

Bonnes pratiques de développement

Claire Mouton - CREATIS, Lyon, France
claire.mouton@creatis.insa-lyon.fr

En collaboration avec
Vincent Miele - CNRS - Biométrie & Biologie Evolutive

Plan

I. Choisir un langage (Vincent)

II. Adopter une méthode de développement

III. Coder

IV. Partager, travailler en équipe

V. Profiling

VI. Débugger

II. Adopter une méthode de développement

I. Choisir un langage (Vincent)

II. Adopter une méthode de développement

III. Coder

IV. Partager, travailler en équipe

V. Profiling

VI. Débugger

II. Adopter une méthode de développement

J'ai choisi mon langage... Comment je procède maintenant?

J'aimerais pouvoir réutiliser et modifier mon code plus tard!

Et si je travaille en équipe?

II. Adopter une méthode de développement

Enthousiaste, je code dans un fichier...

qui finit par atteindre des milliers de ligne!

→ Comment en réutiliser une partie?

Ou faire une modification ?

Bonne pratique :

Séparer le code en plusieurs fichiers, chacun ayant une unité sémantique (fonction, classe),

un programme principal (main) y fera appel.

→ Concevoir l'architecture du logiciel en amont de l'écriture

II. Adopter une méthode de développement

Gestion de projet logiciel

- Penser à l'avance l'architecture du code (UML, ...)
- Penser à l'évolution du code lors de son développement (cycle en V, ...)
- Méthode agile (Xtreme programming, ...)
- Méthode d'intégration continue (tests de non-régression, tests unitaires, ...)

II. Adopter une méthode de développement

Unified Modeling Language (UML)



Dessins qui résument les dépendances, la structure ou les fonctionnalités du programme

II. Adopter une méthode de développement

Unified Modeling Language (UML)

Langage de modélisation unifié

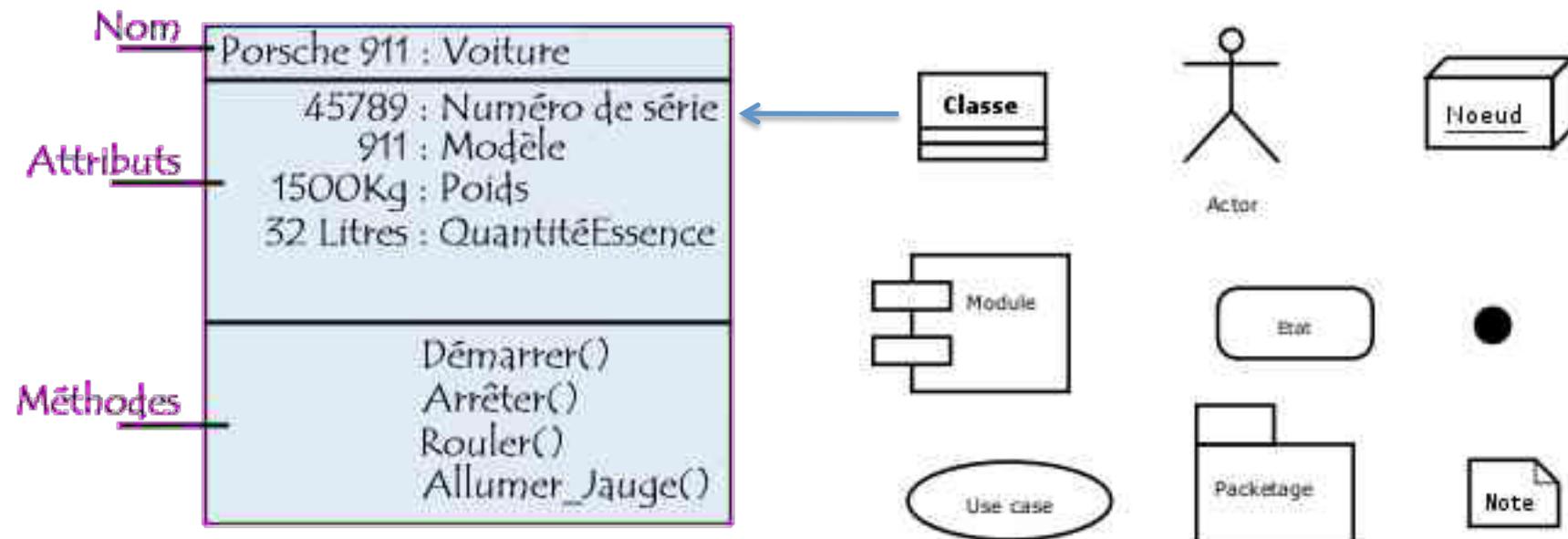
Notation permettant de modéliser un problème de façon standard

- Très lié au monde de la programmation objet
- Il existe un formalisme complet
- Le connaître et l'utiliser peut être utile:
 - pour mettre à plat le problème, réfléchir sur le programme avant de commencer à coder
 - pour être compris par les autres
 - pour comprendre les autres

II. Adopter une méthode de développement

Unified Modeling Language (UML)

Il existe un formalisme complet:



II. Adopter une méthode de développement

Unified Modeling Language (UML)

Éléments importants de l'UML : les « diagrammes »
13 diagrammes.

Les 2 diagrammes les plus utiles:

- Les diagrammes de classes
- Les diagramme de cas d'utilisation

II. Adopter une méthode de développement

Unified Modeling Language (UML)

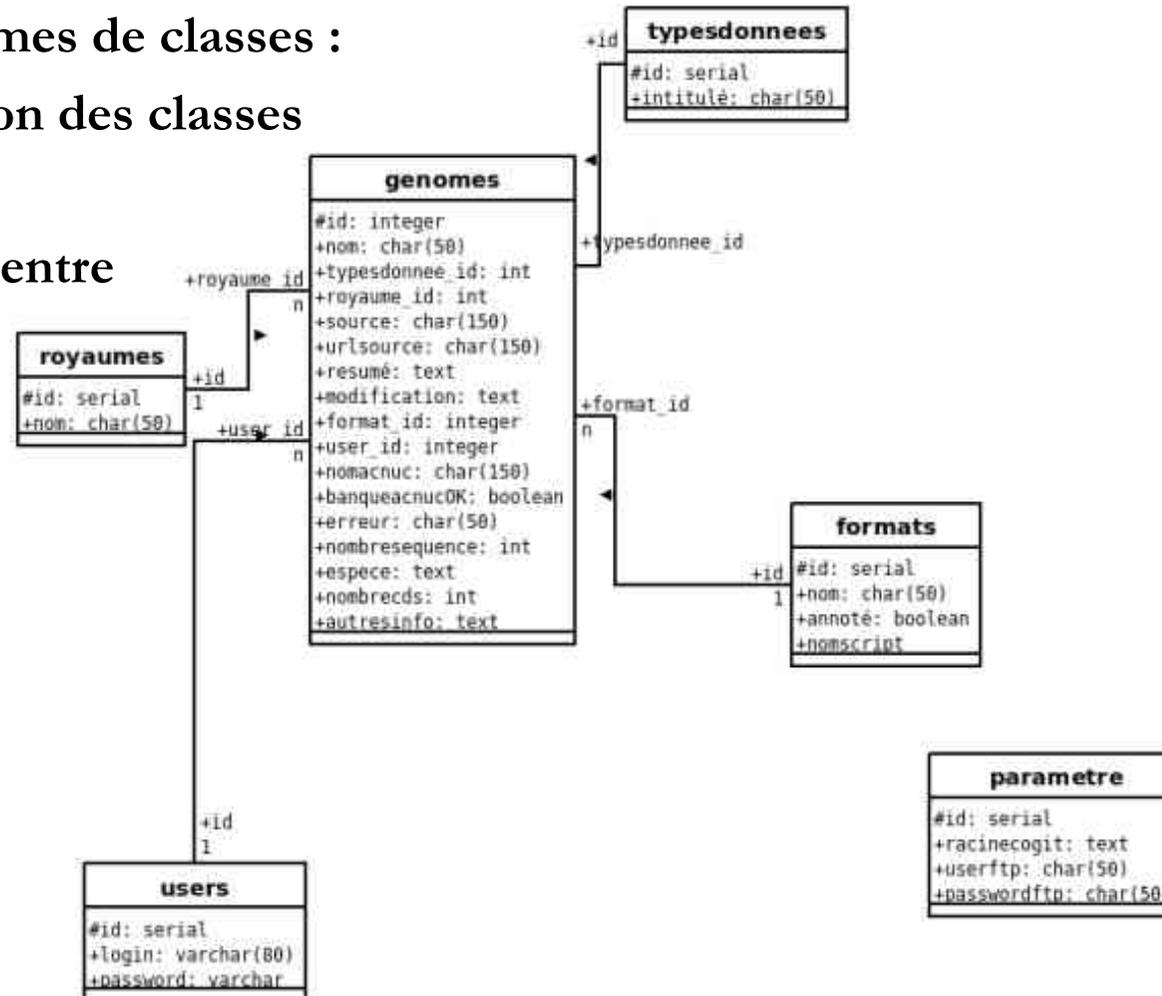
Les diagrammes de classes :

Représentation des classes

et

des relations entre

celles-ci

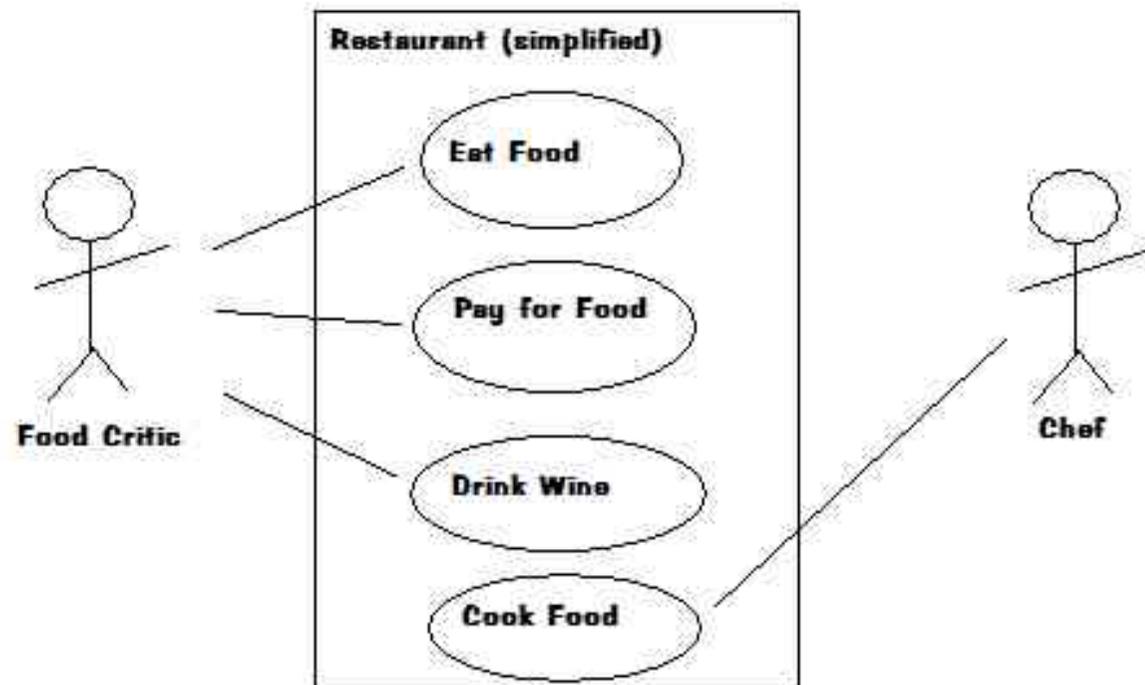


II. Adopter une méthode de développement

Unified Modeling Language (UML)

Les diagrammes de cas d'utilisation :

Vision globale du comportement fonctionnel d'un logiciel



II. Adopter une méthode de développement

Unified Modeling Language (UML)

Les outils logiciels pour faire des diagrammes

UMLet

Dia

XMind

ArgoUML : open source

StarUML : licence modifiée de GNU GPL

reverse engineering : à partir du code, génère le diagramme

ou à partir du diagramme : génère le code en Python, C#, ...

