



Versioning

INTRODUCTION TO API VERSIONING BEST PRACTICES

Thiery Elliott
Pham Tung
Wendling Nathan

Sommaire

- Introduction
- Historique
- Logiciel de Versioning
- Comparatif



The background features several black hexagons of varying sizes. Some hexagons contain white hand-drawn diagrams. One hexagon shows a box with an arrow pointing down to another box. Another shows a box with an arrow pointing up to another box. A hand is visible, pointing its index finger at a diagram on a large black hexagon. The diagram consists of a box with an arrow pointing right to another box, and an arrow pointing up to that second box. The overall theme is technical or conceptual, likely related to the text on the right.

Introduction

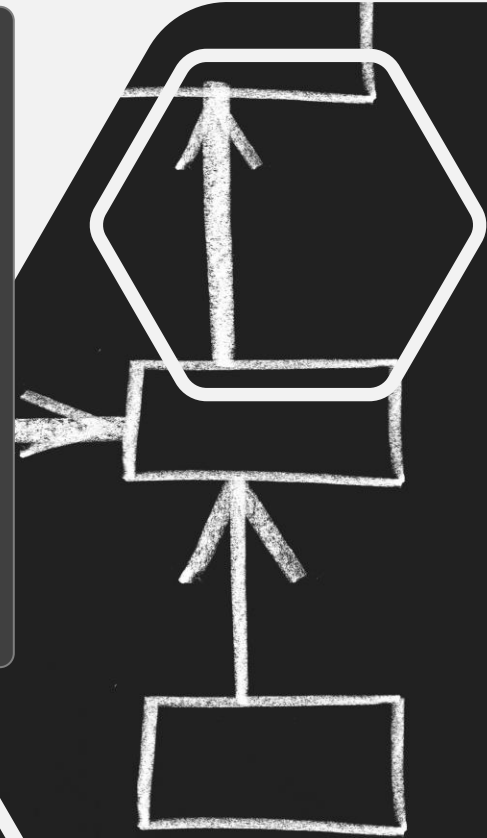
qu'est-ce que le Versioning ?

Versioning

- Cela permet d'établir une traçabilité des différentes versions d'un ou plusieurs fichiers.
- Possibilité d'accéder à une ancienne version d'un fichier en cas de fichier supprimé par exemple.
- Ils permettent de donner la possibilité aux développeurs de pouvoir travailler en équipe sans les contraintes des horaires différent de chaque employé
- améliore la productivité
- Travailler en même temps sans problème.



Historique



Histoire

- 1972 : SCCS (Source Code Control System) par Marc J. Rochkind au Bell Labos où ils ont développé Unix.
- 1982 : RCS (revision control system) 1^{er} Multi platform VCS mais toujours pour que les fichiers textent.
- 1986 : CVS (Concurrent Versions System) le premier qui a un référentiel central et utilisable pour multiple utilisateur.
- 2000 : Subversion
 - TFS par Microsoft
- 2005 : DSCC
 - Git et Github
 - Mercurial



Aujourd'hui, il a existé plusieurs logiciels de gestion libres décentralisés qui ne sont plus tous disponibles tels que

- Darcs (2002),
- DCVS (2003),
- SVK (2003),
- Codeville (2005),
- Mercurial (2005),
- Bazaar (2005),
- Fossil (2007),
- Veracity (2011)
- le premier étant GNU arch sorti en 2001.



Logiciel de
versioning

Logiciel versioning

Different logiciel

- Mercurial SCM
- Git bash
- Apache Subversion



Mercurial SCM

- des performances et une évolution élevées
- la décentralisation
- un développement collaboratif entièrement distribué
- une gestion robuste des fichiers texte et binaires,
- des capacités avancées de branchement et de fusion,
- Et il comprend une interface Web intégrée.
- **tout en restant conceptuellement simples.**



