Gestion de version et de configuration : de svn à git

Philippe Collet

Licence 3 MIAGE – S6

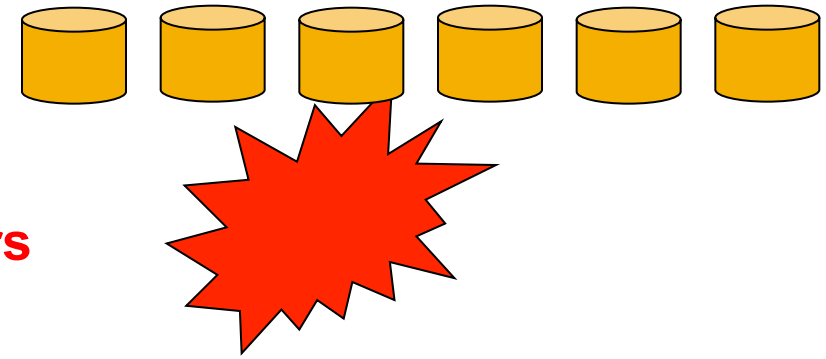
2014-2015

http://miageprojet2.unice.fr/User:PhilippeCollet/Projet_de_d%c3%a9veloppement_2014-2015

Université Nice SOPHIA ANTIPOLIS

Motivations

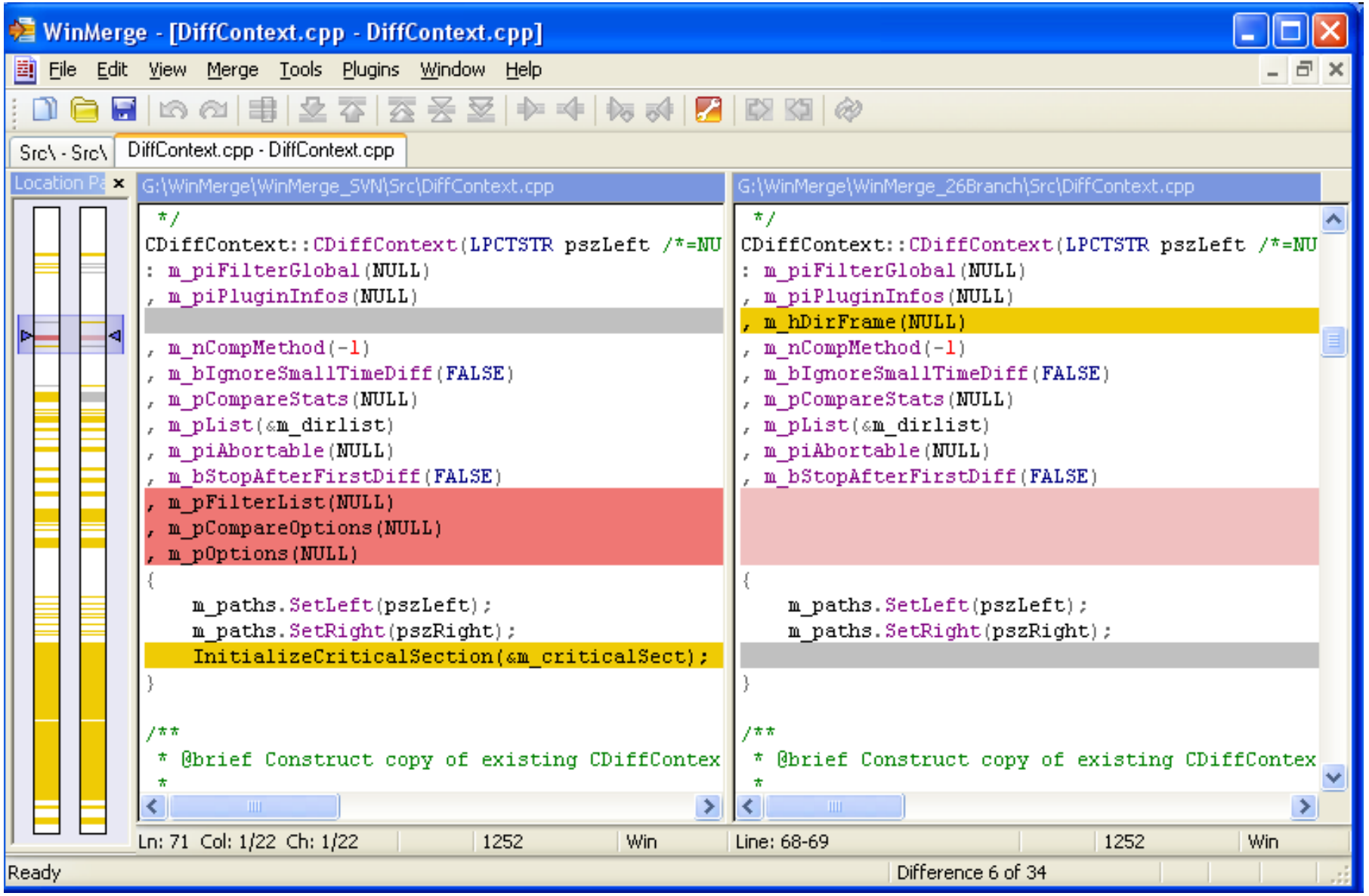
- Quand on modifie des sources :
  - Des bugs apparaissent parfois (souvent !)
- On pourrait sauver chaque version de chaque fichier modifié…
  - Ou ne stocker que les différences !
- Et quand on est plusieurs à modifier
  - ☞ Savoir qui modifie quoi
  - ☞ Ne rien écraser
  - ☞ Fusionner si on modifie à plusieurs
  - ☞ Revenir en arrière (un bug…)
  - ☞ Gérer des développements en //