BOUML

draw.io : création de diagrammes UML collaboratif, connecté à Github-Google Drive

PlantUML : si seule la représentation visuelle compte (sans génération de code)

Plugin d'Eclipse

[Comparaison des logiciels d'UML](#)

II. Adopter une méthode de développement

Unified Modeling Language (UML)

Les outils logiciels pour faire des diagrammes

[Comparaison des logiciels d'UML](#)

Général [[modifier](#) | [modifier le code](#)]

Nom	Créateur	Plateforme / OS	Dernière diffusion stable	Open source ?	Licence logiciel	Langage de programmation utilisé
AmaterasUML	122	Multiplate-forme (Java)	16 juillet 2012	Oui	EPL v1.0	Java
ArgoUML		Multiplate-forme (Java)	15 décembre 2011	Oui	EPL v1.0	Java
Astade		Multiplate-forme	11 mars 2010	Oui	GPL	Basé sur wxWidgets
BOUML (version < 5.0 non maintenu)	Bruno Pagès	Microsoft Windows et Linux	1 ^{er} octobre 2011	Oui	GPL ¹	C++/Qt3, Java ("plug-out")

Fonctionnalités [[modifier](#) | [modifier le code](#)]

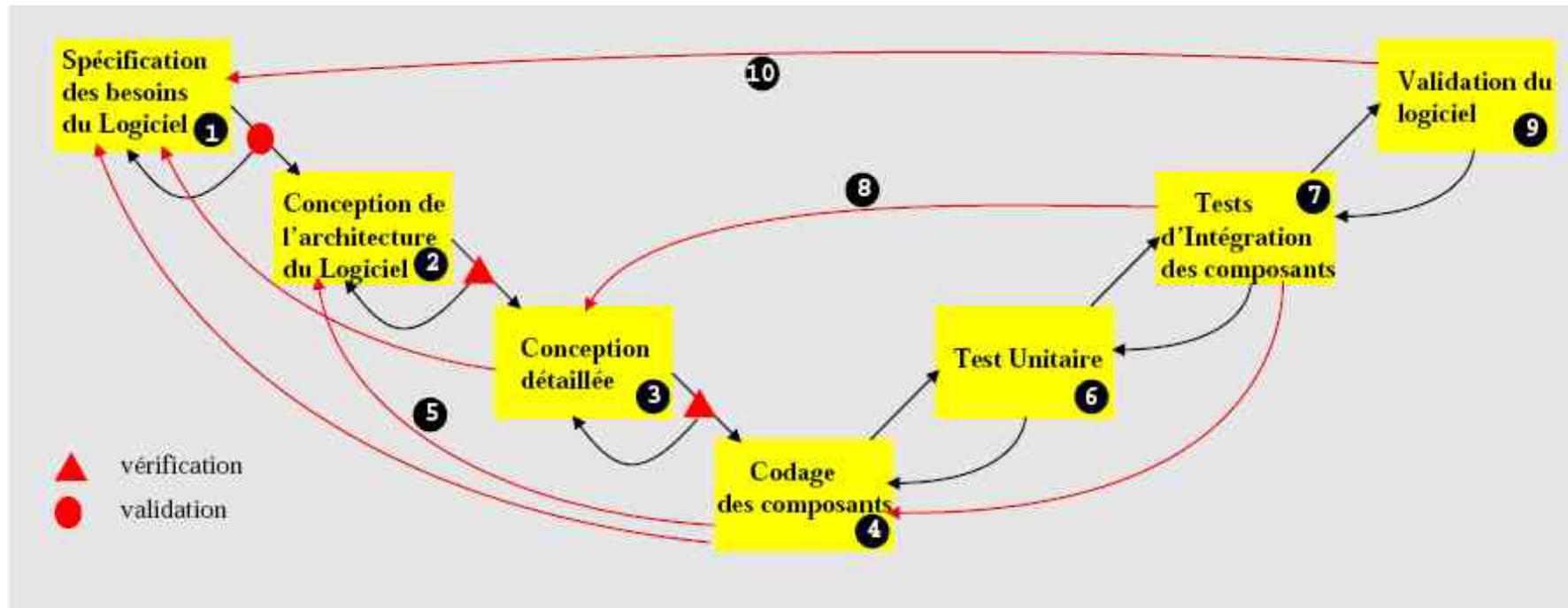
Nom	UML 2 ?	MDA ?	XMI ?	Patrons ?	Langages générés	Langages de rétro-génération	Intégrable dans	Détails
Acceleo	?	Oui	?	Oui	JEE, C#, Java, PHP, Python.		Eclipse, EMF	
AmaterasUML	?	?	?	22222	Java	Java		Standard UML
ArgoUML	Non	Oui	Oui	?	C++, C#, PHP4, PHP5, Java, Ruby, SQL	Java (et autres langages, dont C# en plugins)		Suit de près les standards UML
Astade	?	?	?	?	C, C++			
ATL	?	?	?	?			Disponible via le projet Eclipse M2M (Model to Model).	Peut transformer des modèles UML en d'autres modèles

II. Adopter une méthode de développement

Cycle en V

Vision en un bloc

En réalité, le cycle de développement se déroule ainsi :



II. Adopter une méthode de développement

Extreme Programming

Agile Manifesto, signé par 17 personnalités du Génie Logiciel en 2001 :

- Un processus de développement
- Un état d'esprit
- Un ensemble de bonnes pratiques

→ Adapté aux collaborations ingénieur / chercheur pour lesquels les besoins sont fluctuants

II. Adopter une méthode de développement

Extreme Programming

Développement piloté par les tests

Règles de codage

Conception simple

Livraisons fréquentes

Travail en binôme

Metaphore

Integration continue

Remaniement

Rythme durable

Client sur site

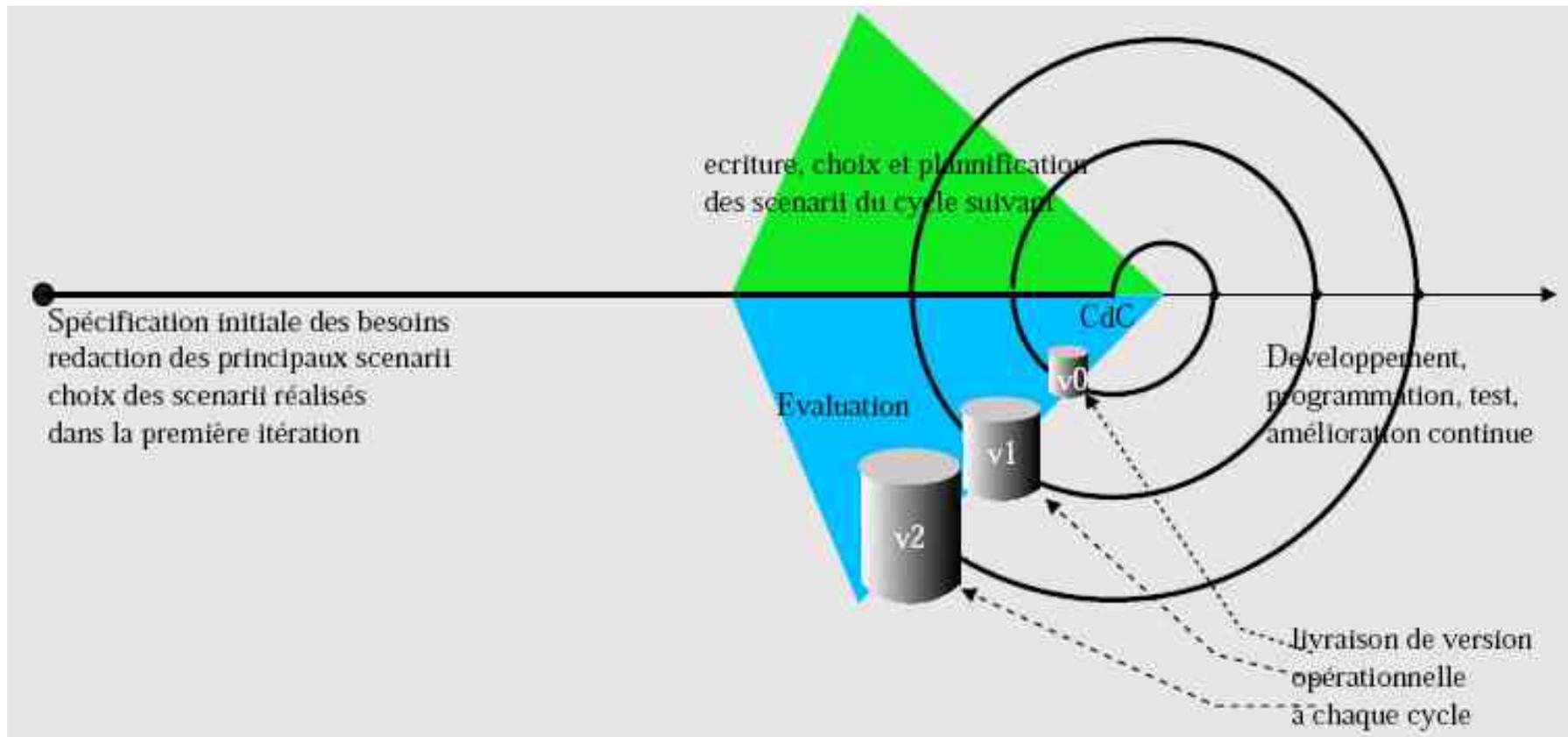
Responsabilité collective du code

Planification itérative



II. Adopter une méthode de développement

Extreme Programming



II. Adopter une méthode de développement

Extreme Programming : Intégration Continue

Vérifier à chaque modification de code source que le résultat des modifications ne produit pas de régression dans l'application développée.

Prérequis:

- partage du code source via un gestionnaire de version
- intégration quotidienne des modifications par les développeurs
- développement de tests d'intégration de l'application

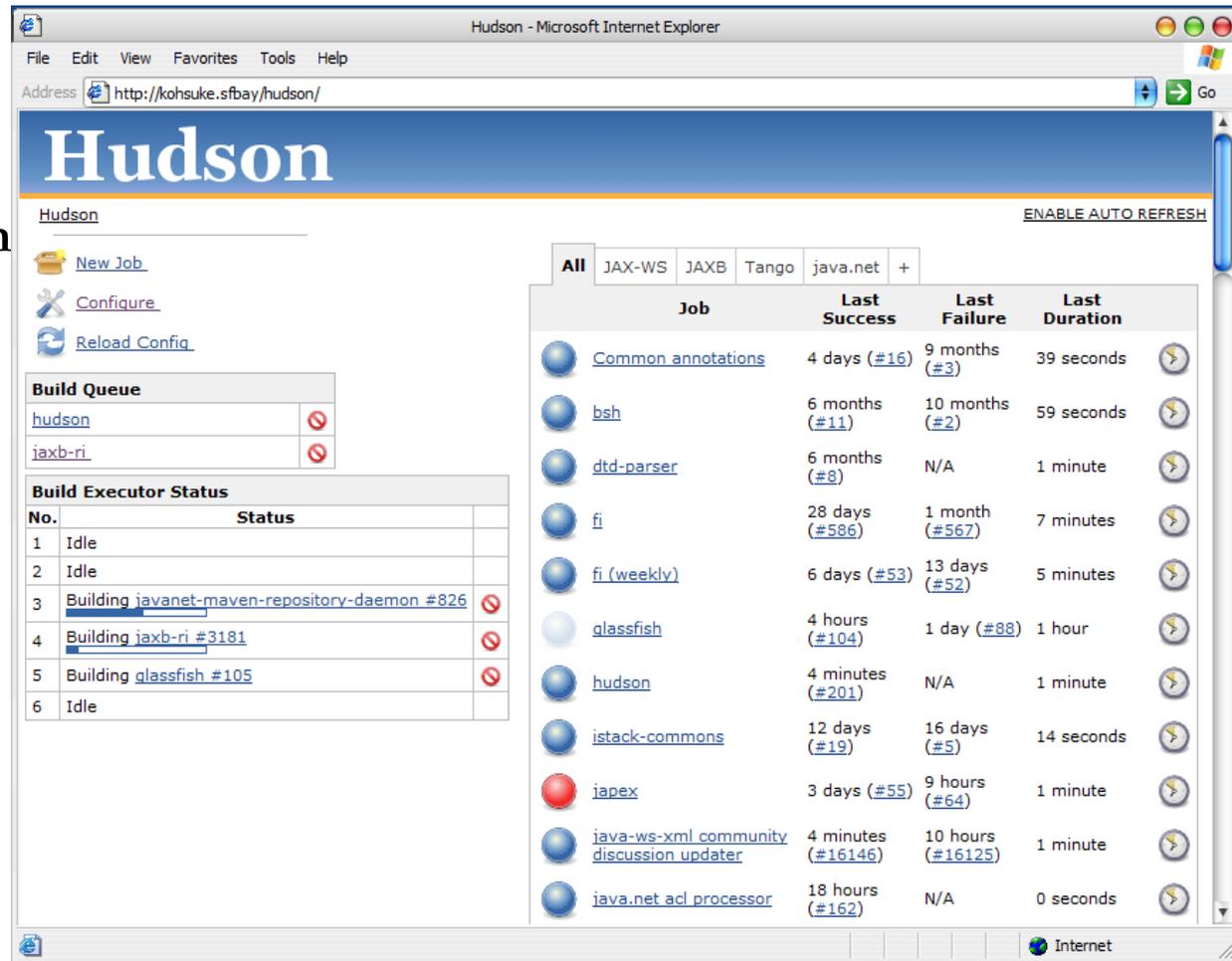
Avantages :

