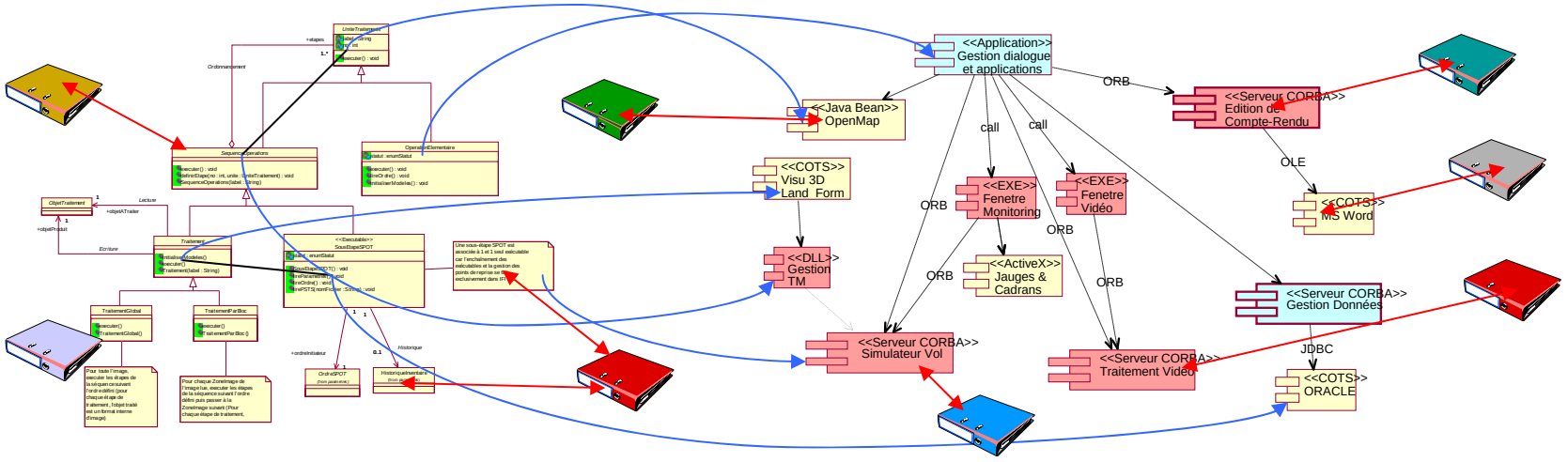
CONDUITE DE PROJET

LA GESTION DE CONFIGURATION DU LOGICIEL

Définition

La gestion des configurations de logiciel est la discipline qui permet de connaître la **composition** d'un logiciel **à tout instant de son développement et de son cycle de maintenance**, d'en archiver les **états successifs** et d'en maintenir la **cohérence** face à l'action désorganisatrice des **modifications**.



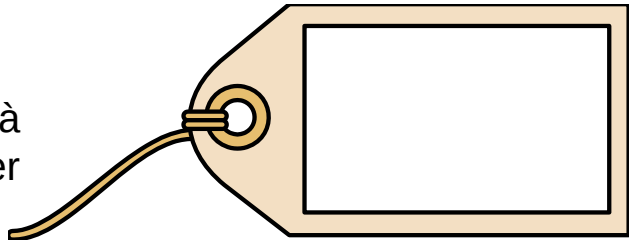
GESTION DES CONFIGURATIONS = GESTION DES VERSIONS
+ GESTION DE LA PRODUCTION
+ GESTION DES MODIFICATIONS

1/ Identification des éléments

- **IDENTIFICATION**:

  activité consistant à:

  - attribuer un identifiant aux articles de configuration et à leurs éléments de configuration associés (leur attribuer un nom et un indice de version),



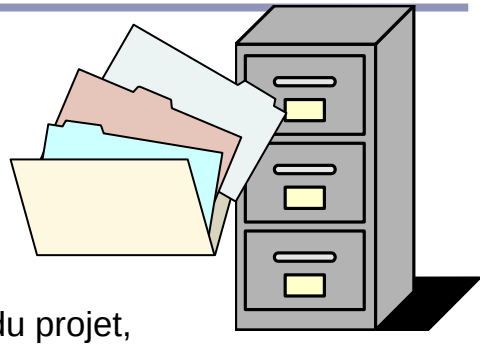
  - marquer les éléments et articles de configuration (y inscrire des informations de manière à pouvoir les reconnaître sans ambiguïté par simple examen visuel: cartouche d'identification des sources, haut et bas de page des documents).

- **ELEMENT DE CONFIGURATION**: constituant élémentaire (indécomposable) d'un logiciel, qui possède un nom et peut évoluer.

  Il s'agit généralement d'un composant logiciel, d'une documentation, de procédures informatiques... C'est la plus petite unité de travail en développement.

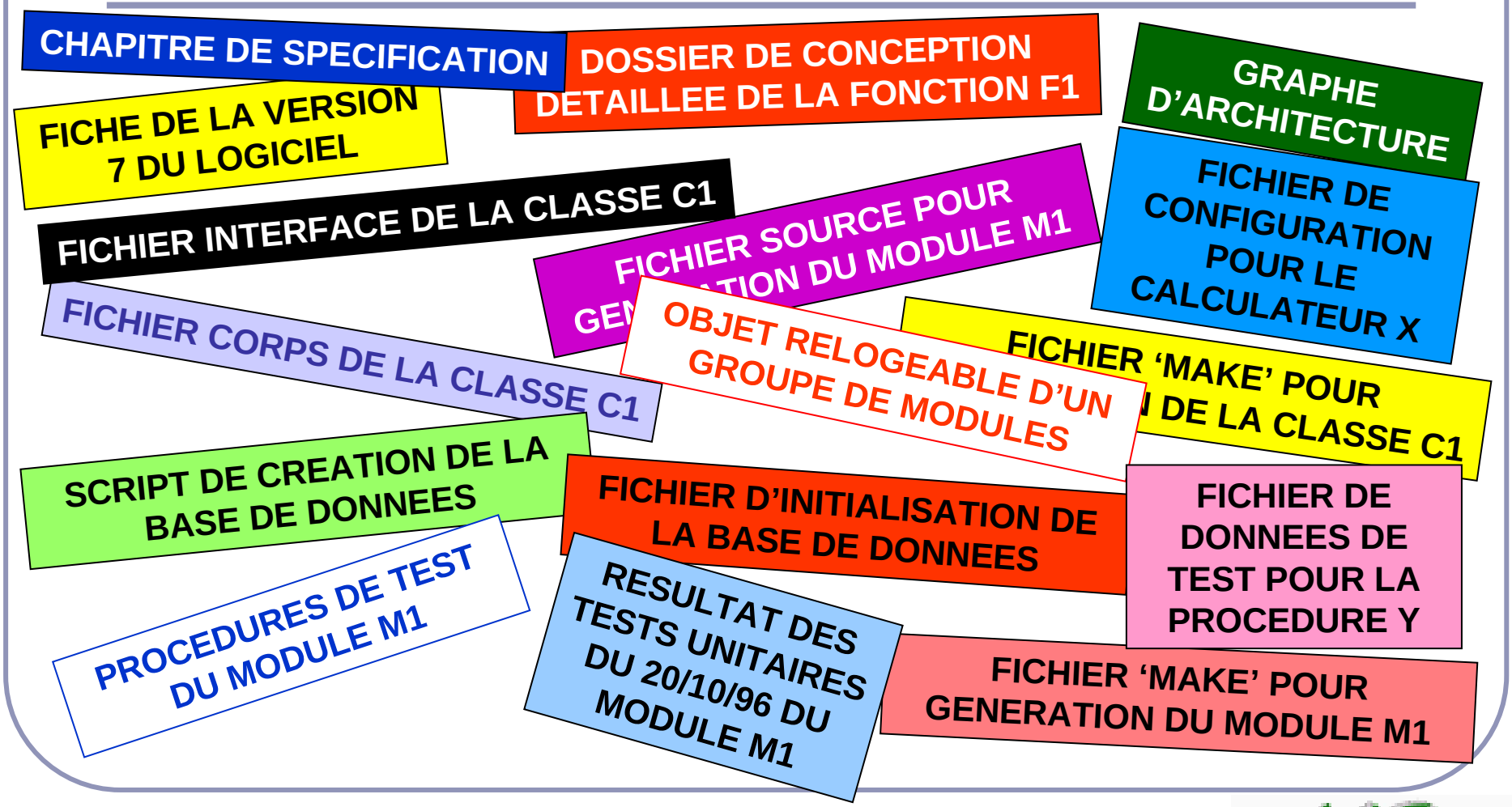
- **ARTICLE DE CONFIGURATION**: ensemble d'éléments de configuration, constituant une unité de gestion de configuration. Un article de configuration peut être éventuellement composé d'autres articles de configuration.

