INF2050 – Outils et pratiques de développement logiciel

Gestion de source

Jacques Berger

Objectifs

Introduire la pratique de gestion des sources

Prérequis

Programmation

Gestionnaire de sources

Logiciel qui gère le code source

- Modifications
- Versions
- Modifications concurrentes
- Imputabilité des changements
- Historique des modifications

Utilité

Le développement sans gestionnaire de sources :

- Copies manuelles (backup et versions)
- Impossible de retracer les changements
- Écraser les modifications des autres
- Historique manuel dans le fichier

Fonctionnalités

Historique des modifications

Mises à jour des sources (commit / update)

Retour en arrière (fichier, version, modification)

Consulter les anciennes versions d'un fichier

Fonctionnalités

Branches

Appliquer une modification dans plusieurs branches

Un dépôt centralisé

Fonctionnalités

Accessible par ligne de commande

Interface visuelle (GUI)

Intégration à l'IDE

diff

Gestionnaires de sources connus

Gratuits :
- CVS
- Subversion (SVN)
- Mercurial
- Git

Payants :
- Team Foundation Server (Microsoft)
- ClearCase

Et plusieurs autres...

Qualités recherchées

Simplicité d'utilisation
- Productivité du développeur
- Documentation pertinente

Flexibilité

Rapidité

Inclure

Ce qu'on inclut dans le gestionnaire de sources :

- Code source (évidemment)
- Prototypes jetables
- Tout ce qu'on tappe
  - Tests
  - Documentation
  - Etc.

Inclure

Format texte autant que possible

- Les formats binaires ne passent pas l'épreuve du temps
- Les gestionnaires de source sont faits pour manipuler du texte

Exclure

Le code généré
- Par un générateur de code
- Par l'IDE

Les librairies

Documents d'analyse

Diagrammes UML

Centralisé vs décentralisé

2 types de gestionnaires de sources :
- Centralisé
- Décentralisé

SVN est centralisé
Git est décentralisé

Vocabulaire git

Working tree : Le répertoire où vous faites vos modifications dans le code source

Index : L‘espace où l‘on place les fichiers du prochain commit

Local repository : L‘historique de tous les changements du projet, placé sur votre poste

Commandes git

Configurations de départ

```
git config --global user.name "Votre nom"
git config --global user.email courriel@courriel.org
```

Commandes Git

`git init`

Création d'un dépôt local pour un nouveau projet

`git clone <url>`

Création d'un dépôt local à partir d'un projet déjà existant

Commandes git

`git add <fichier>`

Ajoute un fichier modifié à l‘index

`git checkout <fichier>`

Le fichier du Working Tree sera écrasé par celui placé dans l‘index

Commandes git

`git rm <fichier>`

Supprime un fichier et place la suppression dans l‘index

`git mv <fichier>`

Déplace un fichier et le déplacement est mis dans l‘index

Commandes git

`git status`

Indique l‘état du dépôt, incluant les fichiers modifiés et le contenu de l‘index

`git diff <fichier>`

Présente les modifications faites dans un fichier

Commandes git

`git commit`

Ajoute les modifications dans l‘index au Local Repository

`git reset <fichier>`

Retire les modifications d‘un fichier de l‘index

Commandes git

`git commit -a`

Toutes les modifications seront mises dans l‘index

`git commit -m <message>`

Effectue un commit en spécifiant une description pour le commit

Commandes git

`git reset –hard HEAD`

Replace l‘index et le Working Tree comme dans le commit HEAD

`git clean -df`

Efface les fichiers et répertoires qui n‘existent pas dans l‘index

Commandes git

`git log`

Affiche l‘historique des commits

`git blame <fichier>`

Affiche l‘information sur la dernière modification de chaque ligne du fichier

Commandes git

git revert <commit>
Crée un commit pour annuler l‘effet d‘un autre commit

Commandes git

`git clone <url>`

Télécharge un dépôt distant et crée un dépôt local identique au dépôt distant

Commandes git

`git pull`

Met à jour le dépôt local en fonction du dépôt distant; des conflits peuvent survenir lors de cette opération

`git push`

Envoyer les commits du dépôt local au dépôt distant

.gitignore

On place un fichier nommé .gitignore dans le répertoire principal du dépôt

Il contient des expressions servant à exclure certains fichiers du dépôt

Il devient alors impossible de les inclure dans l‘index

Conflit

Les conflits surviennent quand 2 développeurs ont modifiés la même ligne de code

git placera les deux lignes dans le fichier et vous demandera de choisir celle qu‘il soit conserver

Conflit

Le texte en conflit est identifiable avec les caractères :

<<<<<<<<<<<

>>>>>>>>>>>

==========

Il faut résoudre le conflit avec un nouveau commit

Conclusion

Outil indispensable pour un développeur
- Seul ou en équipe
- Peu importe la taille de l'équipe
- Peu importe la taille du projet

Plus loin...

git
    https://git-scm.com/

Mercurial
    https://www.mercurial-scm.org/

Subversion (SVN)
    https://subversion.apache.org/

Plus loin...

Pragmatic Version Control Using Git
Travis Swicegood
The Pragmatic Bookshelf, 2008