- le test immédiat des unités modifiées
- la prévention rapide en cas de code incompatible ou manquant
- les problèmes d'intégration sont détectés et réparés de façon continue, évitant les problèmes de dernière minute
- une version est toujours disponible pour un test, une démonstration ou une distribution

II. Adopter une méthode de développement

Extreme Programming : Intégration Continue

Exemple : Jenkins
(ex Hudson),
serveur d'intégration
continue pour Java



The screenshot shows the Hudson web interface in a Microsoft Internet Explorer browser window. The address bar shows <http://kohlsuke.sfbay/hudson/>. The page title is "Hudson".

On the left side, there are navigation links: [New Job](#), [Configure](#), and [Reload Config](#).

The "Build Queue" section shows a table with the following data:

Job	Status
hudson	⊘
jaxb-ri	⊘

The "Build Executor Status" section shows a table with the following data:

No.	Status
1	Idle
2	Idle
3	Building javanet-maven-repository-daemon #826 ⊘
4	Building jaxb-ri #3181 ⊘
5	Building glassfish #105 ⊘
6	Idle

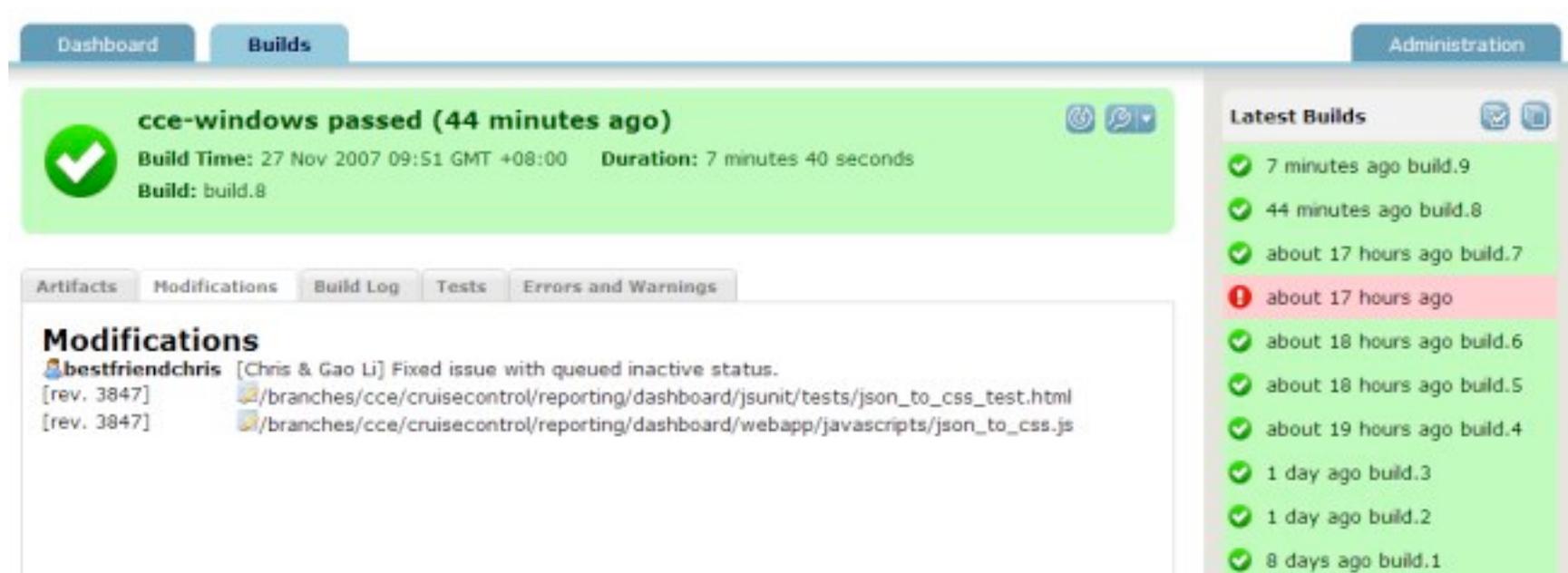
The main content area displays a list of jobs with the following columns: Job, Last Success, Last Failure, and Last Duration. The jobs listed are:

Job	Last Success	Last Failure	Last Duration
Common annotations	4 days (#16)	9 months (#3)	39 seconds
bsh	6 months (#11)	10 months (#2)	59 seconds
dtd-parser	6 months (#8)	N/A	1 minute
fi	28 days (#586)	1 month (#567)	7 minutes
fi (weekly)	6 days (#53)	13 days (#52)	5 minutes
glassfish	4 hours (#104)	1 day (#88)	1 hour
hudson	4 minutes (#201)	N/A	1 minute
istack-commons	12 days (#19)	16 days (#5)	14 seconds
iapex	3 days (#55)	9 hours (#64)	1 minute
java-ws-xml-community-discussion-updater	4 minutes (#16146)	10 hours (#16125)	1 minute
java.net.acl.processor	18 hours (#162)	N/A	0 seconds

II. Adopter une méthode de développement

Extreme Programming : Intégration Continue

Exemple : CruiseControl, logiciel d'intégration continue multilingue

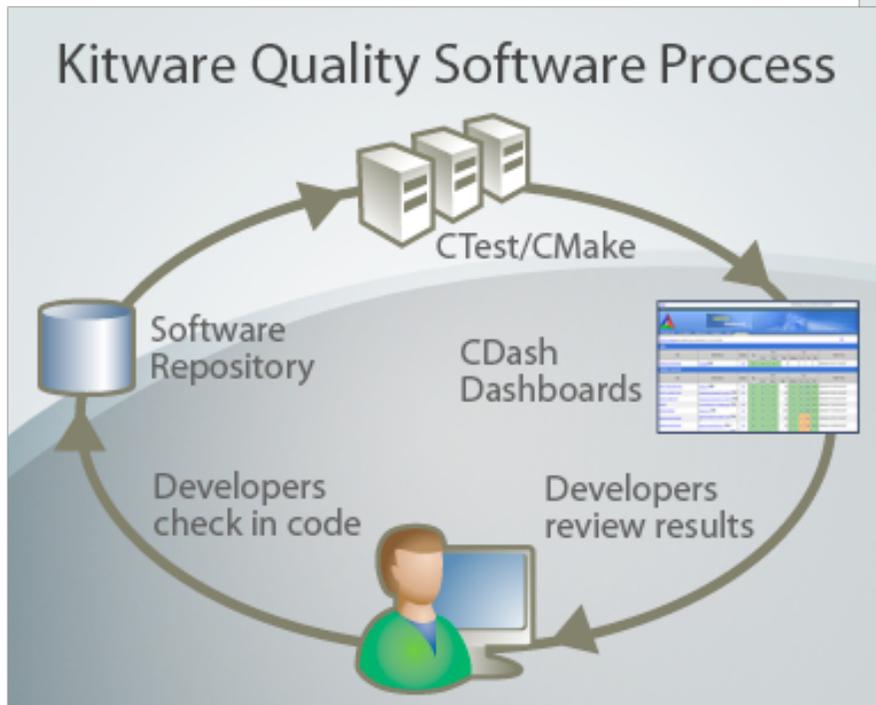


The screenshot displays the CruiseControl web interface. At the top, there are navigation tabs for 'Dashboard', 'Builds', and 'Administration'. The main content area features a large green notification box with a checkmark icon, stating 'cce-windows passed (44 minutes ago)'. Below this, it provides details: 'Build Time: 27 Nov 2007 09:51 GMT +08:00', 'Duration: 7 minutes 40 seconds', and 'Build: build.8'. To the right, a 'Latest Builds' sidebar lists recent builds with their status (green checkmark for success, red exclamation mark for failure) and timestamps. Below the notification, there are tabs for 'Artifacts', 'Modifications', 'Build Log', 'Tests', and 'Errors and Warnings'. The 'Modifications' tab is active, showing a commit by 'bestfriendchris' with the message '[Chris & Gao Li] Fixed issue with queued inactive status.' and two file paths: '/branches/cce/cruisecontrol/reporting/dashboard/jsunit/tests/json_to_css_test.html' and '/branches/cce/cruisecontrol/reporting/dashboard/webapp/javascripts/json_to_css.js', both associated with revision [rev. 3847].

II. Adopter une méthode de développement

Extreme Programming : Intégration Continue

Exemple : CDash, open-source
serveur de test
avec interface web



The screenshot shows the CMAKE Dashboard web interface. At the top, there is a navigation bar with links for **DASHBOARD**, **CALENDAR**, **PREVIOUS**, **CURRENT**, **NEXT**, and **PROJECT**. Below the navigation bar, there is a table of build results. The table has columns for **Site**, **Build Name**, **Update**, **Cty**, **Build** (with sub-columns for **Error**, **Warn**, **Min**), **Test** (with sub-columns for **Success**, **Fail**, **Pass**, **Min**), and **Build Time**. The table lists several build entries, including **200606180100**, **200606180101**, **200606180102**, **200606180103**, **200606180104**, **200606180105**, **200606180106**, **200606180107**, **200606180108**, **200606180109**, and **200606180110**. The table also includes a section for **Nightly Expected** results.