2/ Organisation des articles de configuration



- **CONFIGURATION**: ensemble des articles de configuration composant un Produit Logiciel, établi pour un objectif donné: démonstration, test, livraison.
- **CONFIGURATION DE REFERENCE (ou «référentiel»):** ensemble d'articles de configuration
  - formellement approuvé par un responsable, désigné au démarrage du projet,
  - établi pour un objectif donné,
  - qui sert de base pour les modifications ultérieures (maintenance, garantie.....),
  - ne pouvant être modifiée (seule une nouvelle version des articles peut être créée).
- **ESPACE LOGIQUE**: volume organisé où sont rangés les éléments (espace disque, media ou lieu physique comme un local ou une armoire).Il existe différents types d'espaces:
  - les espaces de travail qui sont sous la responsabilité du développeur et qui contiennent les articles de configuration en cours de réalisation,
  - l'espace de référence de l'application qui contient tous les articles de configuration produits par l'équipe, terminés, validés, prêts pour diffusion,
  - les espaces de test où sont déroulés les tests d'intégration et de validation,
  - etc.

Elements de configuration



Principes d'identification

- Définir les éléments et référentiels
- Décrire chaque élément (classe, rôle, support)
- Décrire les relations entre éléments
- Etablir les règles d'identification des éléments, des configurations, des versions, des modifications, des supports
- Faire appliquer les règles.

Exemple de règle:

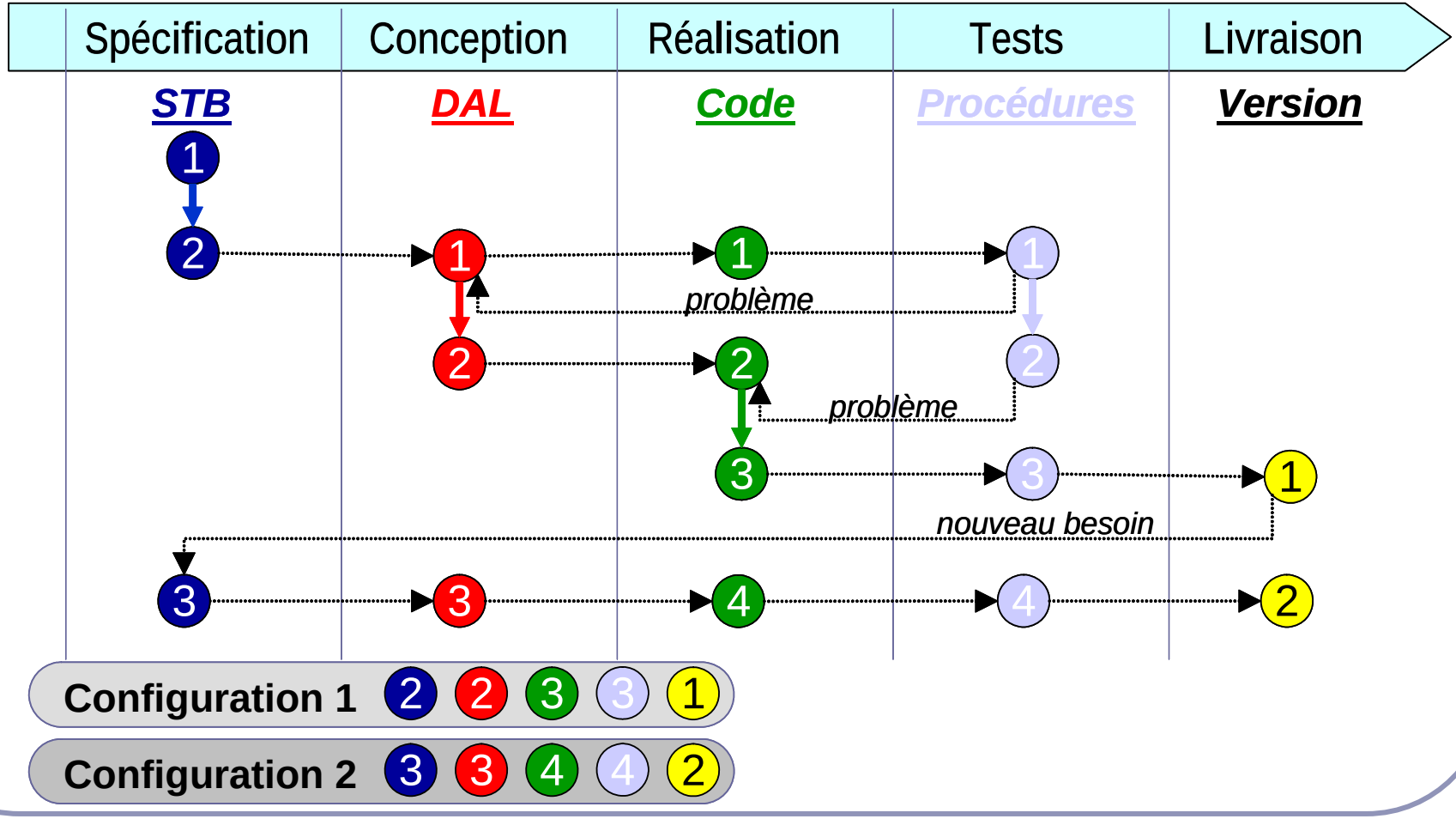
AAA-MMM-XXXXX-YYY-ZZ

avec
- **AAA** → Code projet
- **MMM** → Code application
- **XXXXX** → Référence à l 'élément
- **YYY** → Type élément
- **ZZ** → Numéro de version

Exemples d'identification:

PRJ-EXP-PRINT-SRC-01,
PRJ-ADM-LOGIN-OBJ-02

Petit exemple de gestion des versions **(interdépendance)**



Vocabulaire

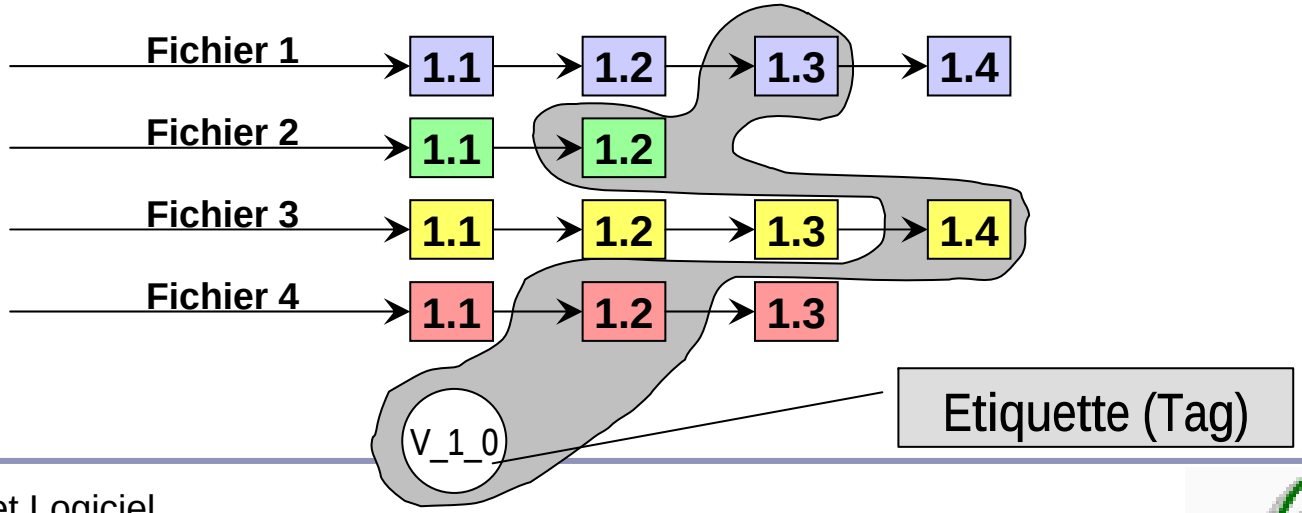
- Une **VERSION** correspond à un état particulier et caractéristique du logiciel (ou de l'un de ses constituants) que l'on veut identifier et gérer dans le temps.
- Chaque fois qu'un composant est modifié, une nouvelle **ITERATION** (ou révision) est créée et identifiée par un numéro attaché au nom du composant.
- Dans le cas d'une **VARIANTE** (ou alternative), l'évolution se fait de façon arborescente plutôt que séquentielle. Des variantes issues d'une même itération peuvent diverger.
- Une **CONFIGURATION** est un ensemble complet et cohérent d'objets en relation les uns avec les autres
  - relation de composition (ex: tous les fichiers sources constituant un module),
  - relation de dérivation (ex: un binaire objet est dérivé du source correspondant)
  - relation d'utilisation (ex: un fichier de données est utilisé par un exécutable)