Git bash

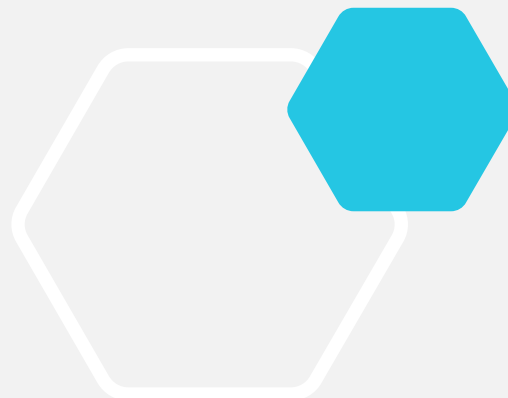
Il est conçu pour coordonner le travail entre les programmeurs, mais il peut être utilisé pour suivre les modifications dans n'importe quel ensemble de fichiers. Ses objectifs incluent la vitesse, l'intégrité des données et la prise en charge de flux de travail non linéaires distribués.



Apache Subversion

Il est distribué sous licence Apache. Il a été conçu pour remplacer CVS. Ses auteurs s'appuient volontairement sur les mêmes concepts (notamment sur le principe du dépôt centralisé et unique) et considèrent que le modèle de CVS est bon, seule son implémentation est perfectible.

CVS: est un système de gestion de versions créé en 1990



Comparatif des logiciels

	Localisation donnée	Stockage	Plateforme	Installation serveur	Sécurité	Droit d'accès
Mercurial SCM	décentralisé	Dépôt de fichiers uniquement	- Windows - Unix - Oracles	Oui (web)	Contenu crypté	Aucune hiérarchie
Git Bash	décentralisé	Métadonnées	- Windows - Unix	Oui possible	Contenu crypté	hiérarchiser
Subversion	Centralisé	Stockage des fichiers uniquement	- Windows - Unix - Oracles	Oui possible	aucune protection	hiérarchiser

Avantages

La gestion des branches. On peut travailler sur plusieurs projets en parallèle sans embêter les autres.

Quand un fichier a été modifié par plusieurs personnes en même temps, Git sait s'adapter et choisir un algorithme qui fusionne intelligemment les lignes du fichier qui ont été modifiées.

La rapidité : lorsqu'il faut mettre à jour, les données sont empaquetées, compressées, et les mises à jour sont fusionnées rapidement.