Principe de la différenciation

- Outil diff
- Différences entre 2 fichiers d'après
  - Ligne de début/de fin
  - Insertion/Suppression d'une ou plusieurs lignes
- Facilité de détection et de construction d'un patch
- Pas de détection des lignes modifiées
  - Traitées comme suppression + insertion

Différenciation : illustration



Historique

- **SCCS (livré avec Unix dès Vx, Bell labs programmer workbench, fusionné en 1983)**
- **RCS (W. Tichy 1985)**
- **CVS (B. Berliner 1989)**
  - Support dans beaucoup d'environnements…
- **Subversion (subversion.tigris.org)**
  - Bonne gestion des modifications de l'arborescence des répertoires
  - Installation et maintenance simplifiée
- **Visual Source Safe :** The Microsoft Way
- **ClearCase (Rational) :** L'usine de gestion de traçabilité
- **Git, Mercurial…**

Ce que SVN n'est pas...

- **Un système de construction (makefile, ant…)**
- **Un système de gestion de projet (Ms-project)**
- **Un substitut à la communication entre développeurs (ex: conflit *sémantique)***
- **Un système de** contrôle **du changement (bug-tracking, ChangeLog)**
- **Un système de tests automatisés**
- **Un système fondé sur un processus particulier**

Commande(s) SVN

- `svn subcommand [switches] [cmd_args]`
  - Commande de base coté client
  - **subcommand : obligatoire**
  - **switches : options spécifiques à la sous-commande**
  - **cmd_args : arguments de la sous-commande**

```
svn checkout http://svn.c.net/rep/svn/trunk subv
```

- `svnadmin subcommand [switches] [cmd_args]`
  - Administration de la base

Base (ou dépôt) svn

- **Locale (accéder directement par le client) :**
  - `file://`

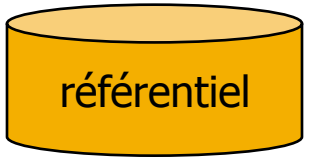
- **Accédée à travers Apache 2 (WebDAV)**
  - `http://`
  - `https://` (SSL encryption)

- **Accédée par le protocole spécifique « svn » (possibilité de passer par ssh)**
  - `svn://` (nécessitée d'avoir un serveur svnserve)
  - `svn+ssh://` identique à svn://, mais *tunneling ssh* (et pas de serveur)

Créer une base SVN

- `svnadmin create /chemin/vers/referentiel`
- **Par défaut format de stockage FSFS (autre format Berkeley-DB moins performant, conservé pour compatibilité)**

administrateur

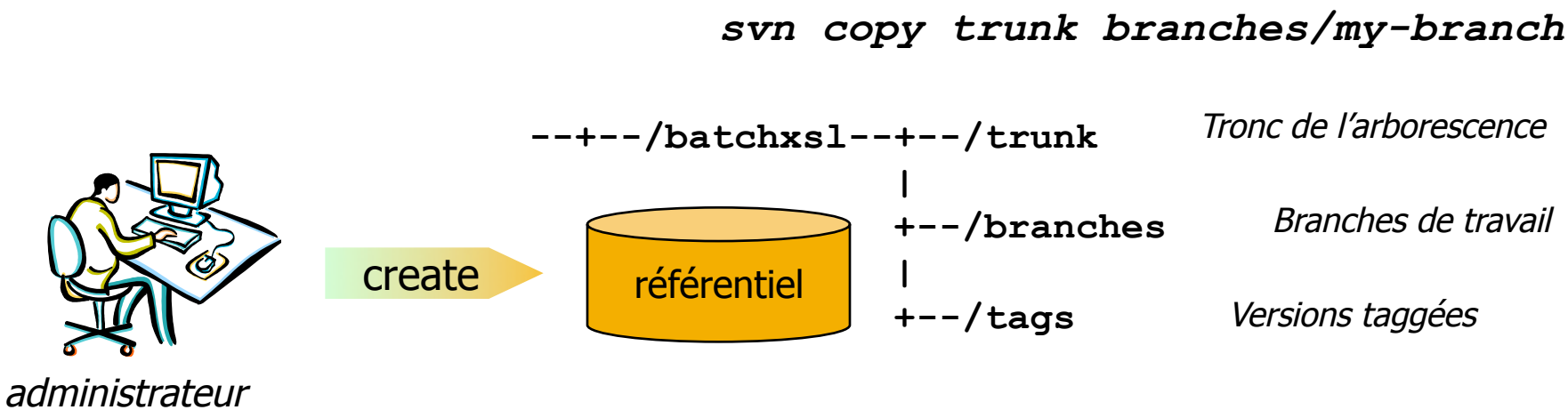


- conf : répertoire des fichiers de config
- dav : répertoire spécifique à mod_dav_svn
- db : les données (pas directement « lisibles »)
- format : un fichier avec un seul entier donnant le numéro de version des hooks de traitement
- hooks : répertoire des scripts de hook
- locks : répertoire des verrous de subversion
- README.txt : des infos sur les autres répertoires