Vocabulaire

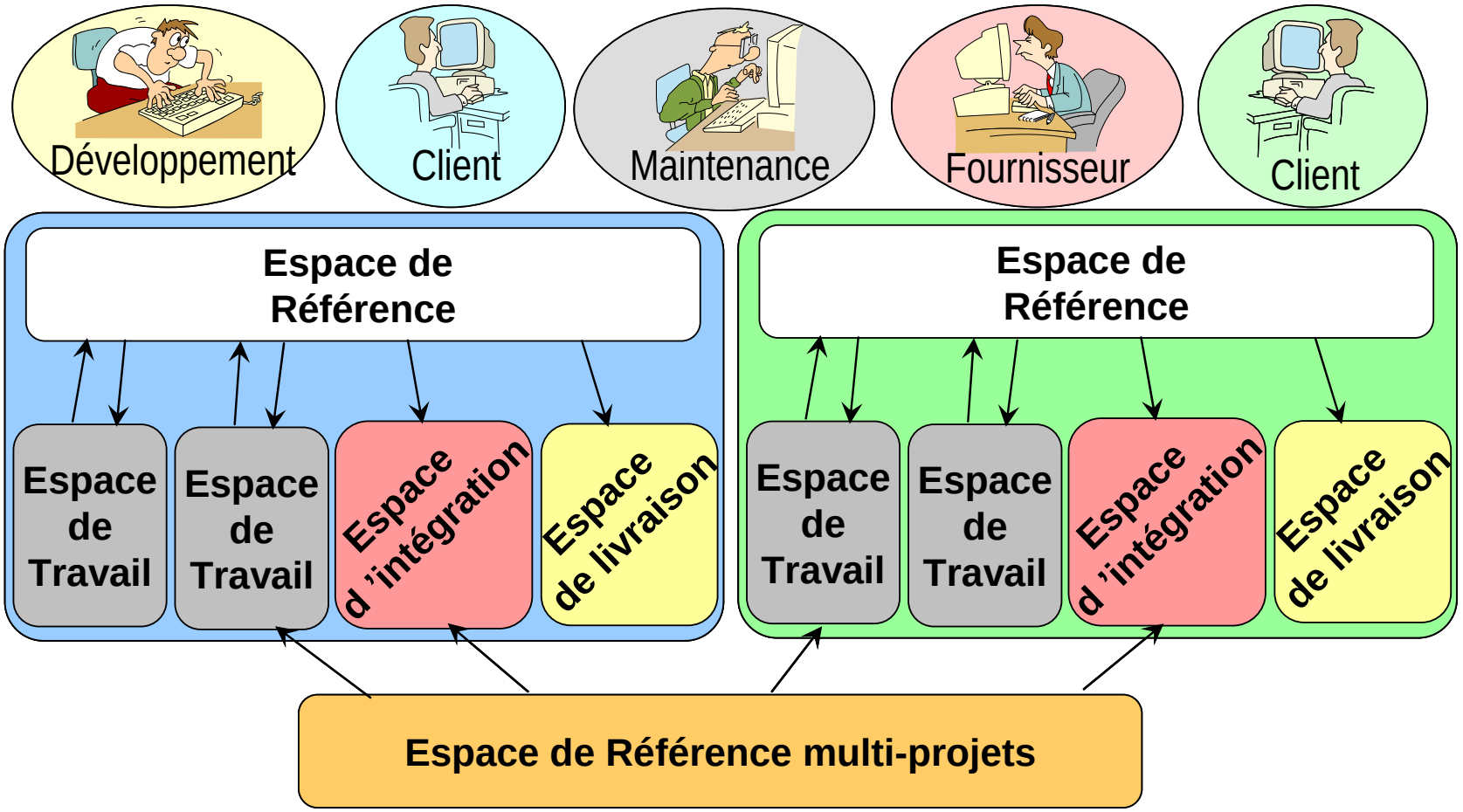
- **révision** : s'applique à un fichier seul

  La modification d'un fichier dans l'espace de référence (commit)provoque la création systématique d'une nouvelle révision et la sauvegarde des révisions précédentes (grâce à une méthode de stockage héritée de RCS (Revision Control System).

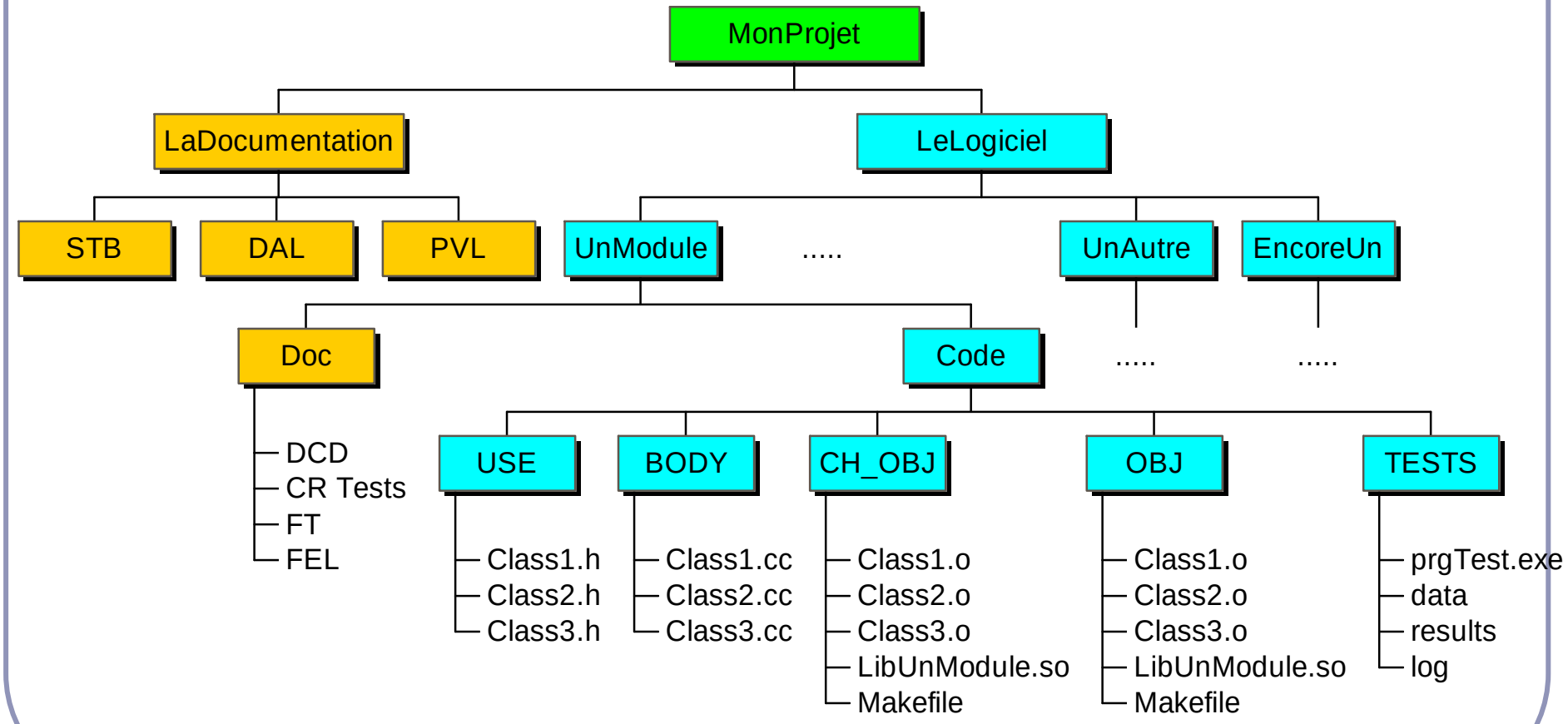
- **version** : plutôt employé pour parler d'une version du projet/module/composant ou d'une version logicielle