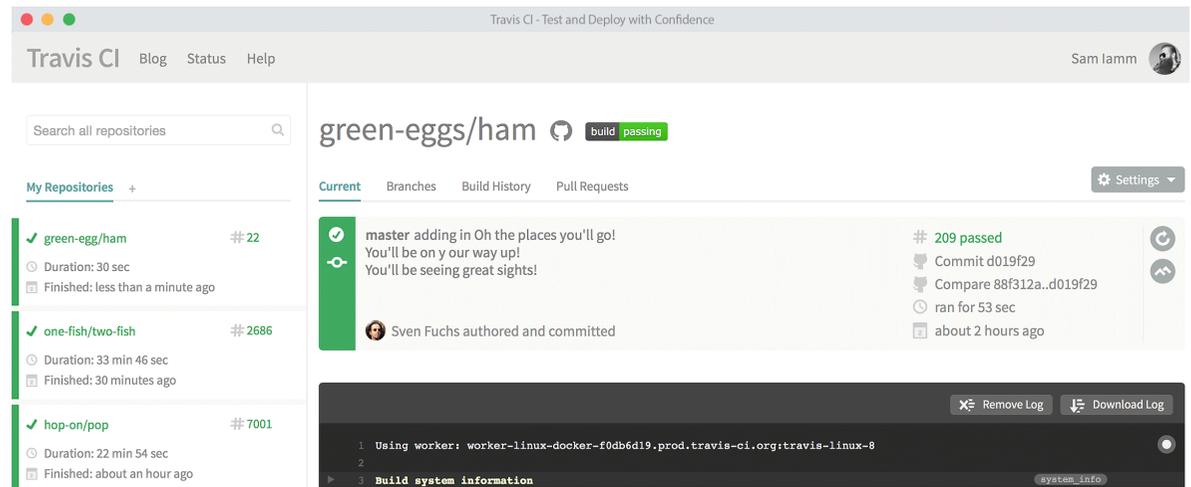
II. Adopter une méthode de développement

Extreme Programming : Intégration Continue

Exemple : Travis, open-source

compilation, exécution dans le cloud (Platform as a Service)

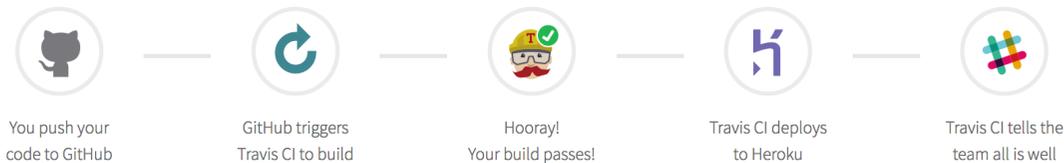
avec interface à Github



The screenshot shows the Travis CI web interface for the repository 'green-eggs/ham'. The interface includes a search bar, a list of repositories, and a detailed view of the current build. The build status is 'passing', and the log shows the following steps:

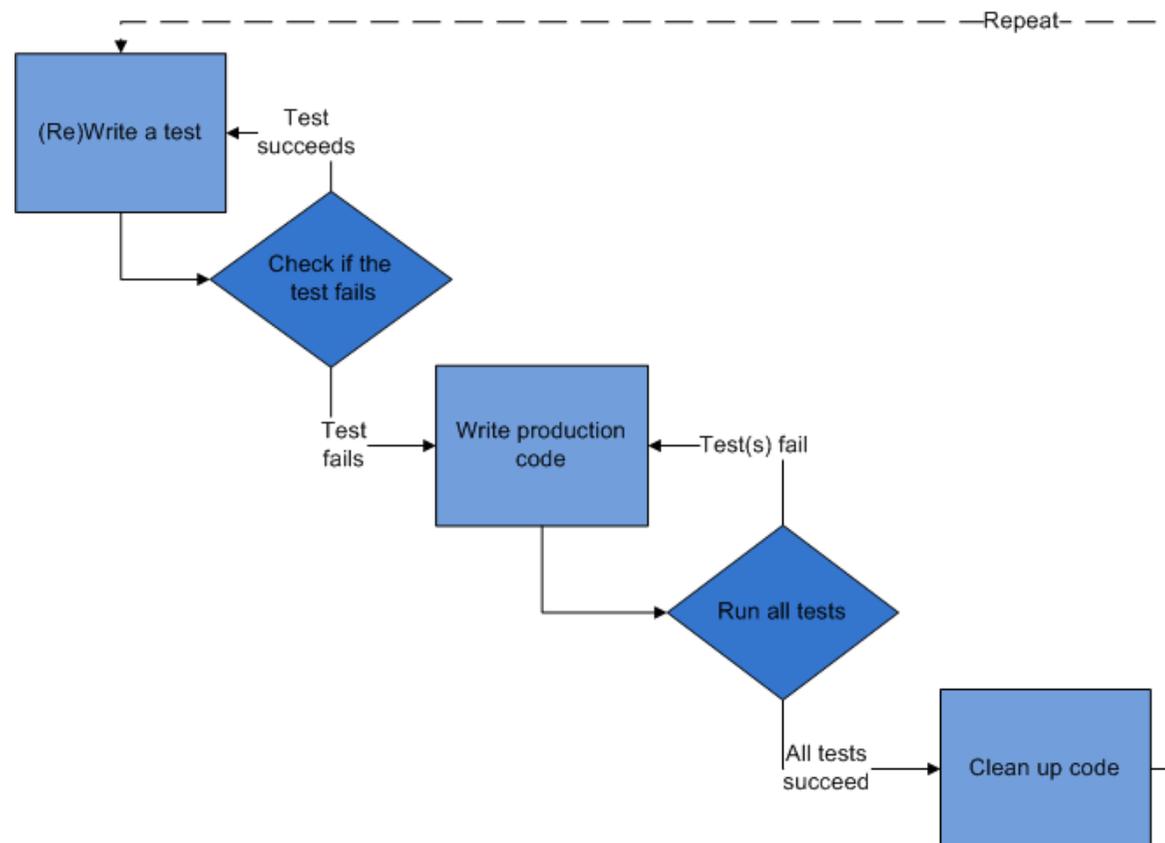
```
1 Using worker: worker-linux-docker-f0db6d19.prod.travis-ci.org:travis-linux-8
2
3 Build system information
  --branch=sf-scenarios git://github.com/travis-ci/travis-github-sync.git
  --git-checkout: 0.15s
  --postgres: 0.3s
  container-based infrastructure, which does not allow use of 'sudo', setuid and setgid
  add 'sudo: required' to your .travis.yml
  s-ci.com/user/workers/container-based-infrastructure/ for details.
  variables from .travis.yml
  cache:1
  call --binary --fuzzy
  LE=$PWD/Gemfile
  rvm
  21.35s
```

Branch build flow



II. Adopter une méthode de développement

Extreme Programming : Développement piloté par les tests



III. Coder

I. Choisir un langage (Vincent)

II. Adopter une méthode de développement

III. Coder

IV. Partager, travailler en équipe

V. Profiling

VI. Débugger

III. Coder

J'ai choisi mon langage, adopté une méthode de développement... Ça y est, je peux écrire du code?

Sur une feuille? Dans un bloc-notes?

Il n'y aurait pas des outils pour coder plus efficacement?

Comment savoir ce qui existe et comment l'utiliser?

Et si je veux documenter pour les autres?

III. Coder

Les éditeurs de texte

Vi, Emacs, Geany,

Notepad ++ (Windows)

The image displays three text editors side-by-side. On the left is Emacs, showing its logo and a terminal window with the command `/tmp/vi.FuxNv18297: new file: line 1`. In the middle is Geany, showing a PHP file named `hello.p` with a class definition for `helloworld`. On the right is Notepad++, showing a PHP file named `Article.php` with a dropdown menu open over the `interface_exists` function, listing options like `intval`, `in_array`, `ip2long`, and `iptcembed`. The Notepad++ status bar at the bottom indicates `PHP Hy length: 139209 lines: 4673 Ln: 2595 Col: 26 Sel: 0`.



III. Coder

Les éditeurs de texte

Comparaison selon les fonctionnalités propres à la programmation

Programming features [\[edit \]](#)

	◊ Syntax highlighting ◊	Function list ◊	Symbol database (ctags or equiv.) ◊	Brace matching ◊	Auto indentation ◊	Auto completion ◊	Code folding ◊	Text folding ◊	Compiler integration ◊
Acme	No	Yes	Yes	Yes	Yes	Yes	No	No	Yes
AkelPad	Plug-in	Plug-in	Plug-in	Plug-in	Yes	Plug-in	Plug-in	No	Plug-in
Alphatk	Yes	?	?	Yes	Yes	Yes	Yes	Yes	Yes
Atom	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Plug-in

<https://en.wikipedia.org/wiki/>

[Comparison of text editors#Programming features](https://en.wikipedia.org/wiki/Comparison_of_text_editors#Programming_features)

Les environnements de développement intégré (IDE)

Application fournissant

- Éditeur de code (auto-complétion du code, surlignage syntaxique)
- Compilateur et/ou interpréteur
- Debugger
- Gestion de version
- Outils de création d'IHM
- Pour les développements orientés objet :
 - Class browser, class inspector, diagramme hiérarchique de classe
- Indépendance par rapport à l'OS
- Pour plein de langages :
 - Java, C, C++, PHP, Perl, Ruby, HTML, Python, Latex... Fortran

III. Coder

Les environnements de développement

Atom (libre, multi plate-forme)

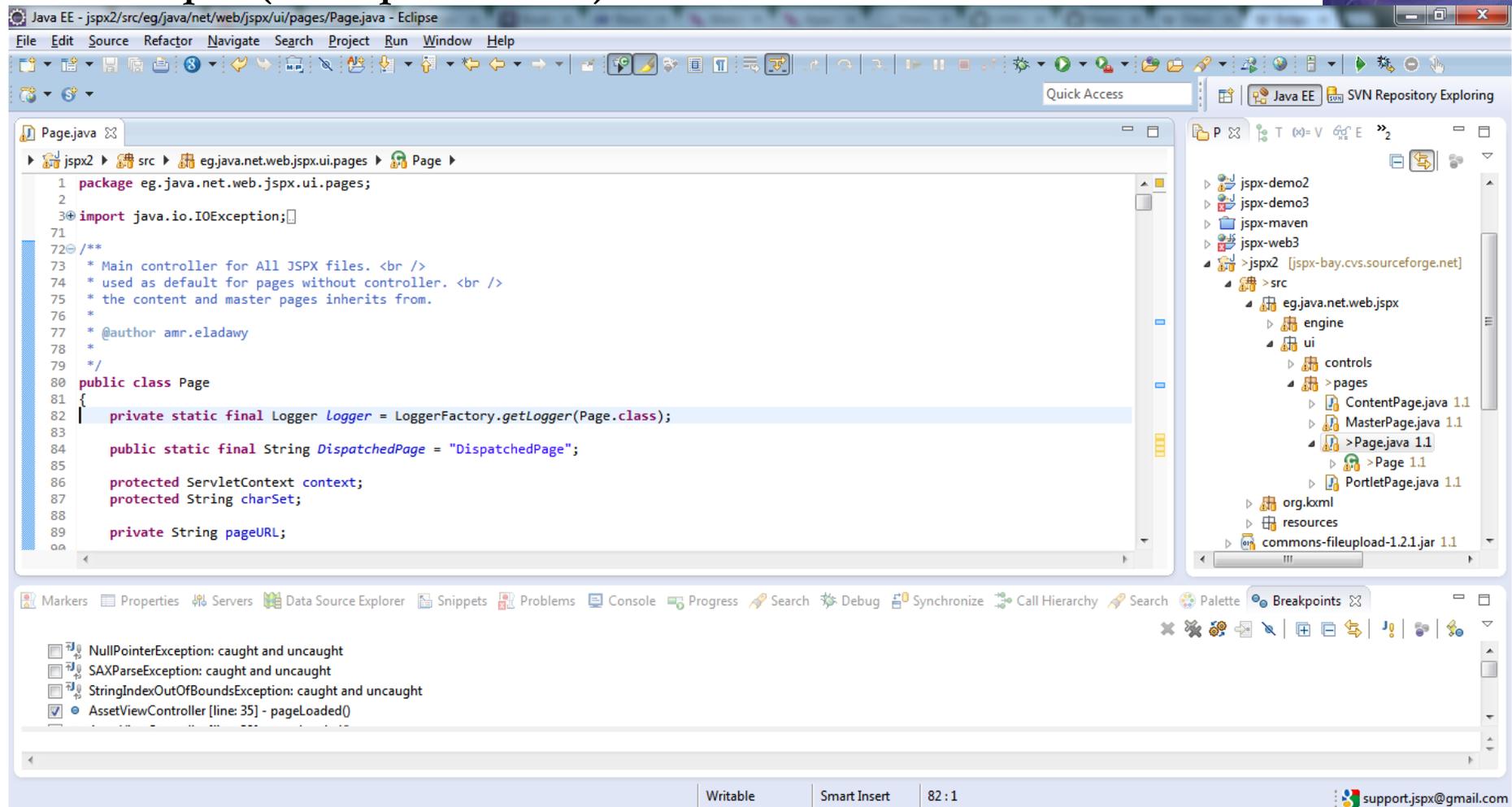
A screenshot of the Atom text editor interface. The left sidebar shows a file explorer with a project named "atom" containing various folders like ".git", ".github", "apm", "benchmarks", "docs", "dot-atom", "electron", "exports", "keymaps", "menus", "node_modules", "out", "resources", "script", "spec", and "src". The main editor area shows a JavaScript file named "text-editor-element.js" with the following code:

```
272  
273  
274   getComponent () {  
275     if (!this.component) {  
276       this.component = new TextEditorComponent({  
277         element: this,  
278         mini: this.hasAttribute('mini'),  
279         updatedSynchronously: this.updatedSynchronously  
280       })  
281       this.updateModelFromAttributes()  
282     }  
283     return this.component  
284   }  
285 }  
286  
287 module.exports =  
288 document.registerElement('atom-text-editor', {  
289   prototype: TextEditorElement.prototype  
290 })  
291
```

The status bar at the bottom shows "src/text-editor-element.js", "Babel", "master", and "1 file".