Administrer une base SVN

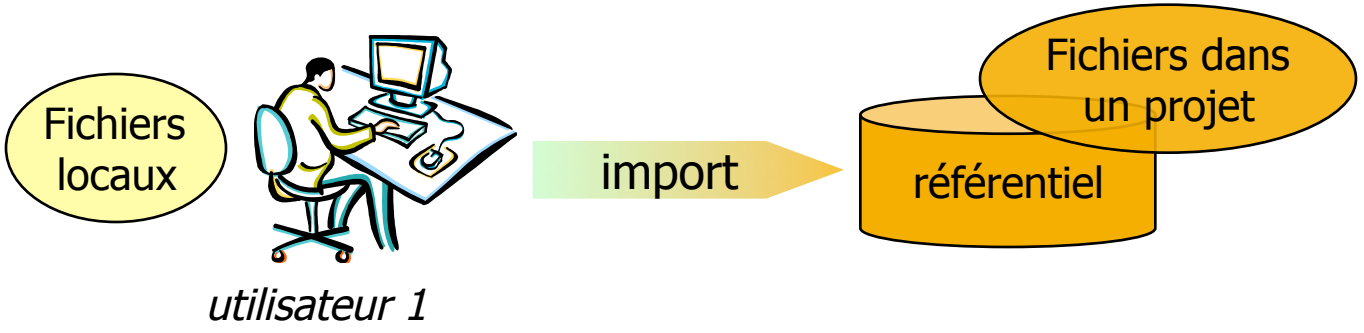
- Au sein d'une base se trouvent un ou plusieurs projets.

- À chaque projet correspond en général un répertoire situé à la racine du dépôt et qui contient lui-même les fichiers et dossiers du projet.
  - Organiser les répertoires :



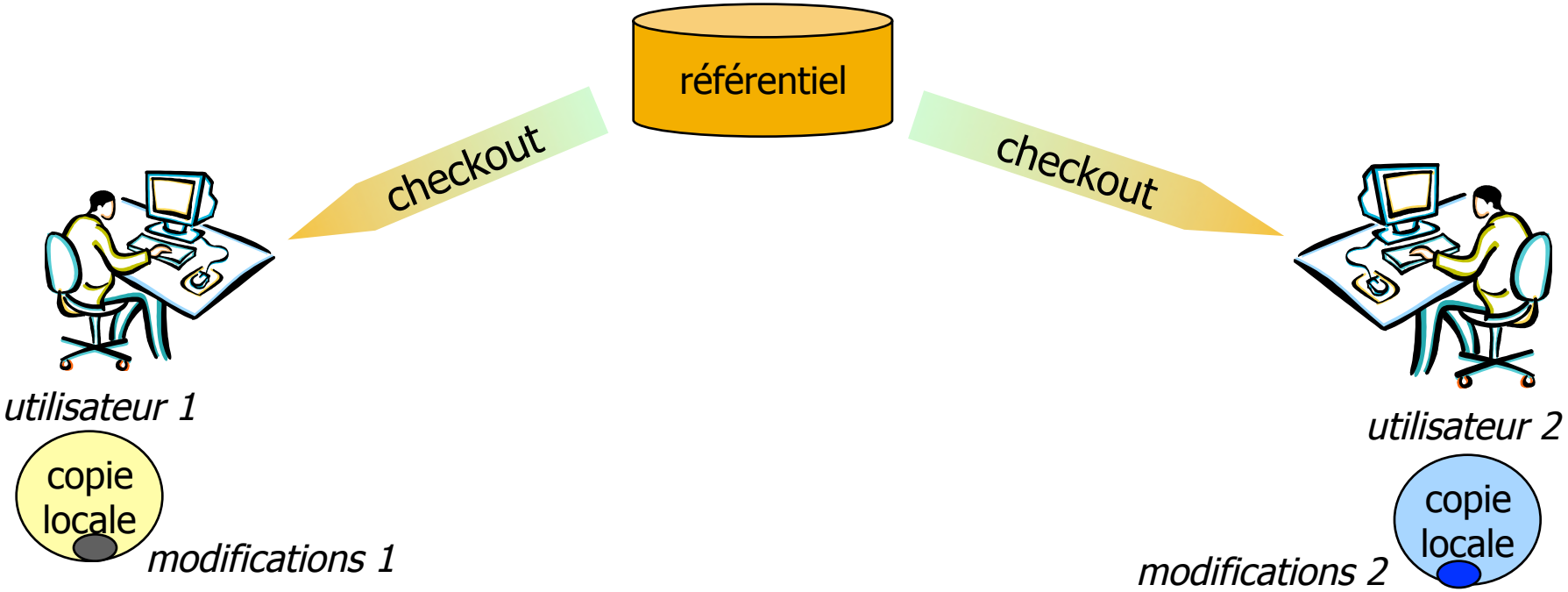
Importer des sources

- **Importer des sources**
  - `svn import rep_local /chemin/vers/referentiel [options]`
  - `svn import myTree file:///usr/local/svn/newrepos/batchxsl/trunk -m "Initial import"`