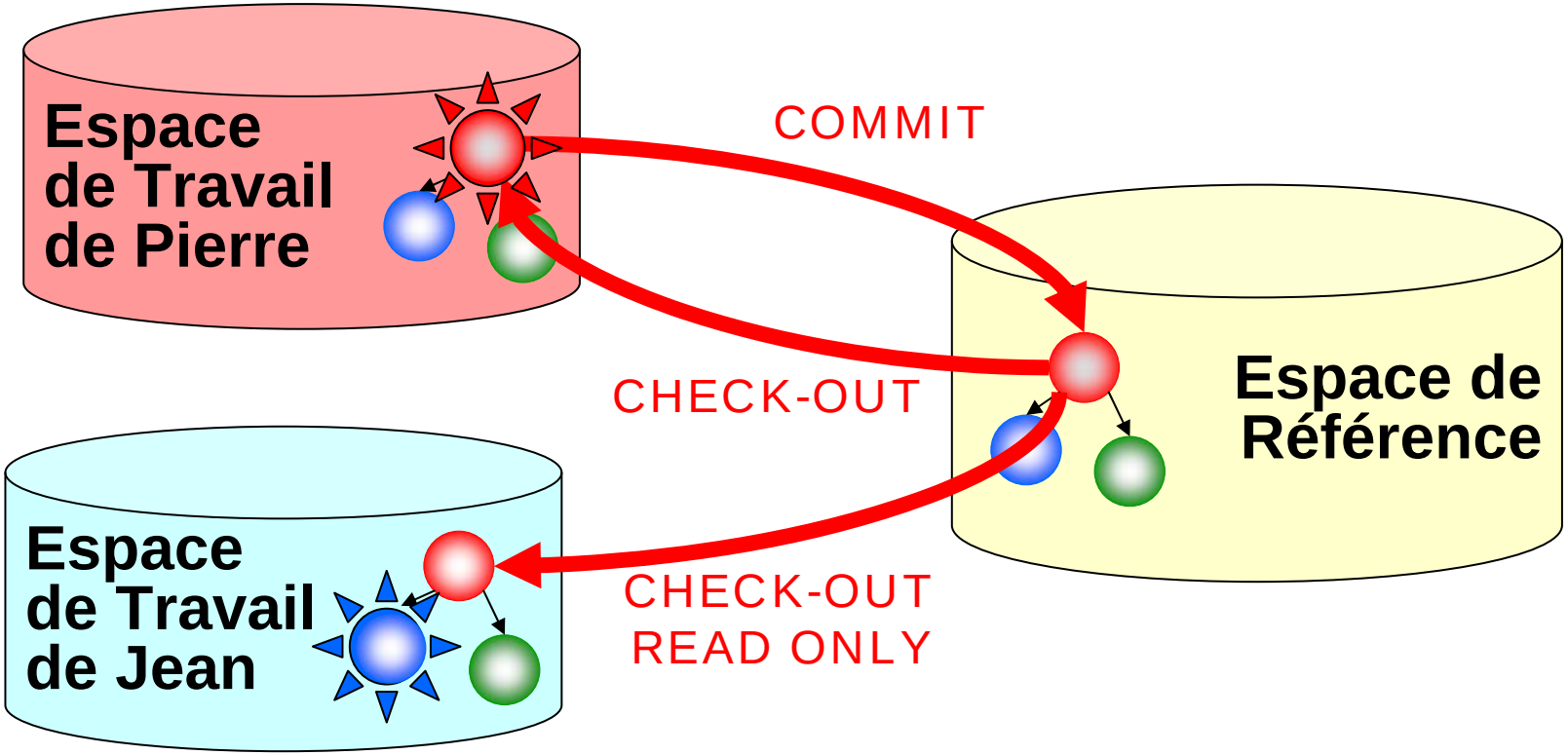
Espaces logiques



Exemple d'organisation des espaces



Partage des éléments



SVN – Subversion

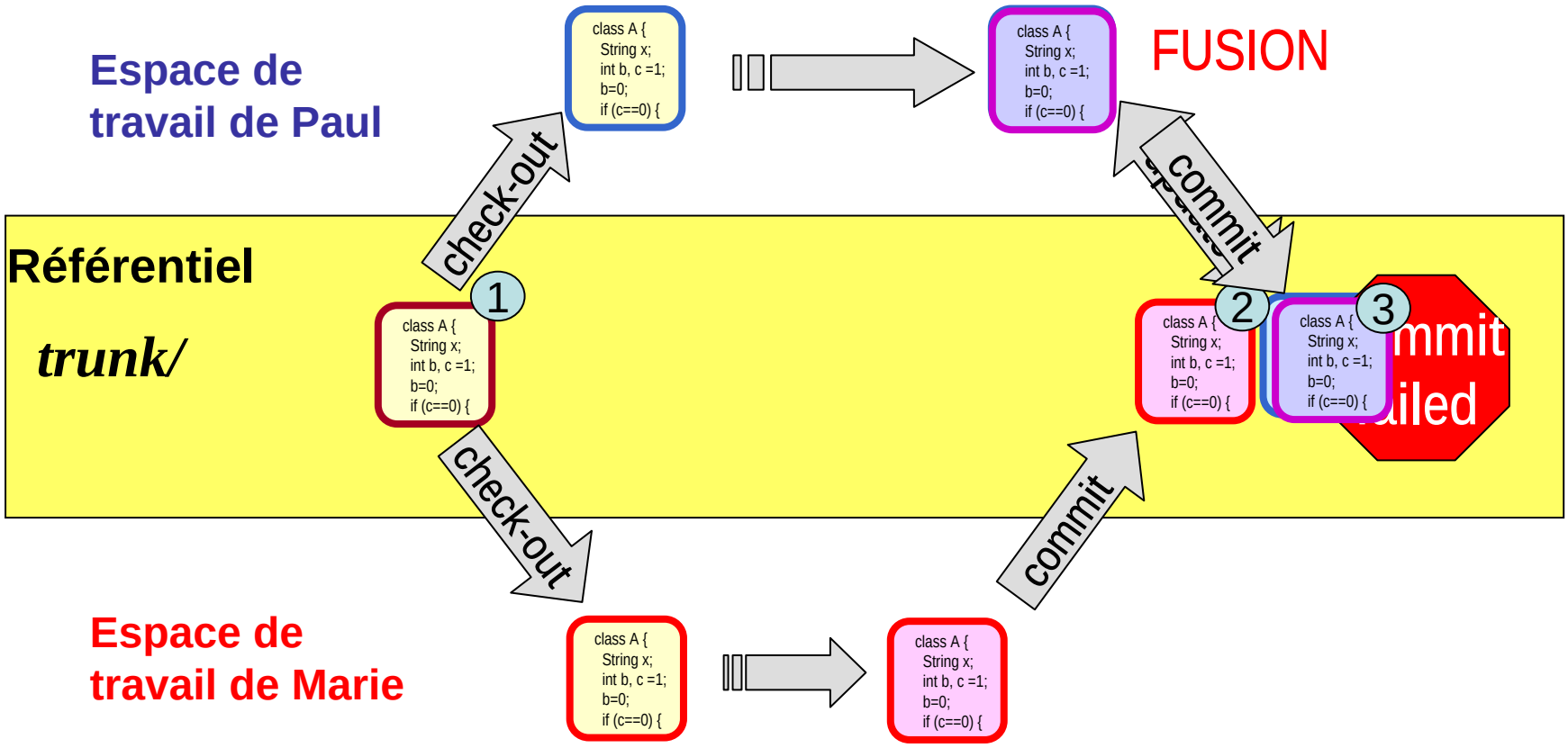
- Destiné à faciliter la gestion de versions et le travail en équipe.
- Permet de gérer l'historique des modifications du projet
- Présenté comme le successeur de CVS
- Lancé en 2000 par CollabNet
- Version actuelle 1.7.7,
- Produit open source sous licence Apache et BSD
- Utilisé en entreprise et par de nombreux projets du libre (Apache, KDE, Gnome, Python)
- Construit autour d'un dépôt centralisé (repository)
- Alternatives : systèmes décentralisés (Git, Mercurial…)

