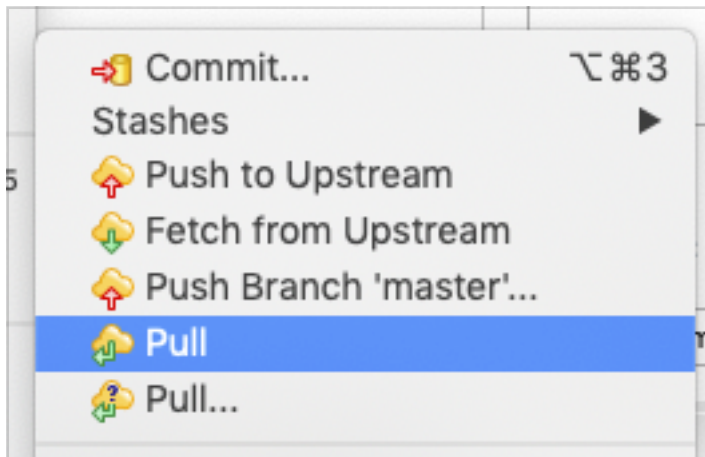
- Sie merken den Konflikt, wenn für das Projekt beim *Push* folgende Meldung erscheint...



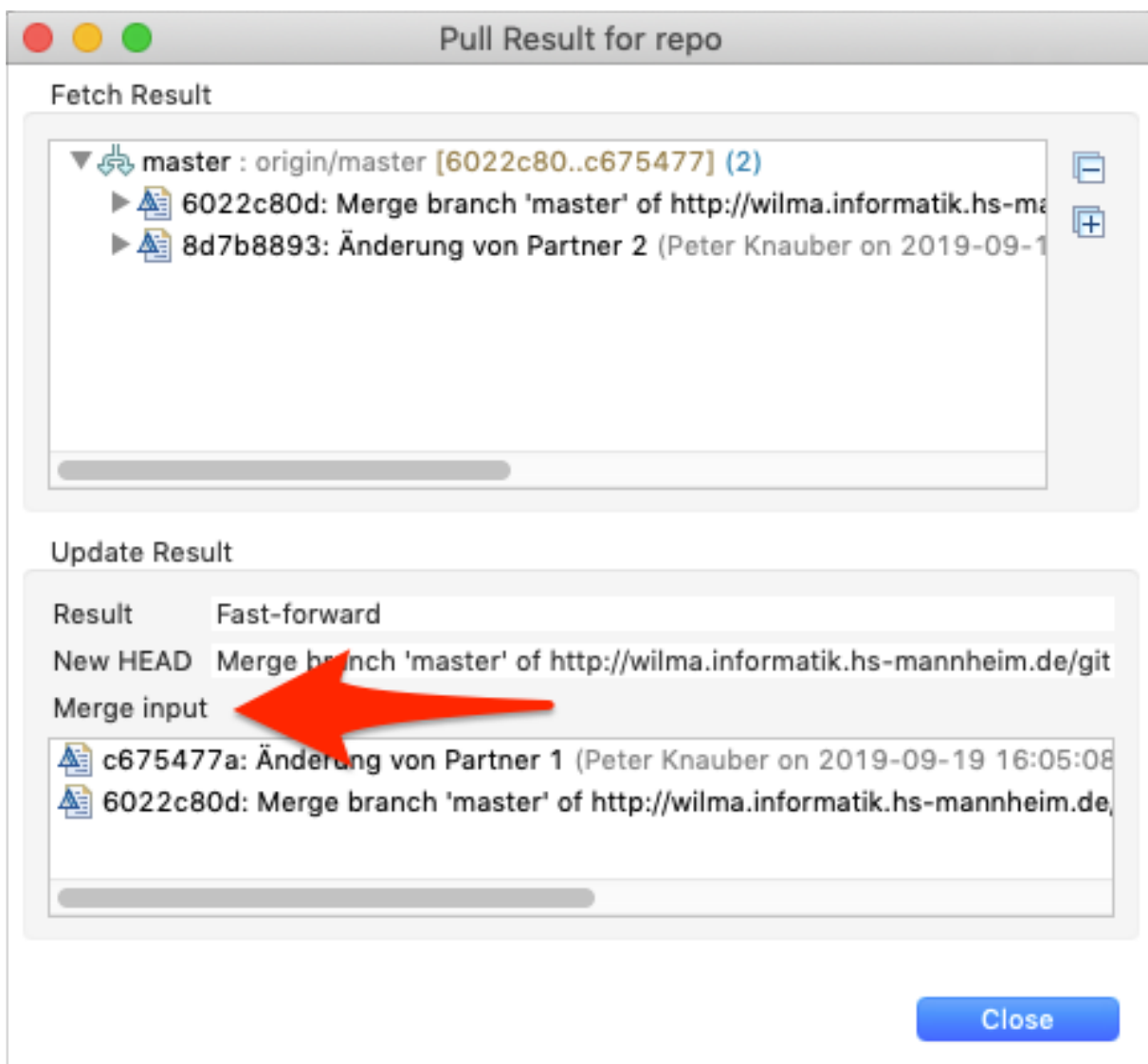
... und im Projekt anschließend angezeigt wird, dass noch lokale Änderungen zu *pushen* sind:



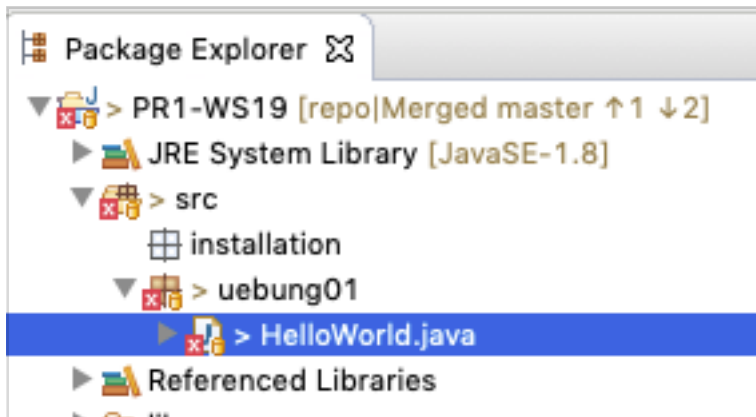
- Führen Sie dann ein *Pull* aus, um die Änderungen aus den Repository auf *wilma* in Ihr Repository zu übernehmen:



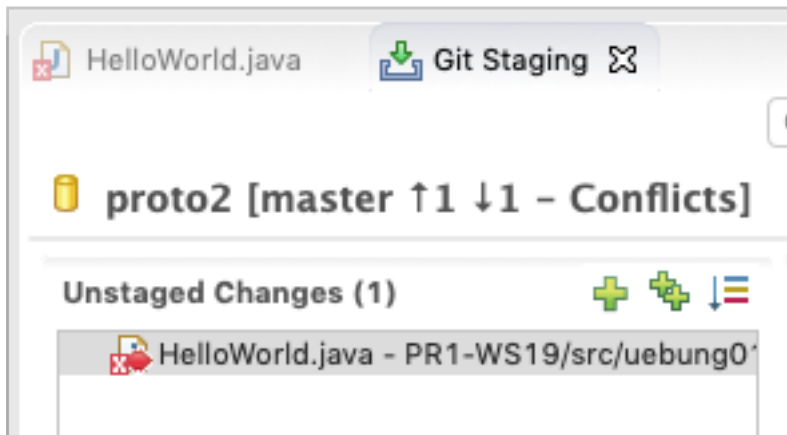
- Sie sehen dabei etwa folgende Meldung, die darauf hinweist, dass Sie die lokale Änderung und diejenige Ihres Partners auf *wilma* "mergen", also zusammenführen, müssen:



- Nach dem Pull finden Sie die Datei (es können auch mehrere sein) mit dem Konflikt im Projekt markiert:



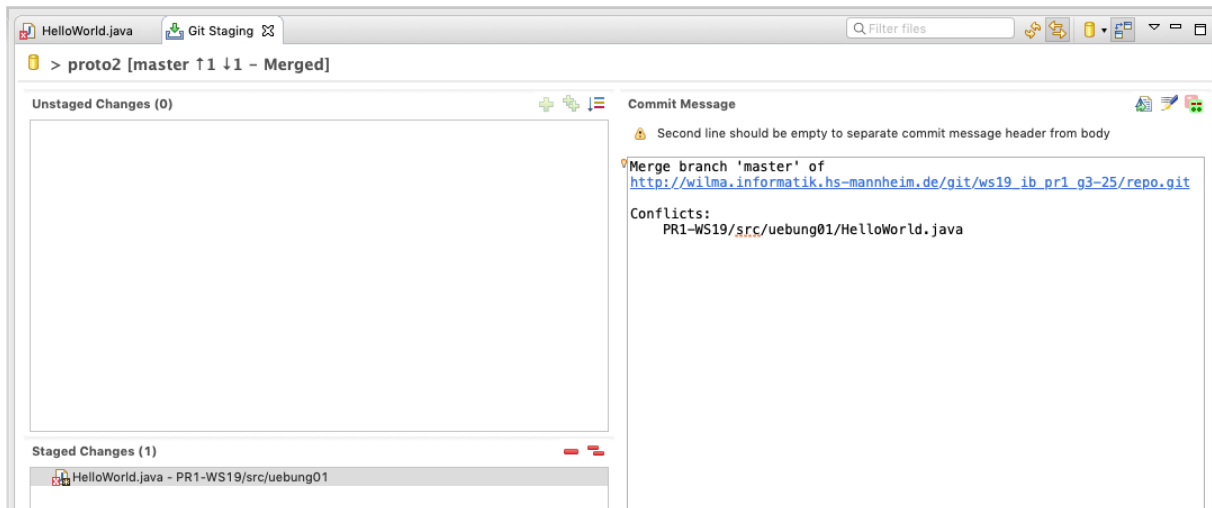
- Die *Staging View* für das Projekt zeigt folgende Information:



- Ein Doppelklick auf diese Datei (in der *Staging View*) zeigt die Konflikte beider Versionen im Vergleich (links Ihre lokale Datei mit beiden Änderungen nacheinander, rechts die Version im Repository auf *wilma*):



- Passen Sie (links) Ihre Datei so an, wie Sie es für richtig halten, speichern Sie diese Änderung und schließen Sie das Vergleichsfenster.
- Klicken Sie wie üblich auf das "+" in der *Staging View*. Dadurch wird automatisch eine *Commit Message* generiert:

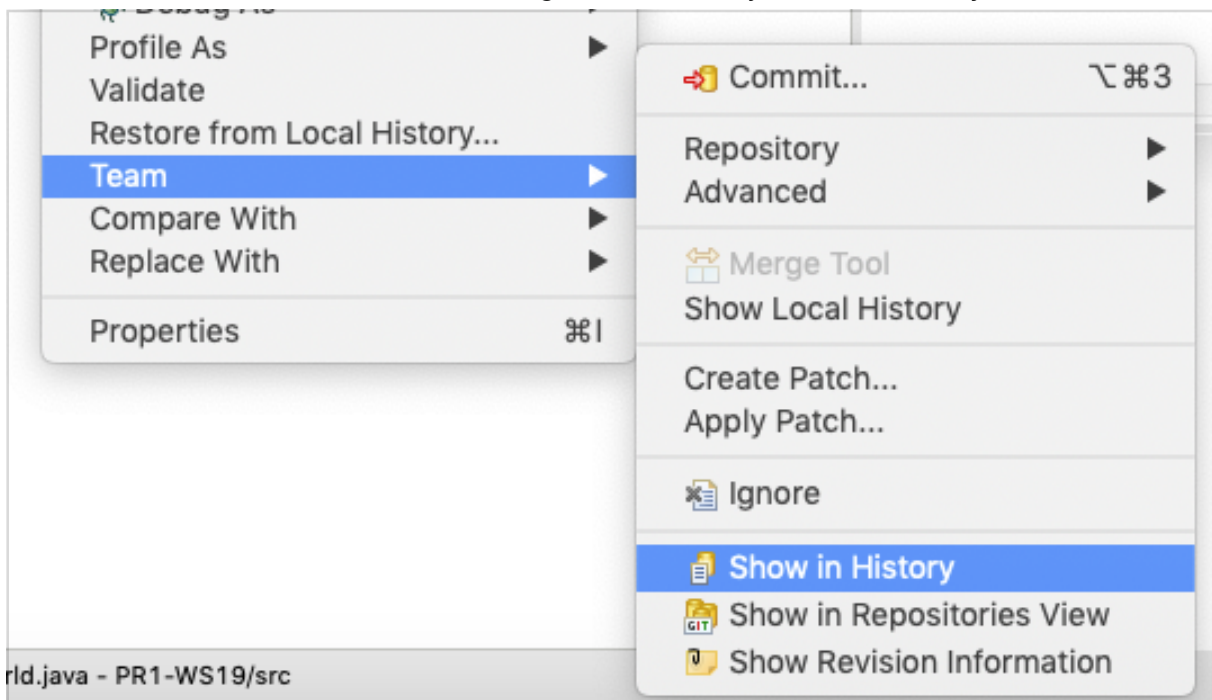


- Jetzt müssen Sie nur noch "Commit und Push..." klicken um die zuletzt gespeicherte Änderung auf *wilma* zu übertragen und den Konflikt als "beseitigt" zu markieren.

Damit sind Sie fertig; Ihre lokale Version und die im zentralen Repository stimmen überein. Auf den anderen Rechnern sollte jetzt ein *Pull* durchgeführt werden!

Information

Wenn Sie Ihre Änderungen im zeitlichen Verlauf sehen oder noch einmal eine alte Version anschauen möchten, dann öffnen Sie sogenannte *History View* für Ihr Projekt:



Das neue Fenster zeigt Ihnen, wer wann welche Änderungen am Projekt vorgenommen hat und könnte zum Beispiel so aussehen (die *Commit Messages* wären im Normalfall ungeschickt und sollen hier nur illustrieren, was passiert ist):

Problems @ Javadoc Declaration Console Git Repositories History

File: PR1-WS19/src/uebung01/HelloWorld.java [repo]

Id	Message
858b123	master origin/master HEAD Programm korrigiert
9c3d72c	Merge branch 'master' of http://wilma.informatik.hs-mannheim.de/git/ws19_ib_pr1_g3-25/repo.git
e316583	neue Änderung von Partner 2 ohne vorheriges Pull
a365af6	Änderung 2 von Partner 1 wieder gelöscht
6022c80	Merge branch 'master' of http://wilma.informatik.hs-mannheim.de/git/ws19_ib_pr1_g3-25/repo.git
8d7b889	Änderung von Partner 2
c675477	Änderung von Partner 1
cfa7c72	Basis
661efb3	Merge branch 'master' of http://wilma.informatik.hs-mannheim.de/git/ws19_ib_pr1_g3-25/repo.git
e729e05	proto2
67d9a3d	Änderung im Clone
89ff1f1	Änderung im Clone
f8d4526	initiale Version

#PR1