III. Coder

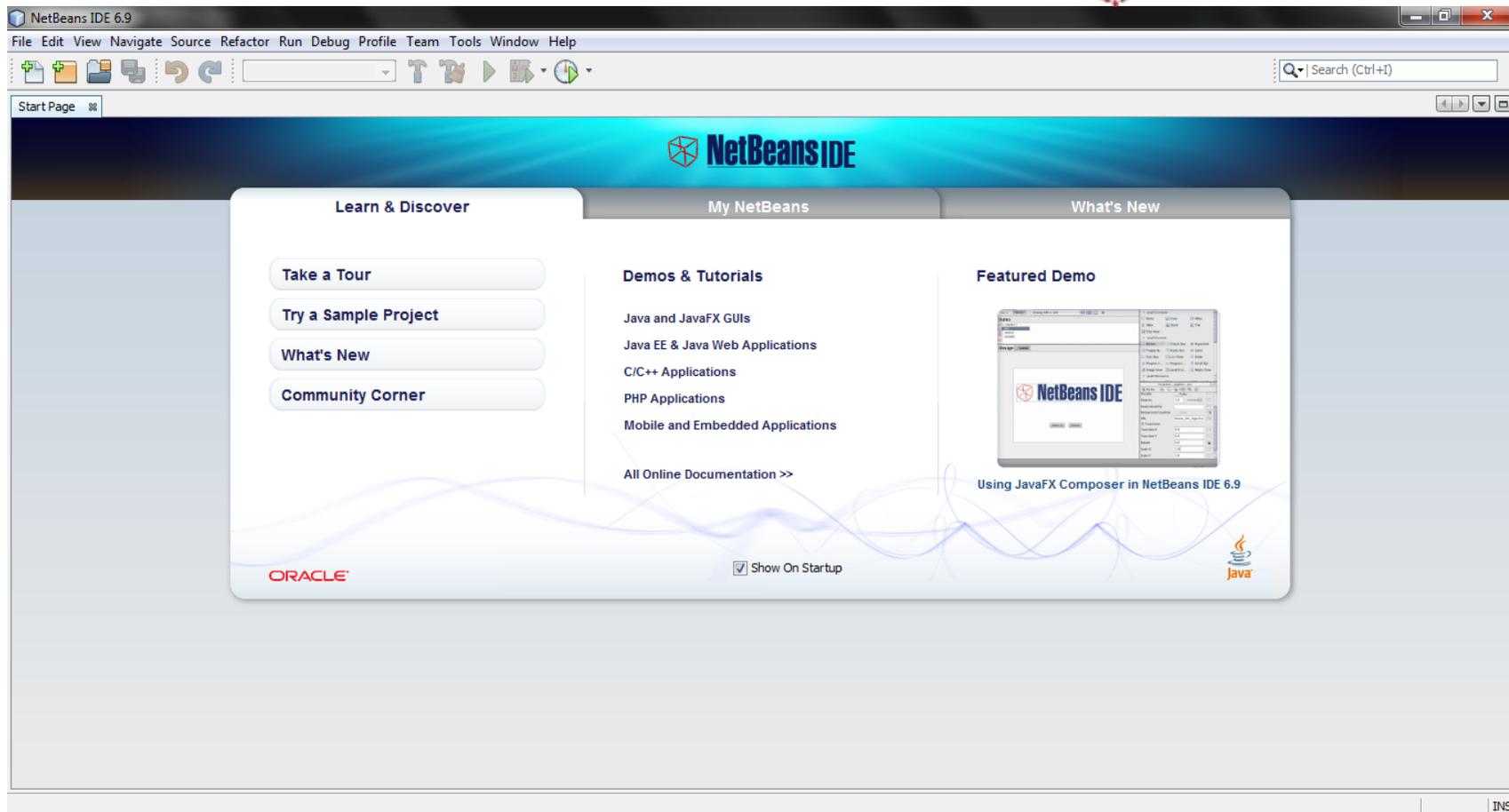
Les environnements de développement Eclipse (multi plate-forme)



III. Coder

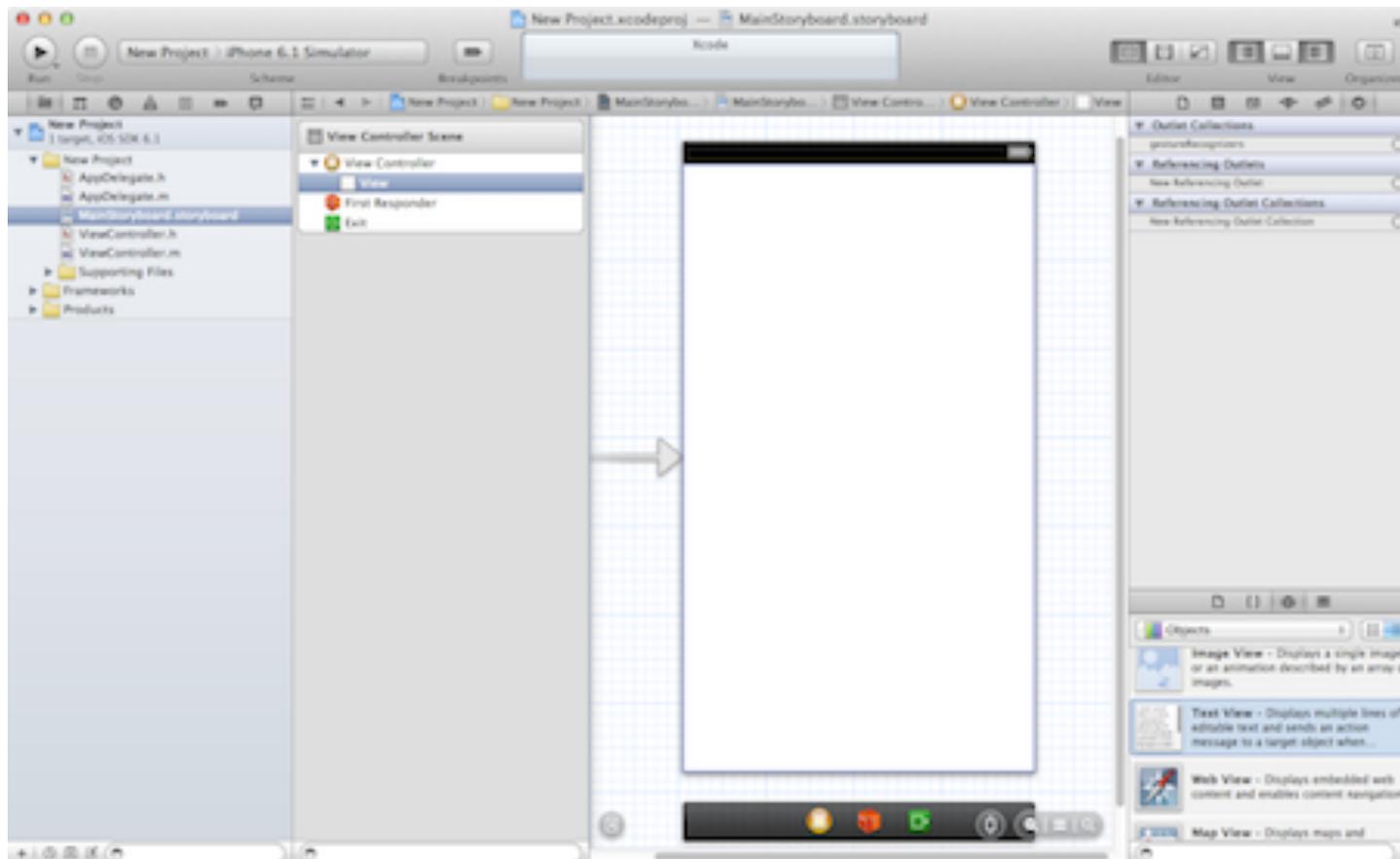
Les environnements de développement

Netbeans (Windows, Linux)



III. Coder

Les environnements de développement Xcode (Mac OS X)



III. Coder

Les environnements de développement

Visual Studio (Windows)



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Diagnostics;

namespace Form1
{
    public class Form1
    {
        private void InitializeComponent()
        {
            this.InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            MessageBox.Show("Have you tried turning it off and on again?", "Restart", MessageBoxButtons.YesNo);

            do While 1 = false
            {
                if (MessageBox.Show("Have you booted it?", "Single 1?", MessageBoxButtons.YesNo) = DialogResult.No)
                {
                    MessageBox.Show("Single 1? again?");
                }
                if (MessageBox.Show("Is it fixed?", "Fixed?", MessageBoxButtons.YesNo) = DialogResult.No)
                {
                    L = true;
                    MessageBox.Show("Congratulations, you are now a certified computer technician!");
                }
            }
            else
            {
                MessageBox.Show("Single 2?");
            }
        }
    }
}