subversion.apache.org

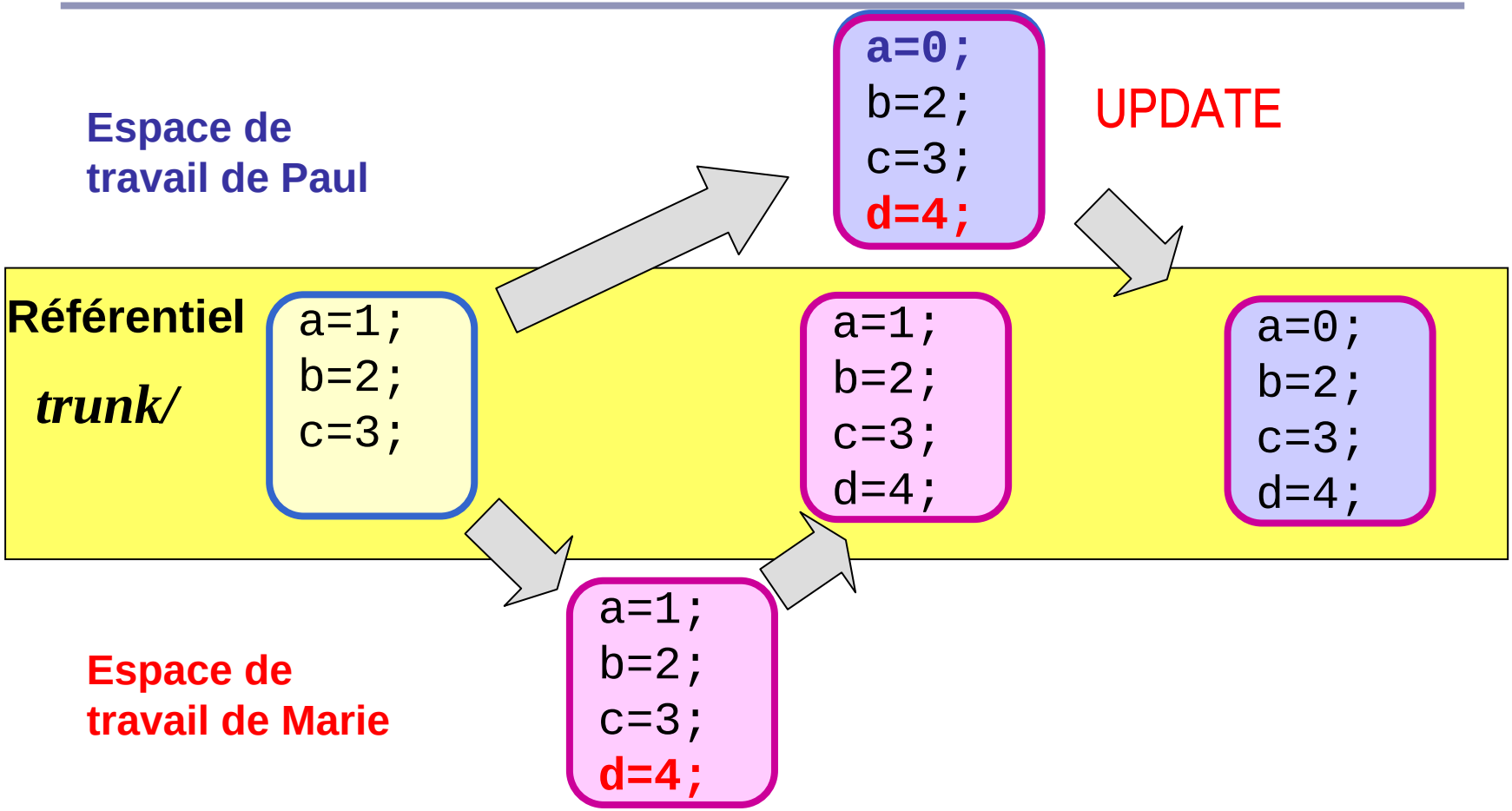
Fonctionnalités de SVN

- **Attribution d'un numéro de révision unique et possibilité de revenir sur une version antérieure à tous moments.**
- **Enregistrement dans l'espace de référence de versions d'articles de configuration et étiquetage (TAG) des éléments qui le constituent.**
- **Gestion du partage entre développeurs.**
- **Gestion de branches.**

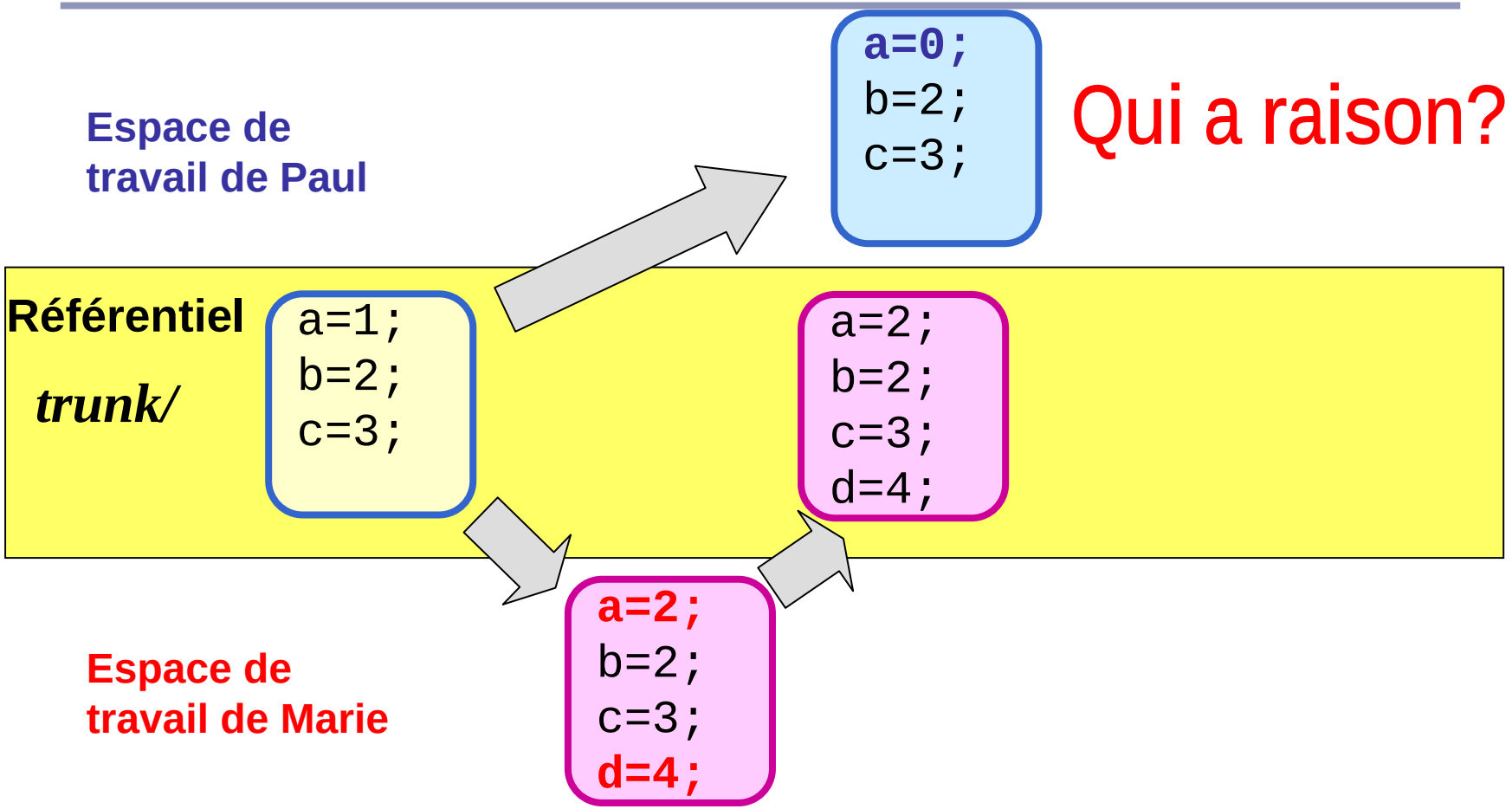
Gestion du partage



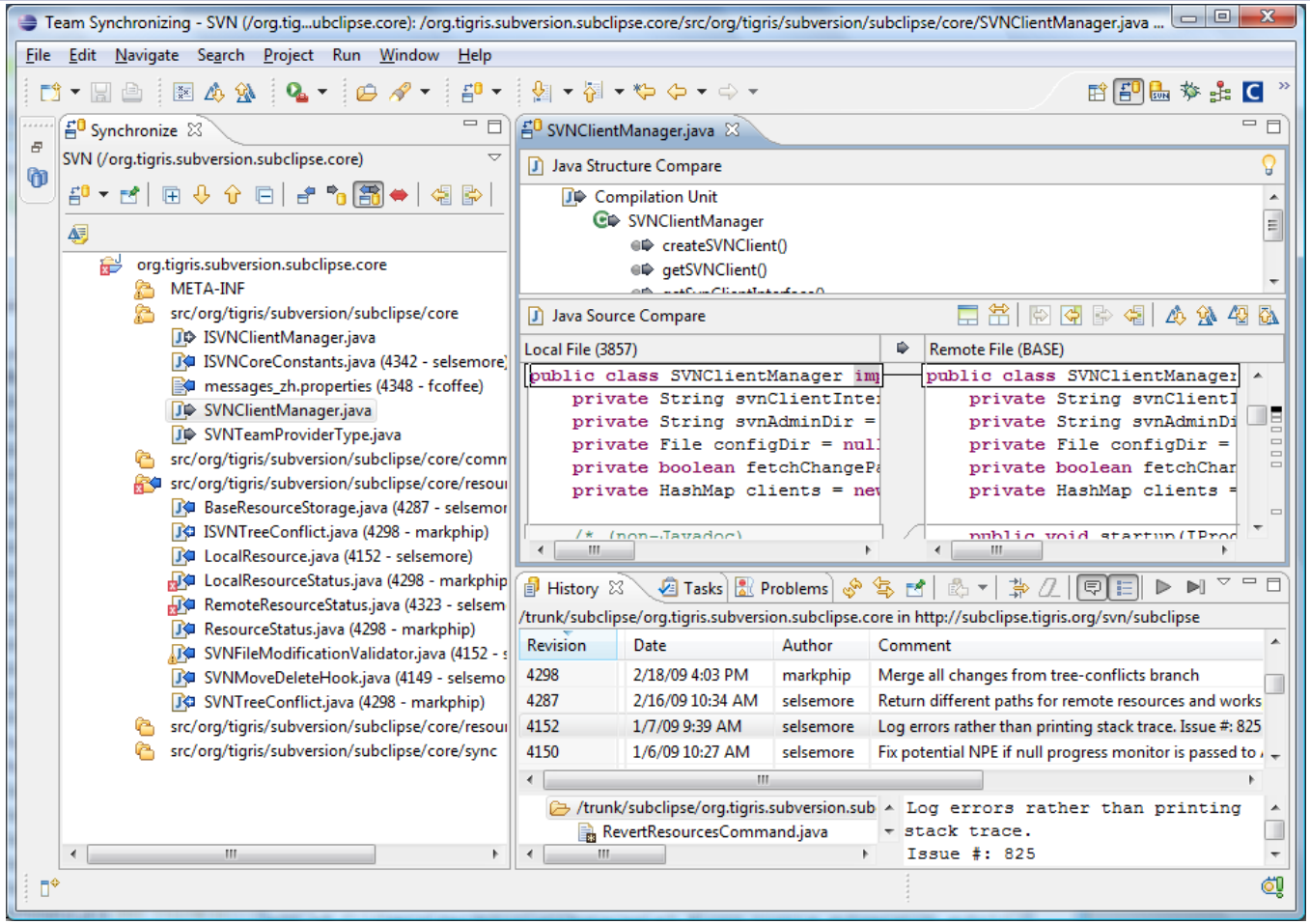
Synchronisation automatique (merge)