Récupérer une copie locale des sources

- `svn checkout chemin/vers/referentiel/et/projet [options]`
- `svn checkout http://svn.collab.net/repos/svn/trunk`



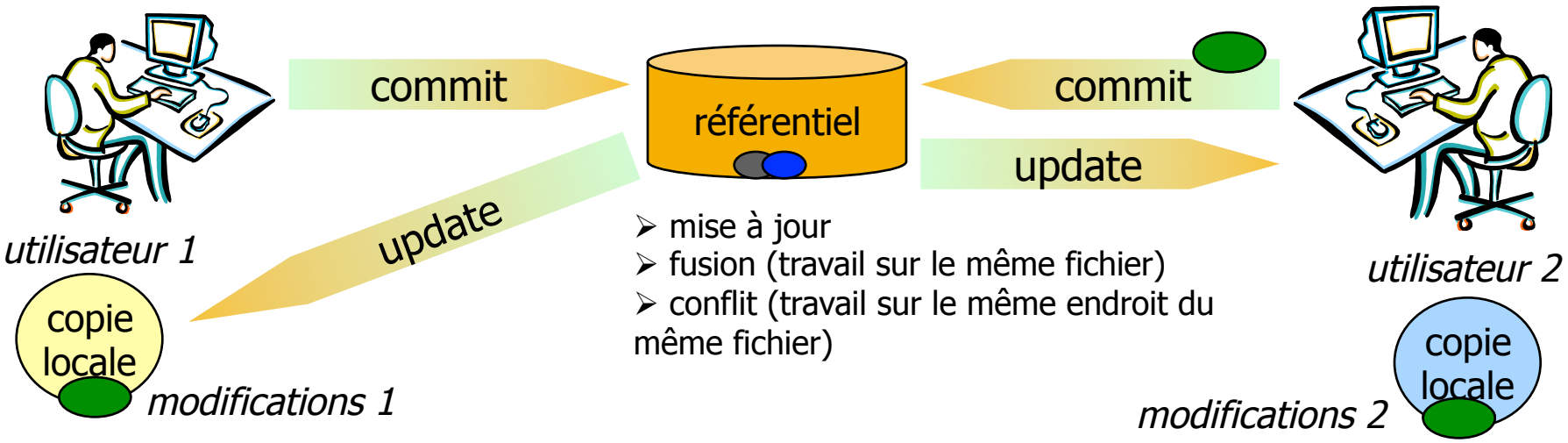
Propager ses changements
Mettre à jour par rapport à la base

❒ Propagation de vos changements

- `svn commit`

❒ Récupération de nouvelles mises à jour

- `svn update`



Signification des sorties SVN pour *update* et *checkout*

- `U file` : **votre répertoire a été mis à jour**
- `A file` : **fichier ajouté à votre copie privée, sera propagé après commit**
- `D file` : **fichier effacé… définitivement après commit**
- `C file` : **conflit détecté lors de fusion**
- `G file` : **fusion effectuée (car pas de conflit)**

Quelques commandes et options

- **Ajouter un fichier/répertoire :** `svn add`
  - **+ commit**
- **Retirer un fichier/répertoire :** `svn delete`
  - **+ commit**
- **Copie des fichiers/répertoires :** `svn copy`
  - **+ commit**
- **Déplacer des fichiers/répertoires :** `svn move`
  - **+ commit**

Quelques commandes et options (suite)

- **Liste des répertoires dans le référentiel :**
  - `svn list`
- **Affichage des messages de commit :**
  - `svn log`
- **Mes modifications locales (pas de connection au référentiel) :**
  - `svn status`
- **Visualiser les différences :**
  - `svn diff`
- **Revenir en arrière (undo) :**
  - `svn revert`
- **Indiquer qu'un conflit est résolu sur un fichier :**
  - `svn resolved sandwich.txt`

