Développement collaboratif

Franck Silvestre <franck.silvestre@ticetime.com>

Table of Contents

Développement collaboratif

Développement collaboratif

Développer de manière collaborative, c'est travailler à plusieurs sur la même base de code pour produire un logiciel.

1. Partager juridiquement le code source

Le logiciel est régi par le droit d'auteur

Warning

This HTML code couldn't be converted: <tag0:em xmlns:tag0="http://www.w3.org/1999/xhtml">Code de la proprie#te# intellectuelle - Article L113-9</tag0:em>

Sauf dispositions statutaires ou stipulations contraires, les droits patrimoniaux sur les logiciels et leur documentation cre#e#s par un ou plusieurs employe#s dans l'exercice de leurs fonctions ou d'apre#s les instructions de leur employeur sont de#volus a# l'employeur qui est seul habilite# a# les exercer. Toute contestation sur l'application du pre#sent article est soumise au tribunal de grande instance du sie#ge social de l'employeur. Les dispositions du premier aline#a du pre#sent article sont e#galement applicables aux agents de l'Etat, des collectivite#s publiques et des e#tablissements publics a# caracte#re administratif.

Logiciel développé par des employés

- Logiciel de#veloppe# par les employe#s d'une me#me entite#
  - Le code source est produit par les employe#s
  - Les droits patrimoniaux appartiennent a# l'employeur
- Exemples
  - CATIA - Dassault Syste#me
  - Oracle - Oracle
  - iTunes - Apple

Logiciel libre et entreprise

Exercice

Un logiciel libre peut-il "appartenir" à une entreprise ?

Logiciel libre et entreprise

FAQ MySql [http://www.mysql.com/about/legal/licensing/oem/#4]

Q4: What is Oracle’s dual license model for MySQL software?

A: Oracle makes its MySQL database server and MySQL Client Libraries available under both the GPL and a commercial license. As a result, developers who use or distribute open source applications under the GPL can use the GPL-licensed MySQL software, and OEMs, ISVs and VARs that do not want to combine or distribute the MySQL software with their own commercial software under a GPL license can purchase a commercial license.

Logiciels développés par une communauté

FAQ PostgreSql [http://www.postgresql.org/about/press/faq/]

Q: What company owns PostgreSQL?
A: None. We are an unincorporated association of volunteers and companies who share code under the PostgreSQL License. The PostgreSQL project involves a couple dozen companies who either support PostgreSQL contributors or directly contribute corporate projects to our repository. Some of our major corporate sponsors are on the sponsors page, and there are many more companies who contribute to the project in other ways.

Logiciel libre, entreprise et communauté

Exercice

Un logiciel libre appartenant a# une entreprise et de#veloppe# par une communaute#, c'est possible ?

Logiciel libre, entreprise et communauté

Wiki Oracle MySql [https://wikis.oracle.com/display/mysql/Contributing +Code+to+MySQL]

If you expect to make code-related contributions, you must sign and return theOracle Contributor Agreement (OCA). Without an OCA on file, Oracle cannot integrate your contribution into the MySQL code base or engage in extended discussions on proposed patches. The OCA gives Oracle and the contributor joint copyright interests in the code: The contributor retains copyright while also granting those rights to Oracle as the open-source development project sponsor. The OCA is applicable to all products and projects owned or managed by Oracle; signing it once means you can contribute code to any Oracle-sponsored open-source project. More detail, and the OCA itself, is available on the Oracle Contributor Agreement page.

2. Partager techniquement le code source

Version Control System (VCS)

- Gestion de version de code source
- Permet de ge#rer les changements sur le code source
- Prise en compte des modifications provenant de plusieurs contributeurs
- Re#vision : un lot de changements reporte# par un contributeur induit une nouvelle version du code source

- Possibilite# de retrouver a# tout moment le code source correspondant a# une re#vision donne#e
- Et bien d'autres choses...

Exemples de VCS

- Open Source
  - Git - http://git-scm.com/
  - SVN - https://subversion.apache.org/
  - Mercurial - http://mercurial.selenic.com/
  - CVS - http://cvs.nongnu.org/
- Propriétaires
  - Perforce - http://www.perforce.com/
  - Clearcase - http://www-03.ibm.com/software/products/en/clearcase

3. Construire de manière uniforme un projet

"Build" d'un logiciel

http://en.wikipedia.org/wiki/Software_build

> In the field of computer software, the term software build refers either to the process of converting source code files into standalone software artifact(s) that can be run on a computer, or the result of doing so. One of the most important steps of a software build is the compilation process where source code files are converted into executable code.