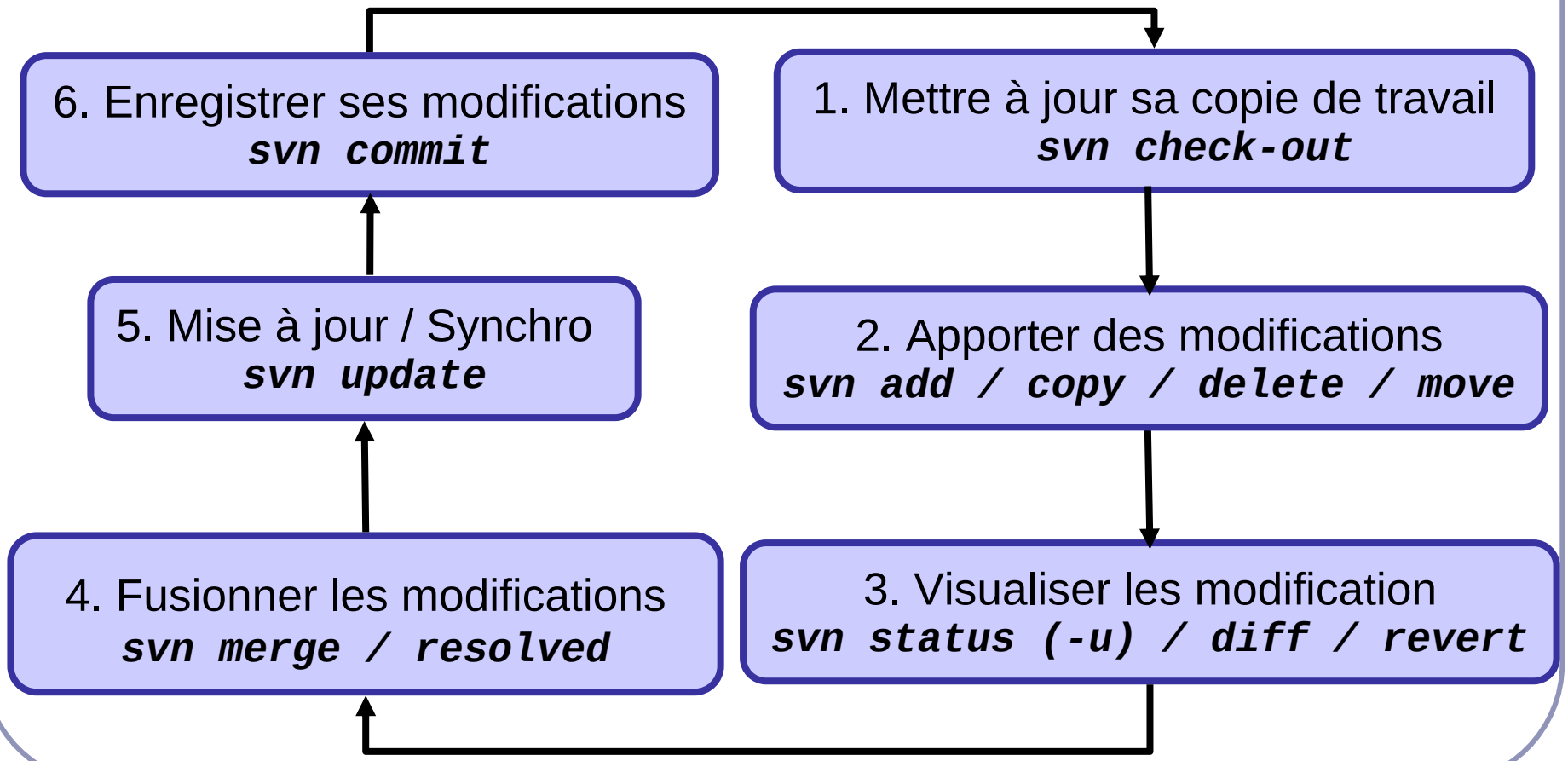
Synchronisation avec conflit



Synchronisation manuelle (subclipse)

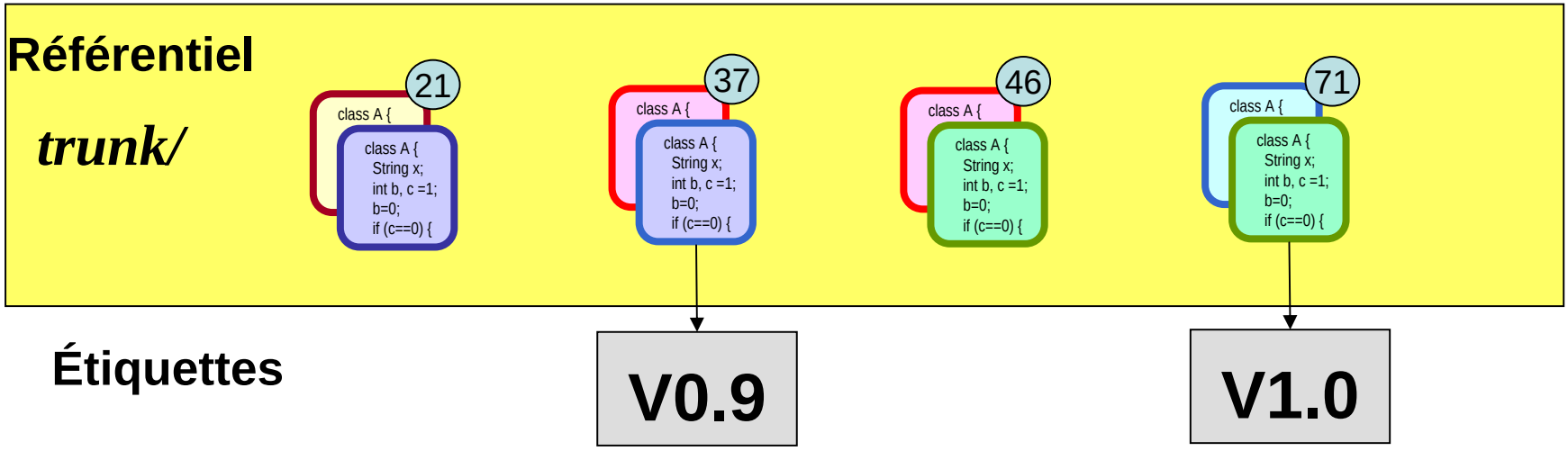


Cycle de travail



Les étiquettes (Tags)

Une étiquette est un nom donné à une révision particulière (version majeure)



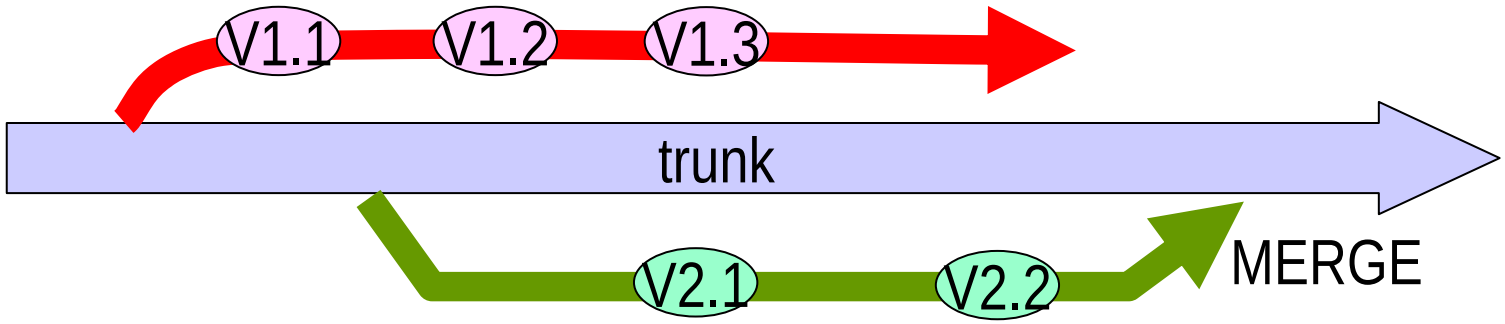
Une étiquette est stockée comme une copie