Illustration : TortoiseSVN

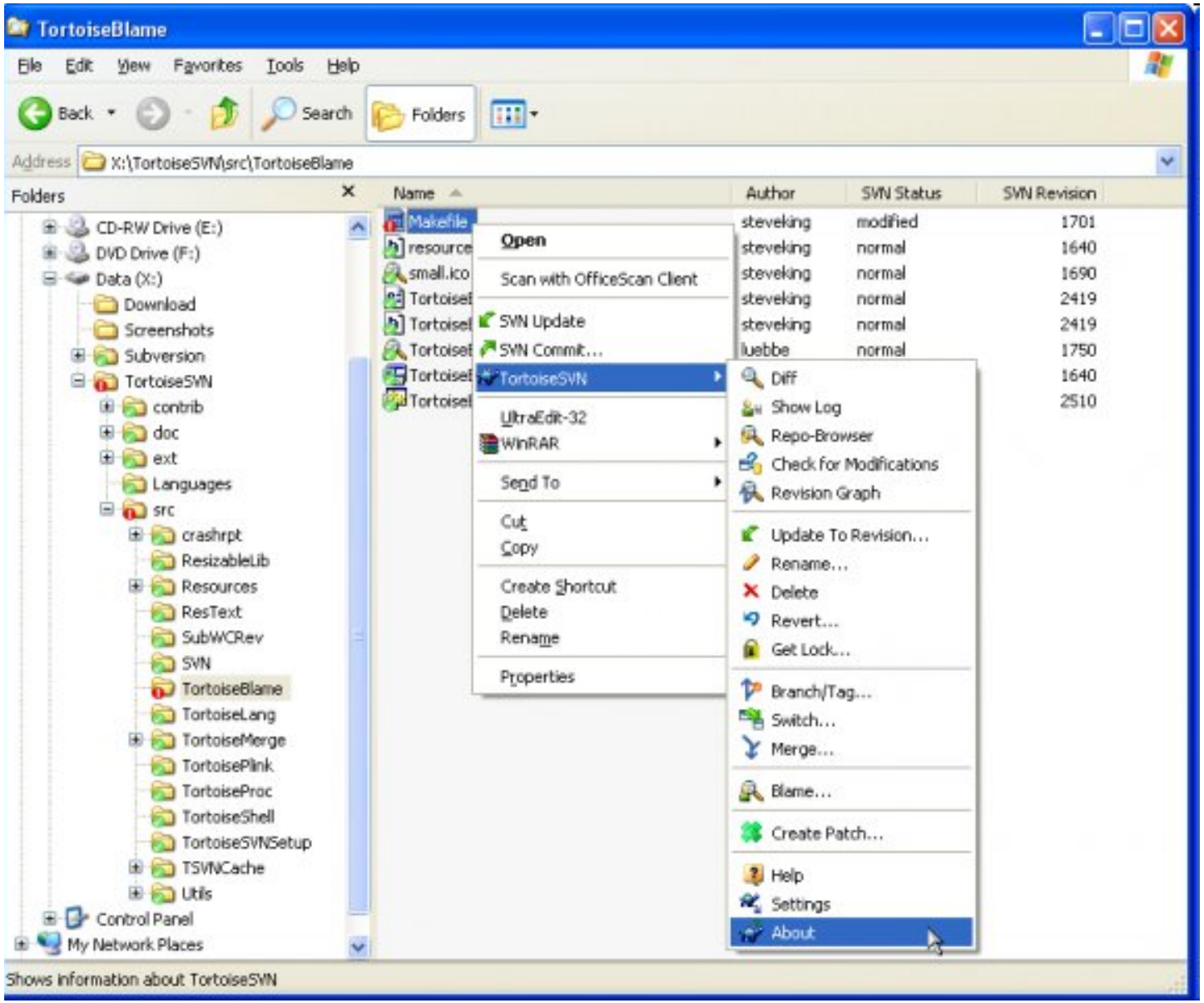
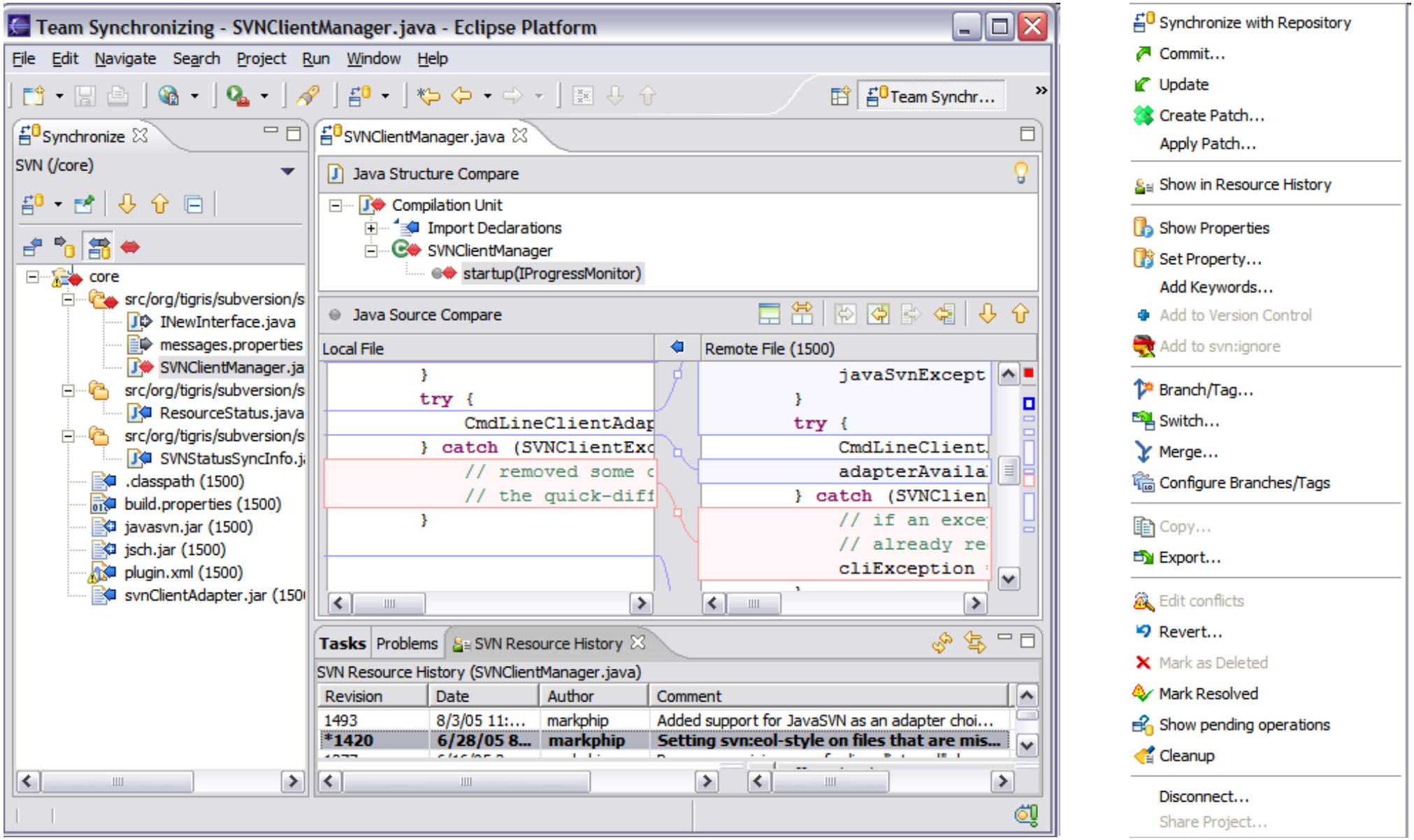