namespace Form1
{
    public class Program
    {
        [STAThread]
        static void Main()
        {
            Application.EnableDefaultButtons();
            Application.Run(new Form1());
        }
    }
}
```

III. Coder

Les environnements de développement Comparaison par langage

C/C++ [edit]

Main articles: *C (programming language)* and *C++*

IDE	License	Written in	Windows	Linux	macOS	Other platforms	Debugger	GUI builder	Integrated toolchain	Profiler	Code coverage	Autocomplete	Static code analysis	GUI-based design	Class browser	Latest stable release	C compiler	C++ compiler	Refactoring
Anjuta	GPL	C	No	Yes	No	FreeBSD	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes	2016-03	Yes	Yes	No
AppCode (IntelliJ IDEA)	Proprietary	Java	No	No	Yes	No	Yes	Yes	No	Yes (Xcode profiler)	No	Yes	Yes	Yes	Yes	2012-12	Yes (Xcode toolchain)	Yes (Xcode toolchain)	Yes
C++Builder	Proprietary, Freeware (Starter edition only)	C++ and Object Pascal	Yes	No (Cross compiler planned)	Yes (Cross compiler)	cross-compiles for Android and iOS	Yes	Yes	Yes	Yes (AQTime Standard in package manager)	Yes	Yes	Yes	Yes	Yes	2017-03 Tokyo 10.2	Yes	Yes	Yes
Code::Blocks	GPL	C++	Yes	Yes	Yes	FreeBSD, OpenBSD, Solaris	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes ^[5]	Yes	2016-01 ^[6]	Yes (MinGW + custom)	Yes (MinGW + custom)	Yes
CodeLite	GPL	C++	Yes	Yes	Yes	FreeBSD	Yes	Yes	Yes	Yes (As of CodeLite 6.1, integration with Valgrind)	No	Yes	Yes	Yes ^[7]	Yes	2016-07 ^[8]	Yes (GCC, Clang, VC + custom)	Yes (GCC, Clang, VC + custom)	Yes
Dev-C++	GPL	Object Pascal	Yes	No ^[9]	No	FreeBSD	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	2013-10 (Unofficial Version), 2005-02 (Official Version)	Yes	Yes	No
Eclipse CDT	EPL	Java	Yes	Yes	Yes	FreeBSD, JVM, Solaris	Yes	Yes ^[2]	Yes ^[10]	Yes ^[11]	Yes ^[12]	Yes	Yes	Yes	Yes	2016-07 ^[13]	External	External	Yes
Geany	GPL	C	Yes	Yes	Yes	FreeBSD, AIX, OpenBSD, Solaris, other Unix	Yes (via a plug-in)	No	No	No	No	Yes	No	No	Yes	2016-03 ^[14]	External	External	No

<https://en.wikipedia.org/wiki/>

[Comparison of integrated development environments#C/C++](#)

La documentation

Diverses vocations :

- Pour les développeurs comme moi (on peut générer une documentation de référence automatiquement!)
- Pour l'utilisateur final (site web avec tutoriels)
- Pour les chercheurs (articles au contenu scientifique)

III. Coder

La documentation pour l'utilisateur final

Site Web

Guide de l'utilisateur

Contenu scientifique

Verdandi, data assimilation library

<http://verdandi.sourceforge.net/index.php>

Verdandi user's guide

<http://verdandi.sourceforge.net/doc-1.5/index.php>

http://verdandi.sourceforge.net/doc-1.5/reduced_order_unscen...

Verdandi generic library for data assimila

INTRODUCTION CONTENTS DOCUMENTATION DOWNL

Verdandi is a generic C++ library for data assimilation.

Verdandi is currently developed at **INRIA**. It aims at providing methods and tools for data assimilation large class of problems involving high-dimensional numerical models.

To guarantee the highest performance, the library is implemented in C++. In addition, Verdandi provides **Swig**.

Models implemented in Fortran, C, C++, Python, ... can be plugged to Verdandi using either a C++ or C

Verdandi is provided under the **GNU Lesser General Public License (LGPL)**.

Scientific Context

Data assimilation is the process of combining different sources of information in order to better estimate extension, some parameters can also be estimated. These methods were originally introduced to deal with pertaining mostly to geophysics, but it is now widely recognized that they have a tremendous potential in eHeart example below).

Whether the system be biological, environmental, mechanical, etc., the main sources of information are observations and error statistics. Data assimilation methods can be written independently of the system method can be applied to a wide class of systems. Therefore methods are generic and can be put together

What is Verdandi for?

What is Verdandi designed for?

- to provide data assimilation methods to non-specialists;
- to facilitate the application of methods to a great number of problems;
- to provide a framework for perennial development;
- to improve the diffusion and impact of data assimilation algorithms.

Who can be a Verdandi user?

- non-specialists, engineers or researchers, who could directly use the available data assimilation r
- a specialist taking advantage of a modular framework, which should ease development, transfers

The users provide the numerical model and the observations with the appropriate interface.

Acknowledgment



The development of Verdandi is financially supported by the European research at developing, sharing and integrating patient-specific multi-physics and multi-scale models in normal and pathological conditions to address clinical challenges. Data assimilation is thus intended to allow the personalization of the biophysical

1 sur 2

29/08/13 23:21

Verdandi generic library for data assimila

INTRODUCTION CONTENTS DOCUMENTATION DOWNL
PREV. VERSIONS VERSION 1.3 VERSION 1.4 VERSIO

This documentation is also available for [download in PDF format](#).

USER'S GUIDE

Introduction

Getting Started

Dependencies

Assimilation Methods

Example Models

Observations

Tools

Plugging in Verdandi

Debugging

Python Interface

API REFERENCE

Classes

Class List

Class Hierarchy

Class Members

Functions

Search for

Support

Verdandi User's G

Verdandi is a generic C++ library for data assimilation.

Verdandi is currently developed at **INRIA**. It aims at providing methods and tools for data assimilation. It is designed to be relevant to a large class of problem numerical models.

To guarantee the highest performance, the library is implemented in C++. In addition, Verdandi provides **Swig**.

Models implemented in Fortran, C, C++, Python, ... can be plugged to Verdandi using either a C++ or C

Verdandi is provided under the **GNU Lesser General Public License (LGPL)**.

Scientific context

Data assimilation is the process of combining different sources of information in order to better estimate the state of a system. By extension, some parameters can also be estimated. These methods were originally introduced to deal with uncertainties present in many geophysics, but it is now widely recognized that they have a tremendous potential in eHeart example below).

Whether the system be biological, environmental, mechanical, etc., the main sources of information are observations and error statistics. Data assimilation methods can be written independently of the system to which they are applied, and can be applied to a wide class of systems. Therefore methods are generic and can be put together

What is Verdandi for

What is Verdandi designed for?

- to provide data assimilation methods to non-specialists;
- to facilitate the application of methods to a great number of problems;
- to provide a framework for perennial development;
- to improve the diffusion and impact of data assimilation algorithms.

Who can be a Verdandi user?

- non-specialists, engineers or researchers, who could directly use the available data assimilation methods;
- a specialist taking advantage of a modular framework, which should ease development, transfers and interactions.

Verdandi user's guide

Functions

Search for

Support

3. Optionally initialize a step with `InitializeStep()`. This
4. Perform a step forward and propagate the state error variance
5. Compute the analysis with `Analyze()`, whenever observations
6. Compute the data assimilation until the model has finished: if the simulation is done, false otherwise.

Reduced Order Unscented Kalman

Assuming that P is of reduced rank p – typically much smaller than the basic idea in reduced-order filtering is, in essence, to be able to manipulate P in reduced-order form

$$P = LU^{-1}L^T,$$

where U – in the group of invertible matrices \mathcal{GL}_p – is of much smaller rank than P and represents the main uncertainties in the system. What is crucial here is that computations on L and U without needing to compute P .

Simplex case

In this section, we focus on the simplex distribution. Consider some $(V^{(i)})_{1 \leq i \leq r} \in \mathbb{R}^p$ associated with some coefficients $(\alpha) = (\alpha_i)_{1 \leq i \leq r}$ matrix of these sigma-points denoted by $[V^*] \in \mathcal{M}_{r,p}$ and the mean $D_\alpha = \text{diag}(\alpha_1, \dots, \alpha_r) \in \mathcal{M}_r$.

1. Sampling:
 - $C_h = \sqrt{U_h^{-1}}$,
 - $x_h^{(i)a} = x_h^a + L_h C_h I^{(i)}$, $1 \leq i \leq p+1$
2. Prediction:
 - $x_{h+1}^f = E_\alpha(\mathcal{M}_h(x_h^{(i)a}))$
 - $x_{h+1}^{(i)f} = \begin{cases} x_{h+1}^f + [\mathcal{M}_h(x_h^a)] D_\alpha [V^*]^T ([V^*]^T I \\ \text{or} \\ \mathcal{M}_h(x_h^{(i)a}) \end{cases}$
 - $L_{h+1} = [x_{h+1}^{(i)f}] D_\alpha [V^*]^T \in \mathcal{M}_{n,p}$
 - $P_{h+1}^f = L_{h+1} (P_h^\alpha)^{-1} L_{h+1}^T$
3. Update:
 - $y_{h+1}^{(i)} = \mathcal{H}_{h+1}(x_{h+1}^{(i)f})$
 - $\{HL\}_{h+1} = [y_{h+1}^{(i)}] D_\alpha [V^*]^T$
 - $U_{h+1} = P_{h+1}^f + \{HL\}_{h+1}^T R_{h+1}^{-1} \{HL\}_{h+1} \in \mathcal{M}_{n,p}$
 - $x_{h+1}^a = x_{h+1}^f + U_{h+1}^{-1} \{HL\}_{h+1}^T R_{h+1}^{-1} (y_{h+1} - \mathcal{H}_{h+1}(x_{h+1}^f))$
 - $P_{h+1}^a = L_{h+1} U_{h+1}^{-1} L_{h+1}^T$

With:

x_h^f forecast state vector;

x_h^a analysis state vector;

29/08/13 23:28

<http://verdandi.sourceforge.net>

La documentation par et pour les développeurs

Le code lui-même : choix de noms explicites!

Les commentaires – concis et pertinents pour se repérer dans le code

Les diagrammes UML

La documentation de référence automatique : Doxygen, Javadoc

La documentation de référence automatique

Pourquoi générer automatiquement la documentation ?

- La seule source d'information absolument juste est le code
- La rédaction de la documentation technique est laborieuse et complexe
- Faciliter la maintenance, le développement d'un code écrit seul et surtout à plusieurs
- La documentation est écrite dans le code, et il est donc relativement facile de la tenir à jour

La documentation de référence automatique

Comment ça marche ?

- Extraction de l'information à partir du code source et d'autres données laissées à la responsabilité du développeur
- Tient compte de la syntaxe et de la structure du langage du programme ainsi que des commentaires associés

La documentation de référence automatique



The screenshot shows the homepage of the Doxygen website. At the top, the logo 'Doxygen' is displayed in a blue banner, with the text 'latest release v1.5.7.1 - last page update 19 Dec 2008' to its right. Below the banner, the main heading is 'Source code documentation generator tool'. A paragraph describes Doxygen as a documentation system for various languages like C++, C, Java, Objective-C, Python, IDL, Fortran, VHDL, PHP, C#, and to some extent D. It lists three ways it can help: generating an on-line documentation browser or off-line manual, configuring it to extract code structure from undocumented files, and creating normal documentation. A 'Doxygen license' section is also visible, along with a copyright notice for 1997-2008 by Dimitri van Heesch. On the right side, there are several navigation menus: 'Home' (Manual, Mailing Lists, Mail archive, FAQ, ChangeLog, Todo / Wish List, Report bugs, Doxygen users, Articles, Other Doc Tools), 'SVN' (SVN tarballs, Download, Download manual, Helper tools, RPM Packages, Debian Packages), and 'Help doxygen!' (Donate, Contributors, Preisvergleich, Afvallen). Below these is a 'Javadoc Tool' section with 'Downloads', 'Reference', 'Community', and 'Learning' sub-sections.

Javadoc is a tool for generating API documentation in HTML format from doc comments in source code. It can be downloaded only as part of the Java 2 SDK. To see documentation generated by the Javadoc tool, go to J2SE 1.5.0 API Documentation.

- **Javadoc FAQ** - This FAQ covers where to download the Javadoc tool, how to find a list of known bugs and feature requests, workarounds for known bugs, how to increase memory for Javadoc, and more.
- **Javadoc Documentation** - Enhancements, Standard Doclet, Doclet overview, Doclet and Taglet APIs -- See Javadoc 1.5, Javadoc 1.4, Javadoc 1.3, or Javadoc 1.2.
- **Javadoc Reference Pages** - See Javadoc 1.5, Javadoc 1.4, Javadoc 1.3, or Javadoc 1.2 for options and examples for calling the Javadoc tool.
- **How to Write Doc Comments for Javadoc** - Sun conventions for writing documentation comments.
- **Requirements for Writing API Specifications** - Standard requirements used when writing the Java 2 Platform Specification. Covers requirements for packages, classes, interfaces, fields and methods to satisfy testable assertions.

Doclets

The standard doclet generates HTML and is built into the Javadoc tool. Other doclets that Java Software has developed are listed here.

- **Doclet API** is an API provided by the Javadoc tool for use by doclets. See [Doclet Overview](#) for a basic description and simple examples. (These documents are for version 1.3 of Java 2 SDK, Standard Edition.)
- **Taglet API** is an interface provided for custom formatting the text of Javadoc tags. See [Taglet Overview](#) for a basic description and simple examples. (These documents are for version 1.5 of Java 2 SDK, Standard Edition.)
- **MIF Doclet** - Want beautiful PDF? This doclet can automate the generation of API documentation in PDF by way of MIF. It also enables you to print directly to a printer. MIF is Adobe FrameMaker's interchange format.
- **DocCheck Doclet** checks doc comments in source files and generates a report listing the errors and irregularities it finds. It is part of the Sun Doc Check Utilities.
- **Exclude Doclet** is a simple wrapper program that enables you to exclude from the generated documentation any public or protected classes (or packages) that you specify. It takes a list of classes in a file and removes them from the RootDoc before delegating execution to the standard doclet.
- **Doclet Toolkit** is an API and implementation for doclets emulating the standard doclet. We plan to release this toolkit hopefully during 2004. More information will be available here and through the Javadoc announce email listed below.

III. Coder

Génération via le code

```

//! A test class.
/*!
 * A more elaborate class description.
 */

class Test
{
public:

    /*! An enum.
     *! More detailed enum description. */
    enum TEnum {
        TVal1, /*!< Enum value TVal1. */
        TVal2, /*!< Enum value TVal2. */
        TVal3 /*!< Enum value TVal3. */
    }

    /*! Enum pointer.
     *! Details. */
    *enumPtr,
    /*! Enum variable.
     *! Details. */
    enumVar;

    /*! A constructor.
     *!
     * A more elaborate description of the constructor.
     */
    Test();

    /*! A destructor.
     *!
     * A more elaborate description of the destructor.
     */
    ~Test();