Branches

Certaines situations peuvent être problématiques:

- Variantes pour spécificités clients
- Correction d'une anomalie dans une version ➔report de la correction dans les autres versions

Une branche est une ligne de développement parallèle



Gestion de la production

- Pour automatiser la construction du projet
  - Compilation, vérifications, tests, packaging,déploiement.
- Pour rationaliser et standardiser l'organisation des projets
  - Arborescence de répertoires normalisée
  - Gestion des dépendances entre composants et ressources
- Pour maintenir le projet en cohérence
  - Génération d'un site web de documentation
  - Intégration simple de rapports et de métriques projet

Maven

- Logiciel libre du consortium Apache
- Gestion et automatisation de la production de projets Java
- Successeur de Ant
- Dernière version 3.0.4

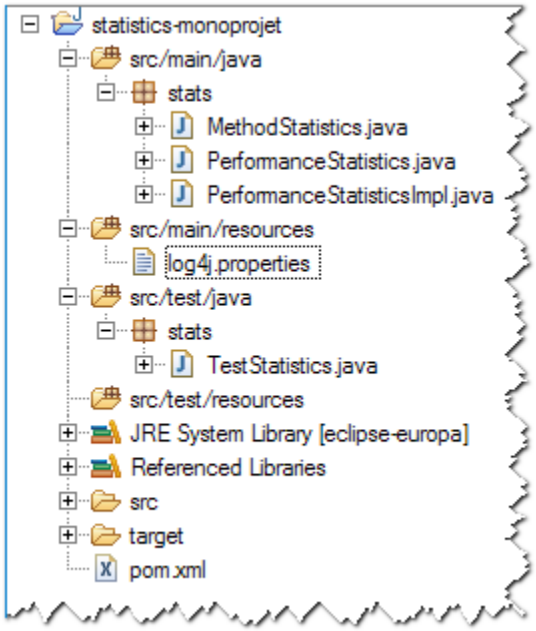
Maven.apache.org

Le fichier POM

- Un fichier XML qui décrit :
  - L'identification du projet
  - Le type de livrable
  - La version du projet
  - Sa description
  - La liste des développeurs
  - Les dépendances
  - Les actions de build
  - …

```
<?xml version="1.0" encoding="UTF-8"?>
<project>
 <modelVersion>4.0.0</modelVersion>
 <groupId>com.octo</groupId>
 <artifactId>webapp-sample</artifactId>
 <version>1.1-SNAPSHOT</version>
 <packaging>war</packaging>
 <name>Simple webapp</name>
 <inceptionYear>2007</inceptionYear>
 <dependencies>
  <dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-struts</artifactId>
   <version>2.0.2</version>
  </dependency>
  …
 </dependencies>
</project>
```

Arborescence d'un projet Maven



- Facilite l'intégration sur le projet
- Peut être changée au besoin

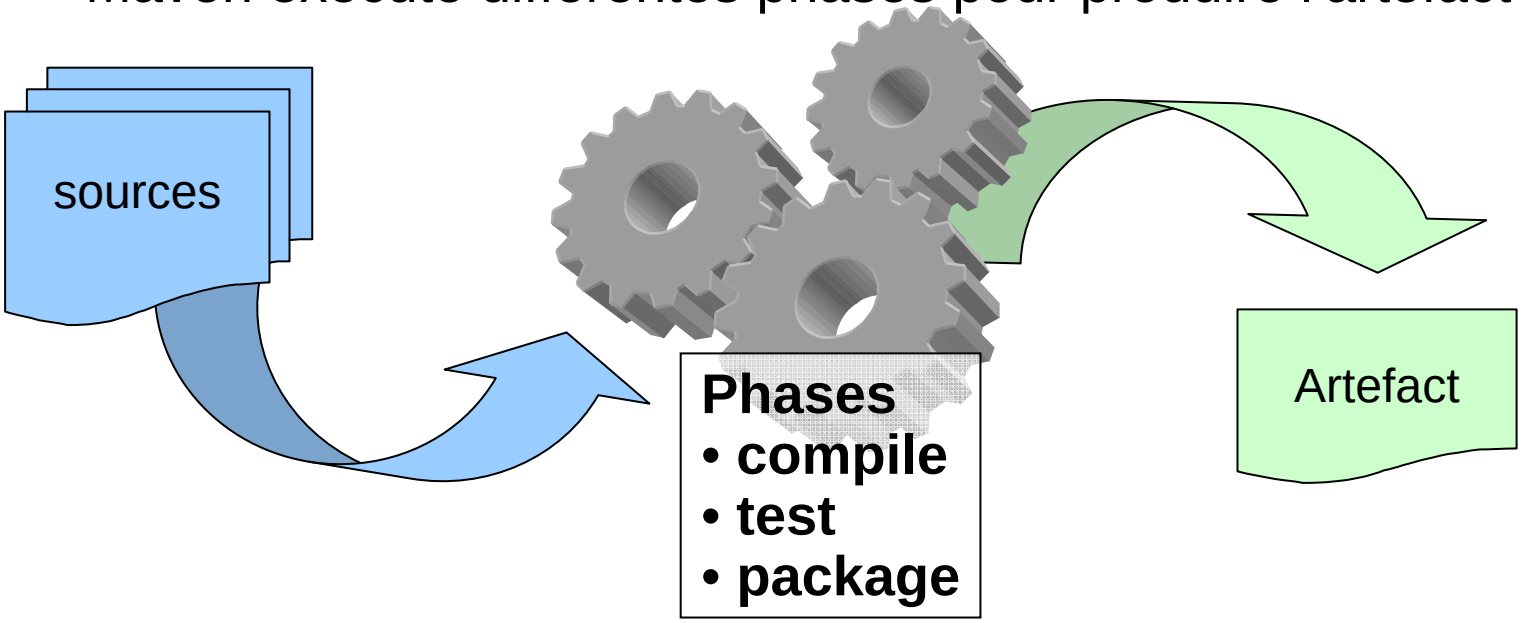
Répertoires standard :

- **sources java : src/main/java**
- **Ressources : src/main/resources**
- **Test java : src/test/java**
- **Ressources de test : src/test/resources**

Le fichier pom.xml est à la racine du projet.

Cycle de production Maven

- Un projet Maven correspond à un artefact
- Maven génère l'artefact à partir des sources du projet
- Maven exécute différentes phases pour produire l'artefact

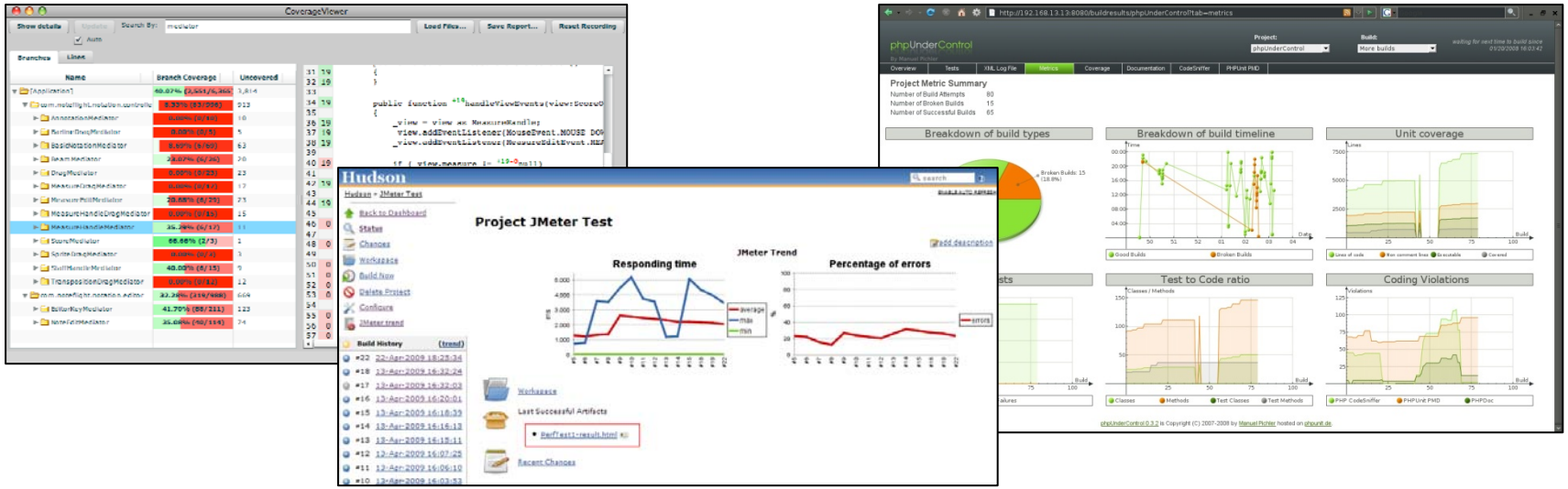


Les repository Maven

- Un repository permet de stocker et de référencer des artefacts produits avec Maven ou des dépendances utilisées par Maven.
- Il existe trois catégories de repository
  - Le repository distant (sur le net)
  - Le repository interne (repository optionnel interne à l'entreprise)
  - Le repository local (poste développeur)