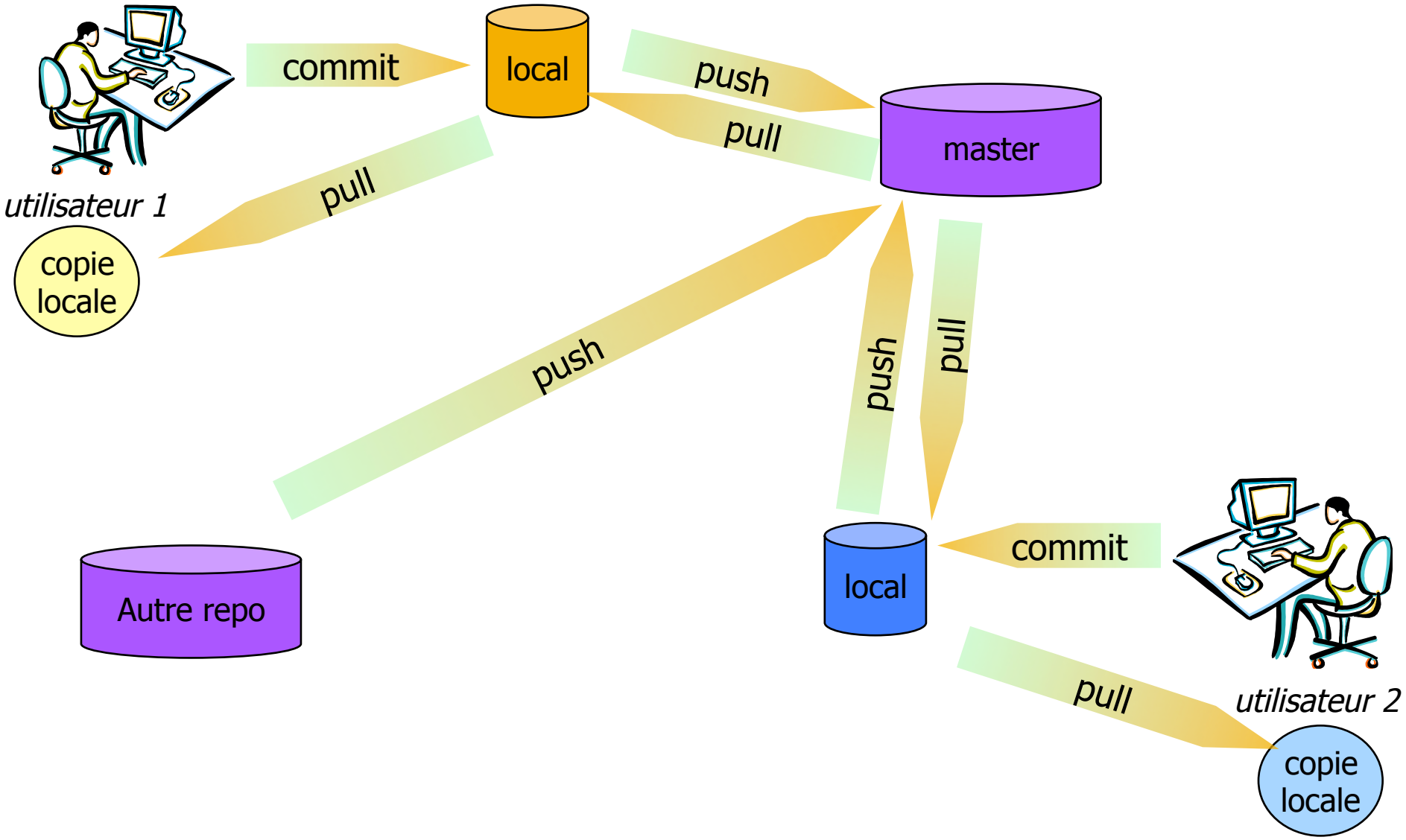
Illustration : plugin subclipse dans Eclipse