Défauts

Sa complexité

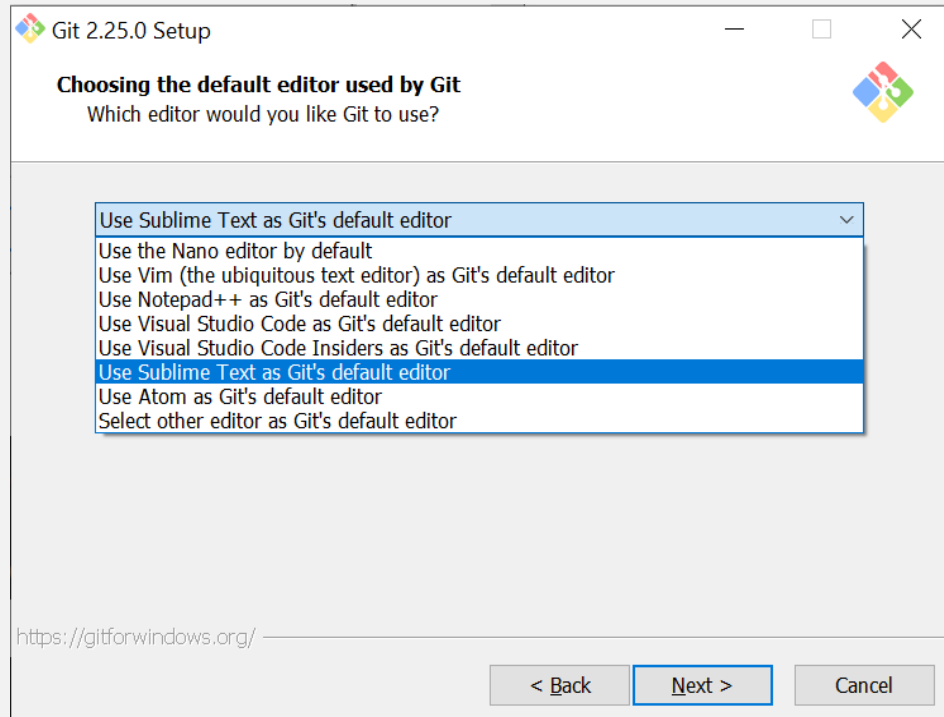
Le très grand nombre de commande



Git bash

Configuration de Git bash

On choisit Sublime Text comme éditeur



Utilisation de GIT Bash

- Identifiant de git

```
ASUS@DESKTOP-BB9CS7B MINGW64 ~  
$ git config --global user.name "PHAM"
```

```
ASUS@DESKTOP-BB9CS7B MINGW64 ~  
$ git config --global user.email "phamsontung.05111995@gmail.com"
```

- Création projet

```
ASUS@DESKTOP-BB9CS7B MINGW64 ~  
$ mkdir nouveau_projet
```

```
ASUS@DESKTOP-BB9CS7B MINGW64 ~  
$ cd nouveau_projet
```

- Créer le répertoire

```
ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet  
$ git init  
Initialized empty Git repository in C:/Users/ASUS/nouveau_projet/.git/
```

- Vérification du statut

```
ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

- Vérifier les fichier à l'intérieure

```
ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    nouveau_projet.sql

nothing added to commit but untracked files present (use "git add" to track)
```

- Ajouter un projet

```
ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (master)  
$ git add nouveau_projet.sql
```

- Vérification du statut une nouvelle fois

```
ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   nouveau_projet.sql
```

- Commettre par 1^e version

```
ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (master)
$ git commit --message "1e version"
[master (root-commit) 6de43d4] 1e version
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 nouveau_projet.sql
```

```
ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (master)
$ git status
On branch master
nothing to commit, working tree clean
```


- Vérification après 1^e modification du fichier

```
ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   nouveau_projet.sql

no changes added to commit (use "git add" and/or "git commit -a")
```

- Vérification des modifications

```
ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (master)
$ git diff
diff --git a/nouveau_projet.sql b/nouveau_projet.sql
index e69de29..5d1a093 100644
--- a/nouveau_projet.sql
+++ b/nouveau_projet.sql
@@ -0,0 +1,3 @@
+DROP database IF EXISTS `projet`;
+Create database projet;
+use projet;
\ No newline at end of file
```

- Application de la nouvelle version

```
ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (master)
$ git add nouveau_projet.sql

ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (master)
$ git commit --message "2e version"
[master 57b8faa] 2e version
1 file changed, 3 insertions(+)
```

- Finalisation

```
ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   nouveau_projet.sql

no changes added to commit (use "git add" and/or "git commit -a")

ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (master)
$ git add nouveau_projet.sql

ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (master)
$ git commit --message "final version"
[master af0ee64] final version
1 file changed, 5 insertions(+), 1 deletion(-)
```

- Affiche les modifications

```
ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (master)
$ git log
commit af0ee6482c6cfb633ba2b4945adfe13fc1ab0234 (HEAD -> master)
Author: PHAM <phamsontung.05111995@gmail.com>
Date: Thu Jan 23 11:15:47 2020 +0100
```

final version

```
commit 57b8faa31fd40825b5ef5b91d02a2f296c56a2c7
Author: PHAM <phamsontung.05111995@gmail.com>
Date: Thu Jan 23 11:11:22 2020 +0100
```

2e version

```
commit 6de43d43fa4445cdac00b204d7e0ee7e82900d3c
Author: PHAM <phamsontung.05111995@gmail.com>
Date: Thu Jan 23 11:06:21 2020 +0100
```

1e version

Activate Windows

- Les branches

```
ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (master)
$ git branch
* master

ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (master)
$ command git checkout -b Elliott
Switched to a new branch 'Elliott'

ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (Elliott)
$ git checkout master
Switched to branch 'master'

ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (master)
$ command git checkout -b Nathan
Switched to a new branch 'Nathan'

ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (Nathan)
$ git branch
Elliott
* Nathan
master

ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (Nathan)
$ git branch -d Elliott
Deleted branch Elliott (was af0ee64).

ASUS@DESKTOP-BB9CS7B MINGW64 ~/nouveau_projet (Nathan)
$ git branch
* Nathan
master
```