Intégration continue

- Les développeurs « committent »
- Le serveur d'intégration surveille le référentiel et reconstruit le logiciel périodiquement
- Le chef de projet analyse le reporting



Les technologies existantes

- Hudson
- CruiseControl / CruiseControl.NET
- Apache Continuum
- QuickBuild (open-source: LuntBuild)
- Et beaucoup d'autres …

Gestion des modifications

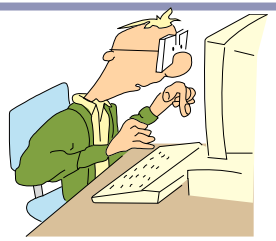
Après avoir identifier les objets et les liens entre objets, le système de gestion de configurations doit permettre de gérer des évolutions

- version/itération/variante
- suivi d'un cycle de modification
- gestion des dépendances inter-objets

A chaque objet on peut associer

- une liste de problèmes connus
- une liste d'évolutions demandées
- un historique des modifications réalisées

Traitement des faits techniques



Utilisateur

description

description

Valideur

Enregistrement (référence)

OUVERT

prise en compte

rejet "sans objet"

Commission de Suivi

A TRAITER

classement évolution

classement anomalie(criticité)

A PLANIFIER

A CORRIGER

développement

correction

A VALIDER

validation/livraison

FERME

Niveaux de criticité

- **CRITIQUE = BLOQUANT**
  - ARRET TOTAL

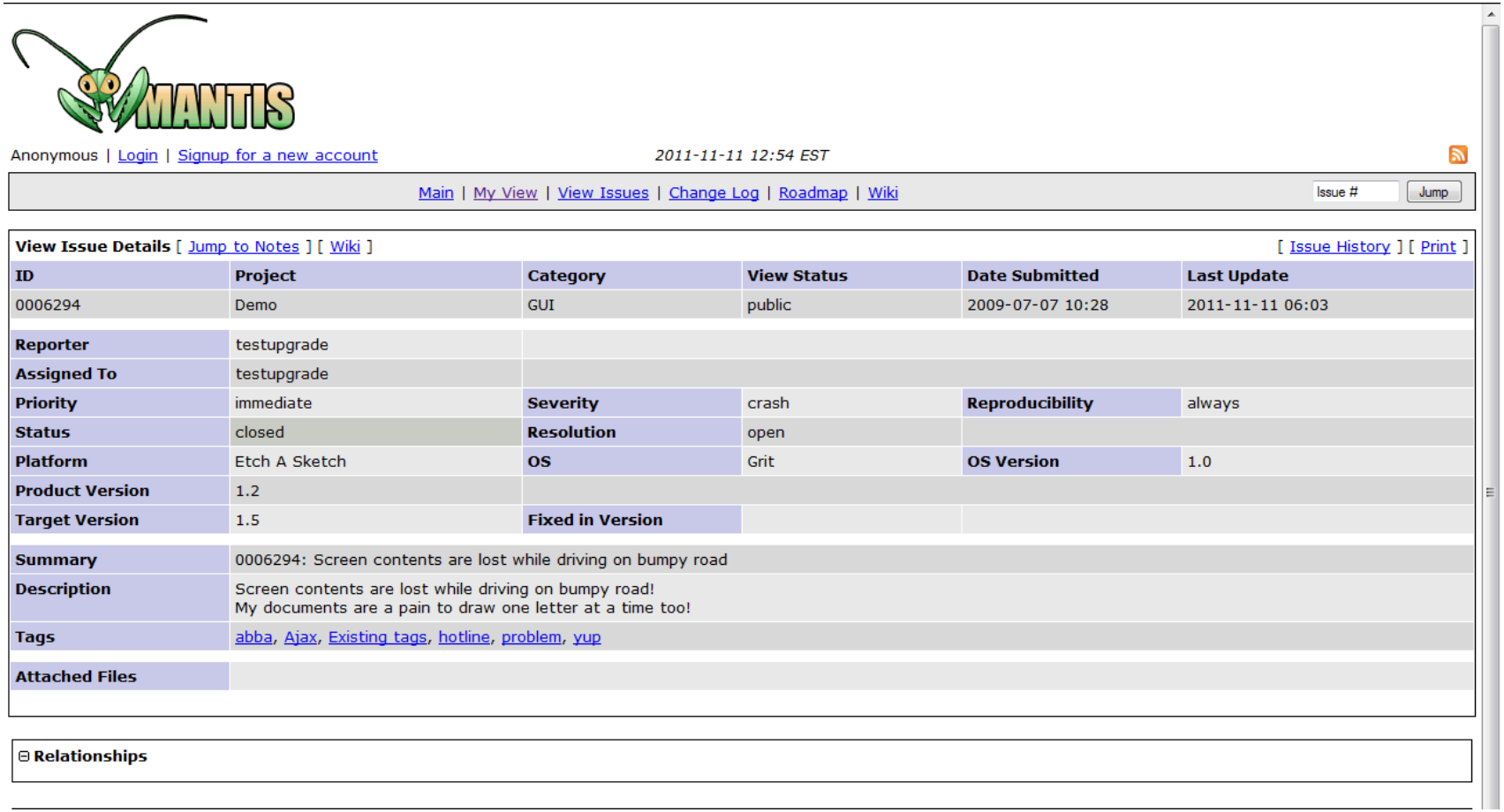
    EX: Plantage, Erreur d'installation, impossibilité de lancer le programme.
- **MAJEUR = TRES GENANT**
  - DEFAUT DE PRECISION OU DE LOGIQUE INACCEPTABLE, IMPOSSIBLE A EVITER PAR UNE ADAPTATION DE LA NOTICE D'EXPLOITATION DU LOGICIEL**.**

    EX: Résultat de calcul faux, Affichage de valeurs incohérentes
- **MINEUR = GENANT**
  - QUE L'ON PEUT EVITER PAR UNE ADAPTATION DE LA NOTICE D'EXPLOITATION DU LOGICIEL.

    EX: Message peu explicite, Difficulté d'accès à une commande
- **SECONDAIRE = SANS CONSEQUENCE**
  - DEFAUT DE COHERENCE OU DE PRESENTATION N'AFFECTANT PAS LE RESULTAT.

    EX: Faute d'orthographe dans un message d'erreur.

Mantis


MANTIS
Anonymous | Login | Signup for a new account
2011-11-11 12:54 EST
Main | My View | View Issues | Change Log | Roadmap | Wiki
Issue #
Jump
View Issue Details [Jump to Notes] [Wiki]
[Issue History] [Print]
ID | Project | Category | View Status | Date Submitted | Last Update
0006294 | Demo | GUI | public | 2009-07-07 10:28 | 2011-11-11 06:03
Reporter | testupgrade
Assigned To | testupgrade
Priority | immediate | Severity | crash | Reproducibility | always
Status | closed | Resolution | open
Platform | Etch A Sketch | OS | Grit | OS Version | 1.0
Product Version | 1.2
Target Version | 1.5 | Fixed in Version
Summary | 0006294: Screen contents are lost while driving on bumpy road
Description | Screen contents are lost while driving on bumpy road!
My documents are a pain to draw one letter at a time too!
Tags | abba, Ajax, Existing tags, hotline, problem, yup
Attached Files
Relationships

G.C.L. et définition

- **Pendant la phase de spécification**
  - Rédiger le plan de G.C.
  - Préparer les outils de G.C.
  - Archiver les spécifications
  - Archiver les maquettes, les rapports d'étude (s ’il y a lieu)
- **Pendant la phase de conception**
  - Identifier les composants logiciels et les interfaces entre composants (architecture)
  - Créer et structurer les espaces logiques
  - Développer et mettre en place les chaînes de production
  - Archiver le document d’architecture, les plans d'intégration et de validation
  - Contrôler les modifications

De la réalisation à la maintenance

- **Pendant les phases de réalisation**
  - Archiver le code et les jeux d'essai
  - Contrôler les modifications
- **Pendant les phases d'intégration et de validation**
  - Intégrer les composants et générer les modules
  - Assembler et produire les versions
  - Contrôler les modifications
  - Archiver les dossiers de tests
- **Pendant les phases d'exploitation et de maintenance**
  - Archiver et livrer les configurations
  - Contrôler les modifications
  - Suivre les demandes d'évolution

Fiche de version - Plan type

1. **Introduction**
2. **Documents applicables :**
3. **Définition de la version**
4. **Identification des livrables et des supports**
5. **Moyens de production**
6. **Environnement de mise en oeuvre**
7. **Procédure d'installation**
8. **Procédure de mise en oeuvre**
9. **Compatibilité avec la version précédente**
10. **Corrections**
11. **Évolutions**
12. **Limitations**