Centralisation vs. Distribution

- SVN: pas d'accès à une base partagée sans connexion réseau
- GIT (et Mercurial, Bazaar):
  - « Distributed » Revision control
    - Plus de serveur central (si le serveur tombe, le service est indisponible)
    - Chacun à une base locale, il existe une ou plusieurs bases distribuées sur des serveurs (une base est la base maître)
  - On peut donc :
    - commiter en local, sans que les autres développeurs soient tenus informés
    - puis « pousser » quand le moment est opportun vers la base maître
    - Updater depuis la base maître
  - Avantages / inconvénients :
    - Si la base maître tombe, on clone sa base locale sur une autre base
    - Complexe à utiliser : parfois on oublie de « pousser » vers la base…

Git : illustration



Git

- http://git-scm.com/
- **Gestion de version distribuée open-source**
  - gestion des branches et des merges
- **Chaque copie de repository distribué est autonome**
  - gère son propre historique et ses versions
  - peut-être utilisé hors connexion indépendamment du repository original
  - Les sources sont publiables d'un repository vers un autre (push)
  - Le partage et la réintégration des sources (pull) est très simple

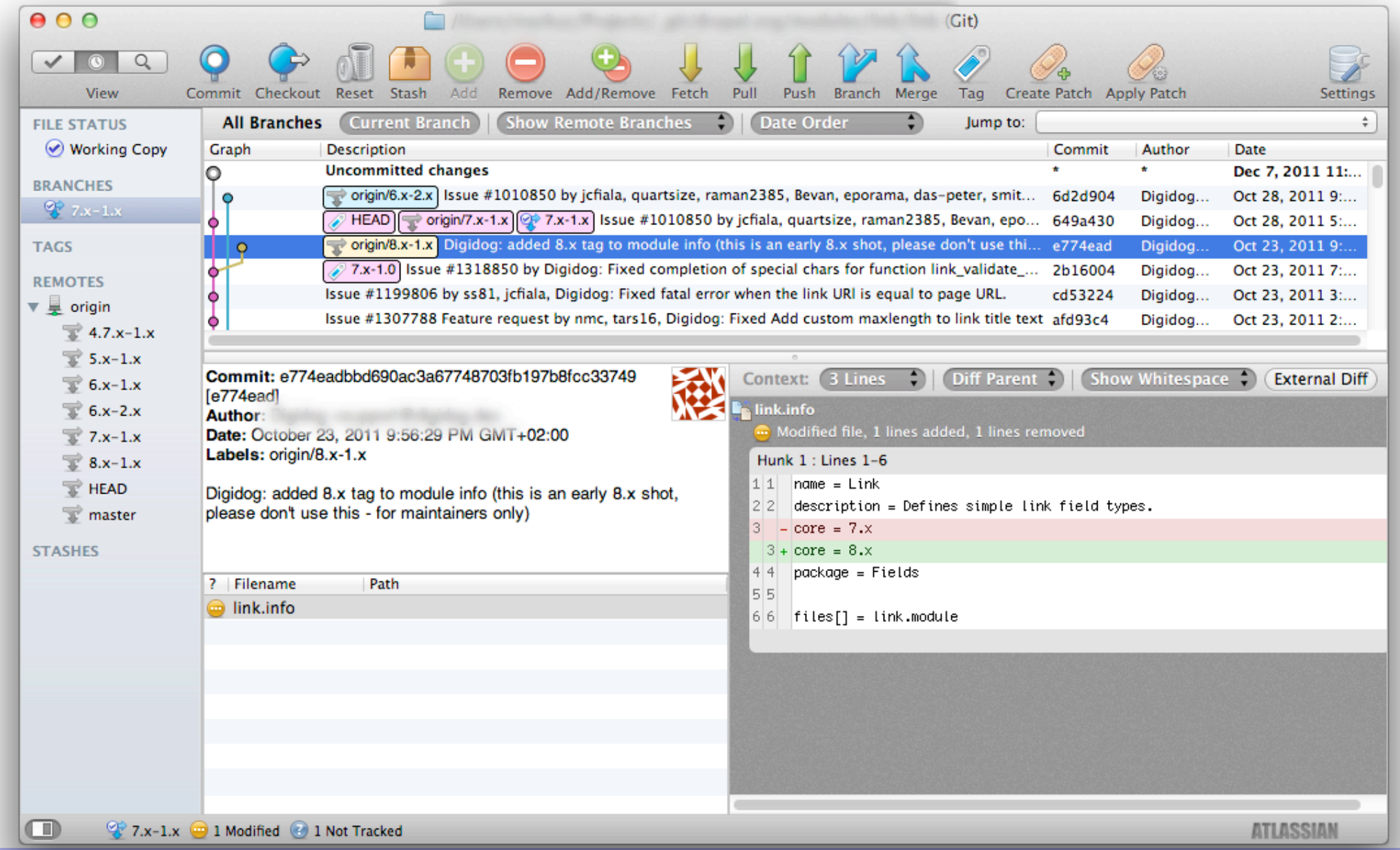
Principales commandes Git

- *git init* : crée un nouveau dépôt
- *git clone* : clone un dépôt distant
- *git add* : ajoute les nouveaux objets depuis le dernier commit. Les objets précédents restent inchangés
- *git commit* : marquer les changements en local
- *git branch* : crée une nouvelle branche de développement
- *git merge* : fusionne plusieurs branches de développement
- *git push* : publie le commit vers un repository distant
- *git pull* : récupère depuis un repository distant