Le "build" en question

- Comment fournir un système de build uniforme ?
- Comment garantir la mise en oeuvre de bonnes pratiques ?
- Comment gérer les dépendances de manière uniforme ?

Maven

- Gestion de build et de dépendances développé en Java
- http://maven.apache.org/
- Objectifs
  - rendre le processus de build facile
  - fournir un système de build uniforme
  - fournir des informations de qualité sur le logiciel

- fournir un cadre pour les bonnes pratiques de développement

Autres outils de build

- Gradle - http://www.gradle.org/
- builder - http://buildr.apache.org/

4. Gérer les exigences

Référentiel d'exigences

- Objectifs
  - Maintenir dans le temps (trac#abilite#) la liste des exigences fonctionnelles, techniques, bogues, autres
  - Donner de la visibilite# aux parties prenantes du projet (utilisateurs, clients,...)
- Outils
  - Outils spe#cialise#s
  - Outils de gestion de bogues de#tourne#s
  - Mantis, Bugzilla, Issues Github, Jira
- Illustration
  - Ruby On Rails - https://github.com/rails/rails/issues
  - Grails - http://jira.grails.org/browse/GRAILS

IDE - VCS - Exigences

- Bonne pratique : associer chaque re#vision a# une exigence
- Certains IDE permettent l'interfac#age a# la fois avec le VCS et gestionnaire d'exigences.
- Exemples
  - Plugin gestion de ta#ches IntelliJ
  - Plugin Mylin pour Eclipse
  - Illustration avec Tsaap-Notes - https://github.com/TSaaP/tsaap-notes

5. Garantir la qualité du code

Tester, tester et tester encore

- Tests unitaires
- Tests d'inte#gration

- Tests fonctionnels

L'automatisation des tests pre#vient les re#gressions.

Outils de test

- Tests unitaires et d'inte#gration de projets Java
  - JUnit - http://www.junit.org
  - Spock - https://code.google.com/p/spock
  - TestNG - http://testng.org/
- Tests fonctionnels d'applications Web
  - Selenium - http://seleniumhq.org
  - Geb - http://www.gebish.org/

Analyse statique de code

- Analyse statique de code
  - de#tection des violations de re#gles de codage
  - calcul de me#triques (complexite#, taille des me#thodes)
  - de#tection des bugs (si, si, si mais pas tous)
- En Java
  - Checkstyle [http://checkstyle.sourceforge.net/]
  - PMD [http://pmd.sourceforge.net/]
  - Findbugs [http://findbugs.sourceforge.net/]

6. Délivrer le logiciel

Délivrer le logiciel

- Figer l'ensemble des sources
  - le code source
  - la documentation
  - la liste des exigences associe#es
- Empaqueter les livrables
  - les binaires

    - la documentation
    - ...
- De#livrer les livrables

Exemple de workflow

- De#finition de la version du logiciel
- Exemples de pattern de nume#ro de version
    - major.minor.maintenance
    - major.minor.build.revision
    - annee.mois
- Sur le de#po#t de code source
    - Fusions ne#cessaires
        - cre#ation d'un tag correspondant a# la version
- Construction des livrables
    - Compilation, tests, paquetages,...
- De#ploiement des livrables
    - Sur un de#po#t d'artefact
    - Sur une ou plusieurs plateformes d'exploitation (test, de#monstration, pre#-prod, prod ...)

Plateforme d'intégration continue

Que permet une PIC ?

Le de#clenchement de manie#re automatique et en continu de...

- la construction du projet
    - compilations, tests, analyses statiques de code
- le de#ploiement des livrables
- La ge#ne#ration et la pre#sentation de rapports
    - Re#sultats des tests, re#sultats des analyses statiques)

En continue : à chaque révision, à période régulière,...

Les outils

- Serveur d'inte#gration continue

    - Jenkins - http://jenkins-ci.org/
    - Travis - https://travis-ci.org/
- Plateforme de gestion de la qualite# du code
    - Sonarqube - http://www.sonarqube.org/