    /*! A normal member taking two arguments and returning an integer value.
     *!
     * \param a an integer argument.
     * \param s a constant character pointer.
     * \return The best results
     * \sa Test(), ~Test(), testMeToo() and publicVar()
     */
    int testMe(int a, const char *s);

    /*! A pure virtual member.
     *!
     * \sa testMe()
     * \param c1 the first argument.
     * \param c2 the second argument.
     */
    virtual void testMeToo(char c1, char c2) = 0;

    /*! A public variable.
     *!
     * Details.
     */
    int publicVar;

    /*! A function variable.
     *!
     * Details.
     */
    int (*handler)(int a, int b);
};
    
```

Utilisation



Main Page Classes

Class List Class Members

Test Class Reference

A test class. More...

List of all members:

Public Types

enum **TEnum** { **TVal1**, **TVal2**, **TVal3** }
An enum. More...

Public Member Functions

Test ()
A constructor.

~Test ()
A destructor.

int **testMe** (int a, const char *s)
A normal member taking two arguments and returning an integer value.

virtual void **testMeToo** (char c1, char c2)=0
A pure virtual member.

Public Attributes

Bio++ Class documentation

Utils NumCalc SeqLib PhyLib PopGenLib

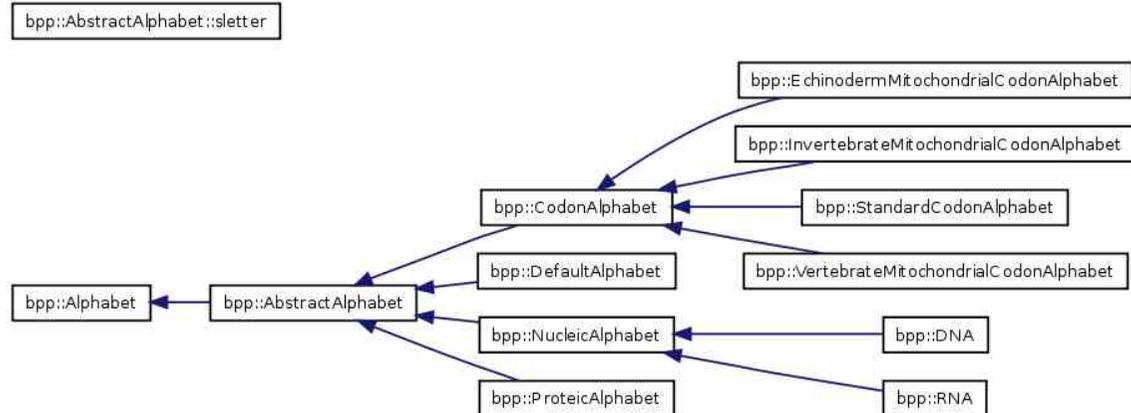
Main Page Classes Files Related Pages

Class List Class Hierarchy Class Members

Search for

SeqLib Graphical Class Hierarchy

[Go to the textual class hierarchy](#)



A more elaborate description of the constructor:

IV. Partager, travailler en équipe

I. Choisir un langage (Vincent)

II. Adopter une méthode de développement

III. Coder

IV. Partager, travailler en équipe

V. Profiling

VI. Débugger

IV. Partager, travailler en équipe

Ça compile, ça tourne, et le résultat est correct!

Des collègues veulent utiliser mon programme, certains veulent même contribuer en développant...

Comment leur donner la main?

Et du point de vue légal?

IV. Partager, travailler en équipe

Licences

Deux aspects :

- Que puis-je faire avec un programme selon sa licence ?
- Dois-je utiliser une licence pour mes programmes ?
Si oui, pourquoi?
Et laquelle ?

IV. Partager, travailler en équipe

Licences

- Le logiciel est couvert par le droit d'auteur
- Le logiciel ne se limite pas au code source, mais comprend aussi documents d'analyse fonctionnelle, de conception technique, maquette, prototype, aide en ligne...
- Quelle est la licence d'un logiciel? Elle doit être spécifiée dans les sources, et dans un fichier (par exemple LICENSE.TXT)
- Pourquoi utiliser une licence?
 - Sans licence, personne ne peut utiliser un programme.
 - Contrôler l'utilisation et la diffusion du programme.

IV. Partager, travailler en équipe

Licences

Qui détient le copyright du logiciel ?

- **Sur commande** : Réglé dans le contrat entre le commanditaire et le prestataire.
- **Salarié** : Droits d'exploitation transmis de plein droit à l'employeur
Le laboratoire n'a pas les droits. Seul le CNRS ou l'université ont les droits.
- **Temps libre** : Droits d'exploitation et droits moraux reviennent à l'auteur.
- **Stagiaires** : Voir la convention de stage...

IV. Partager, travailler en équipe

Licences : Recommandations

Se rapprocher des services de valorisation des tutelles

-> Enregistrer les auteurs

Solliciter l'Agence de Protection des Programmes
(protection par un système de dépôt et de référencement)

IV. Partager, travailler en équipe

Licences de logiciel libre

Qu'est-ce qu'une licence de logiciels libres ?

- La liberté d'exécuter le logiciel
- La liberté d'étudier le fonctionnement du logiciel
- La liberté de redistribuer des copies du logiciel
- La liberté d'améliorer le logiciel et de publier ses améliorations

IV. Partager, travailler en équipe

Licences de logiciel libre : Typologie

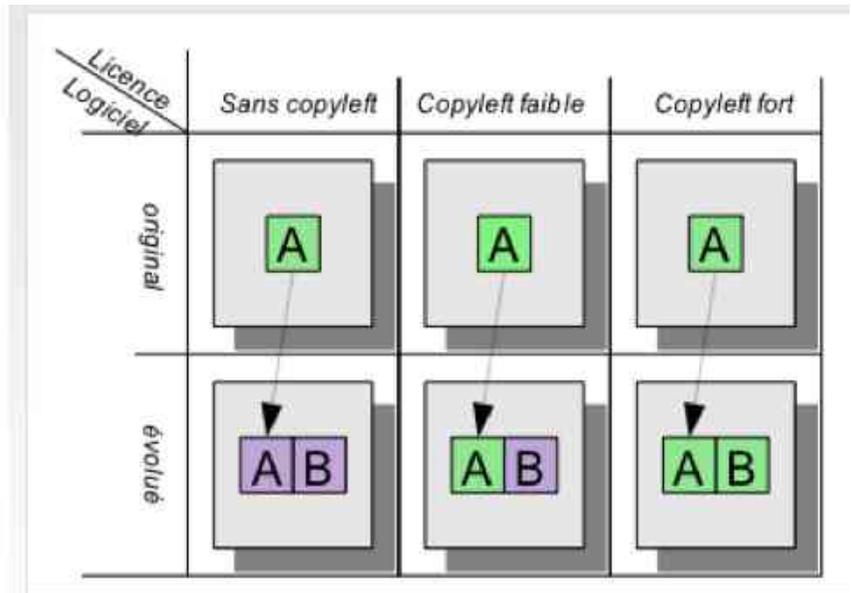
Déterminer le caractère copyleft ou non de la licence

Copyleft :

- Exigence de réciprocité : le code source (le logiciel) sera redistribué sous la même licence qu'il a été reçu
- Héritaire

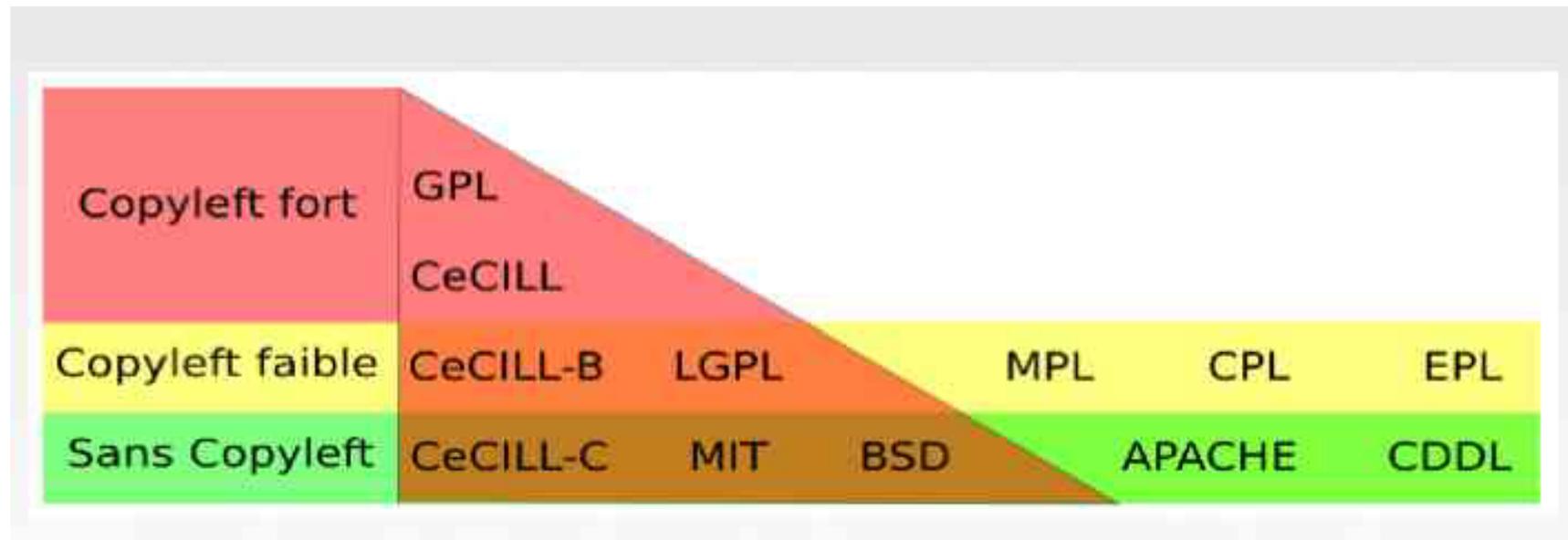
Portée du copyleft

Copyleft faible ou fort



IV. Partager, travailler en équipe

Licences de logiciel libre : Compatibilité



IV. Partager, travailler en équipe

Licences de logiciel libre : Pourquoi?

- Outils idéaux pour la préservation du patrimoine intellectuel d'un laboratoire de recherche
- Aucune renonciation à un usage ultérieur du logiciel
- Coût nul : l'ajout des mentions de licences dans chaque fichier source suffit pour bénéficier des termes de la licence
- Le dépôt des organismes de type APP (pour un coût un coût dérisoire) apporte une preuve d'antériorité
- Mutualisation de l'effort de développement
- Maximisation des possibilités d'irriguer la société

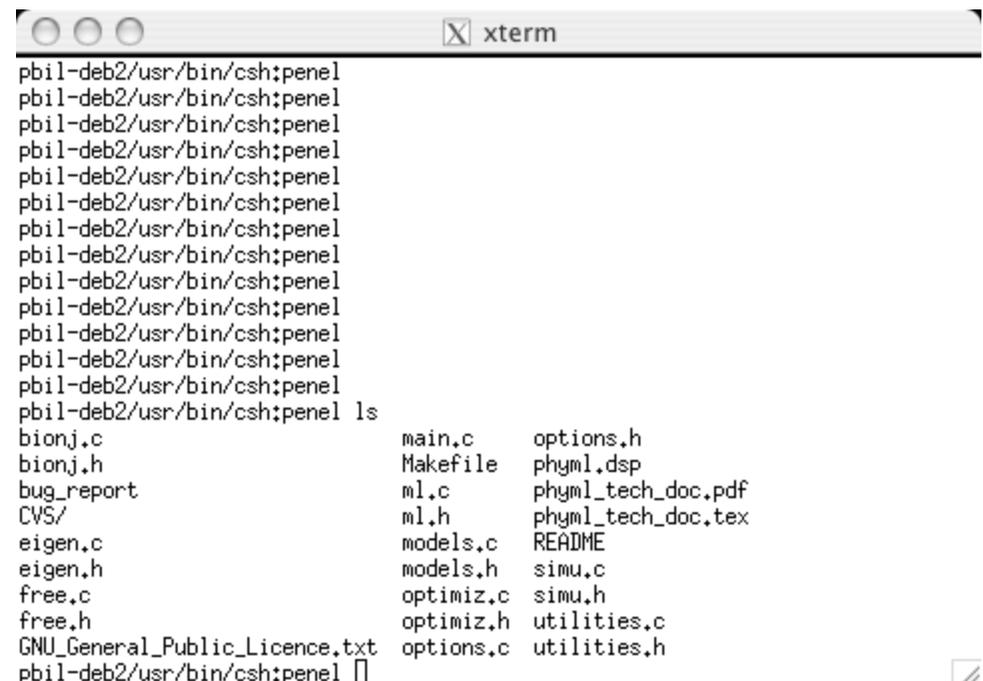
IV. Partager, travailler en équipe

Licences de logiciel libre : Comment?

Mention des auteurs et de la licence dans l'en-tête des fichiers

Licence présente dans l'arborescence

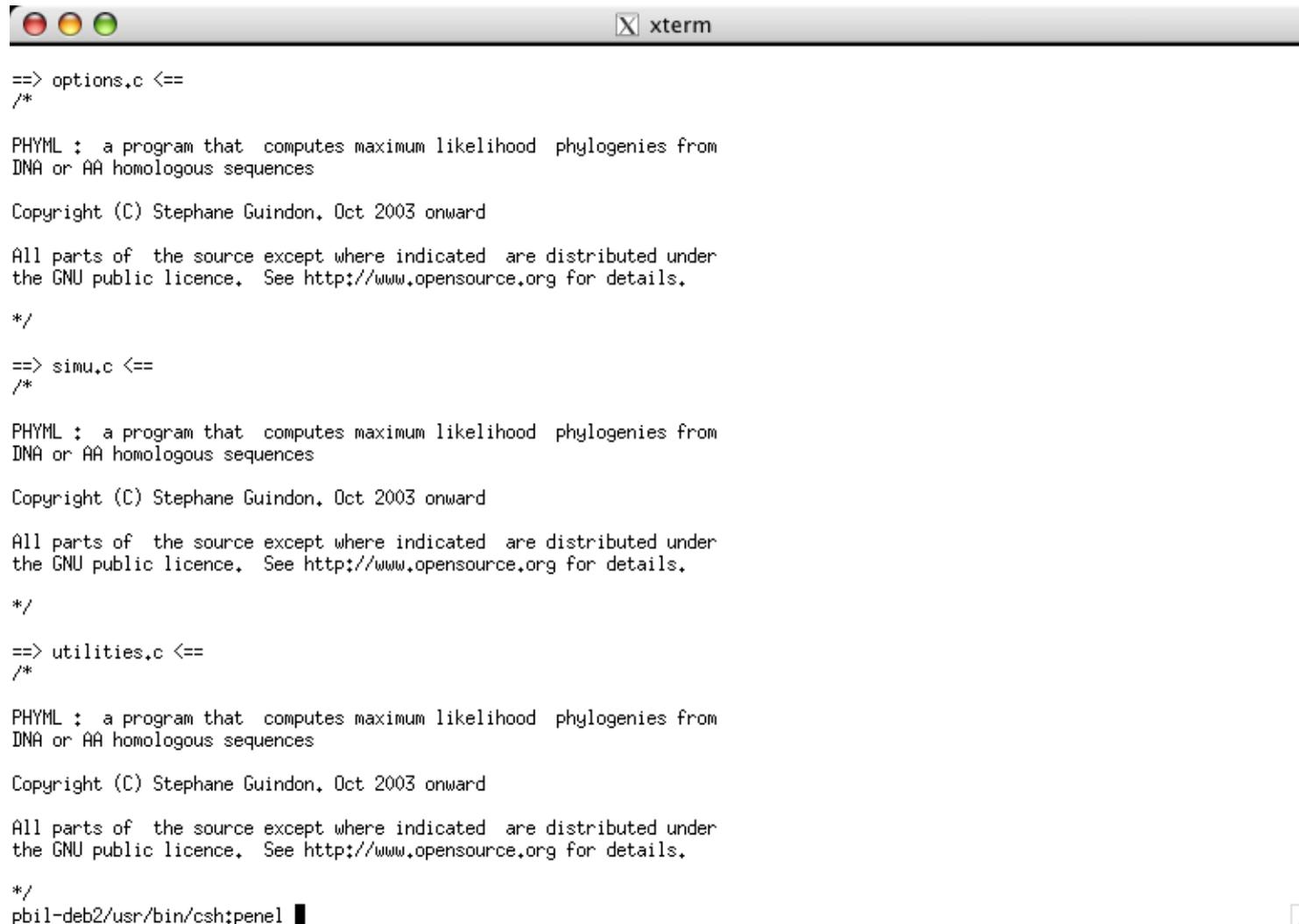
Exemple : le programme phyml