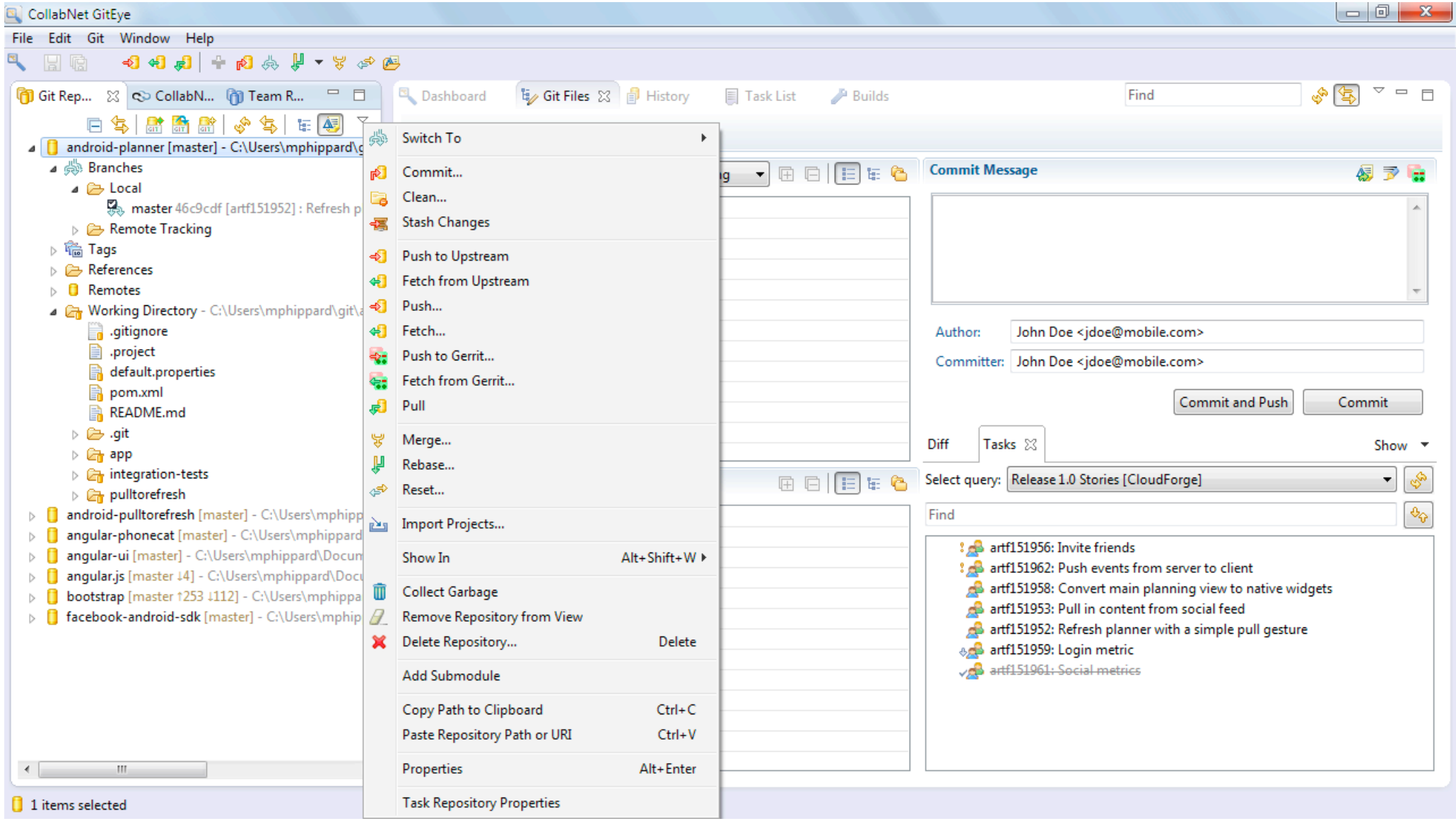
Tags en git

- Pour tagger la tête de votre repository local avec un tag (nommé ESSAI ici) :
  - git tag ESSAI
  - git push origin ESSAI
- Vous pouvez aussi exécuter "git push origin --tags" pour pusher tous les tags existants à la fois.
- Si vous souhaitez déplacer le tag FINAL, vous pouvez d'abord l'effacer :
  - git tag -d ESSAI
  - git push origin :refs/tags/ESSAI
- Puis retagger comme vu précédemment.
- Pour information, je clonerai votre projet par une commande "git clone" classique, puis je récupérerai votre code par :
  - git checkout FINAL
- Si vous n'êtes pas capable d'exécuter la dernière commande sur un autre clone de votre git, c'est que le tag est mal fait, ou pas pushé...

Client Git : SourceTree (Mac, Windows)



Client Git : GitEye (Mac, Windows, Linux)



Attention !

- **Ne pas utiliser EGit sous Eclipse**
  - Effectue des commandes dans votre dos
  - Se plante
  - Met le bazar dans tous les repositories

Références

- Git
  - Pro-Git Book : http://git-scm.com/book/fr
  - http://www.cheat-sheets.org/saved-copy/git-cheat-sheet.pdf
  - http://www.git-tower.com/files/cheatsheet/Git_Cheat_Sheet_grey.pdf
  - http://git-scm.com/
- SVN
  - The SVN Book: http://svnbook.red-bean.com/
  - http://subversion.tigris.org/