```
pbil-deb2/usr/bin/csh:penel
pbil-deb2/usr/bin/csh:penel ls
bionj.c          main.c          options.h
bionj.h          Makefile       phyml.dsp
bug_report      ml.c           phyml_tech_doc.pdf
CVS/            ml.h           phyml_tech_doc.tex
eigen.c         models.c       README
eigen.h         models.h       simu.c
free.c          optimiz.c      simu.h
free.h          optimiz.h      utilities.c
GNU_General_Public_Licence.txt options.c      utilities.h
pbil-deb2/usr/bin/csh:penel
```

IV. Partager, travailler en équipe

Licences de logiciel libre : Comment?



```
xterm

==> options.c <==
/*

PHYML : a program that computes maximum likelihood phylogenies from
DNA or AA homologous sequences

Copyright (C) Stephane Guindon, Oct 2003 onward

All parts of the source except where indicated are distributed under
the GNU public licence. See http://www.opensource.org for details.

*/

==> simu.c <==
/*

PHYML : a program that computes maximum likelihood phylogenies from
DNA or AA homologous sequences

Copyright (C) Stephane Guindon, Oct 2003 onward

All parts of the source except where indicated are distributed under
the GNU public licence. See http://www.opensource.org for details.

*/

==> utilities.c <==
/*

PHYML : a program that computes maximum likelihood phylogenies from
DNA or AA homologous sequences

Copyright (C) Stephane Guindon, Oct 2003 onward

All parts of the source except where indicated are distributed under
the GNU public licence. See http://www.opensource.org for details.

*/
pbil-deb2/usr/bin/csh:penel █
```

IV. Partager, travailler en équipe

Partager : Archive des sources

Donner accès au répertoire compressé des sources
(format tar.gz, tar.bz2, zip)

→ Permet à l'utilisateur de :

- Mieux comprendre qui se cache derrière les appels aux fonctionnalités
- Mieux les utiliser
- Eventuellement contribuer en tant que développeur
- Marche pour toutes les plateformes

IV. Partager, travailler en équipe

Partager : Packaging

Problématique :

- Installation facile : éviter à chaque utilisateur l'étape de compilation, longue (à cause des éventuelles dépendances) et source potentielle d'erreurs
- Portabilité : défi du multi-plateforme (Windows, Mac OS, Linux)

Exemples de packages :

- RPM sous Fedora
- make dist
- CPack multi-plateforme
- Distutils (Python)
- R packaging

IV. Partager, travailler en équipe

CMake

Système de construction logicielle
multi-plateforme

Génération des Makefile,
à la base du mécanisme de compilation
(gestion de plusieurs fichiers source,
édition de lien avec éventuelles dépendances,
via une commande simple)



```
cmake_minimum_required(VERSION 2.6)

#Déclaration du projet
project(MyProject)

#Génération de la liste des fichiers sources
file(
    GLOB_RECURSE
    source_files
    src/*
)

#Déclaration de l'exécutable
add_executable(
    my_exe
    ${source_files}
)
```

IV. Partager, travailler en équipe

Travailler en équipe : Gestion de version

Problématique

- Ça marche plus... je voudrais annuler ma modification!
- On est plusieurs à modifier le même document... quelle version est la dernière?

IV. Partager, travailler en équipe

Travailler en équipe : Gestion de version

Qu'est-ce que c'est ?

Activité permettant de gérer les modifications d'un ensemble de données.

Typiquement : code source d'un logiciel

Également applicable à d'autres catégories de données :

- Documentation (par exemple Latex)
- Site web
- Fichiers de configuration d'un système

IV. Partager, travailler en équipe

Travailler en équipe : Logiciels de gestion de version

Permet de :

- Travailler à plusieurs
- Garder une trace des différents stades de développement, éventuellement de revenir en arrière
- Suivre l'évolution du programme
- Développer des versions différentes du programme en parallèle

IV. Partager, travailler en équipe

Travailler en équipe : Logiciels de gestion de version

Fonctions de base :

- Un dépôt (projet + historique), des copies « locales » du projet
- Conserve un historique des modifications
- Permet de travailler à plusieurs:
fusion, gestion des conflits
- Permet les modifications en parallèle

IV. Partager, travailler en équipe

Travailler en équipe : Gestion de version

Modèle client-serveur :

Programmes :

- CVS 
- Subversion



IV. Partager, travailler en équipe

Travailler en équipe : Gestion de version

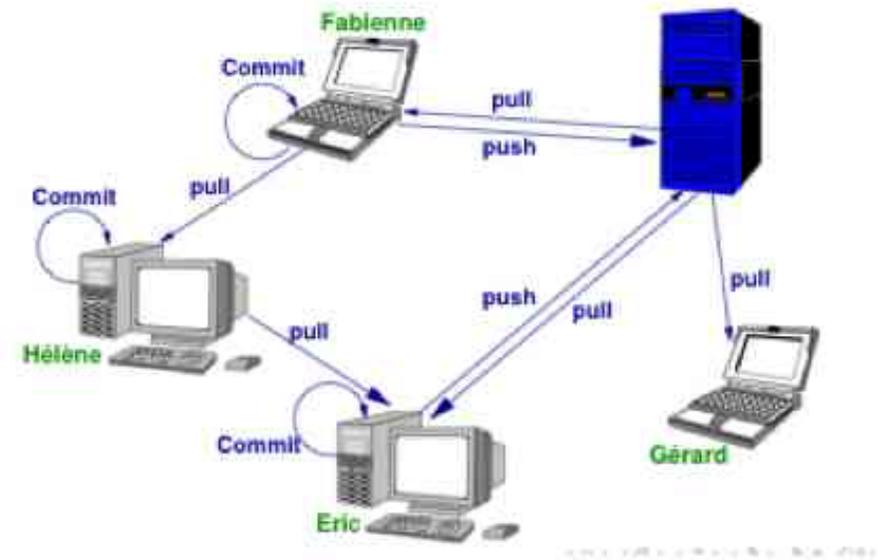
Systeme distribue :

Programmes :

- Git



- Bitkeeper
- Mercurial



Pas de dépôt centralisé

- Chaque développeur a sa copie avec ses branches privées
- Plus de liberté
- Opération de synchronisation des dépôts des développeurs

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Définition :

Une Forge a plusieurs facettes :

- Un portail communautaire
- Un outil de gestion de projets
- Un environnement de développement collaboratif
- Un site pour une communauté

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Contenu d'une Forge : services aux projets

Une Forge offre un ensemble d'outils permettant la gestion des projets logiciels

- Gestion des sources : CVS / SVN / Git
- Trackers : feature requests, bug tracker, tâches
- Livraisons (fichiers, packages)
- Gestion des documents, Wiki
- Autres services aux projets
 - Forums
 - Mailing lists
 - Sondages, news
 - Administration : gestion des membres, des services associés

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

- **Services en ligne d'hébergement de projets**
 - [GNU Savannah](#) (Free Software Foundation, hébergement de logiciel libre)
 - [GitHub](#) (réseau social, compte gratuit pour projet de logiciel libre)
- **Logiciels intégrés pour le déploiement d'une forge**

Déployer sa propre plate-forme pour gagner en indépendance moyennant un coût de mise en place et de maintenance

 - Uniquement Git : [GitLab](#) (alternative libre à GitHub, intégration continue)
 - Multi-logiciel de versionning :
 - [FusionForge](#) (reprise du code sous GPL de Gforge, descendant de SourceForge)
 - [Redmine](#) (gestion de projet complète, pas de gestion des tests)

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Exemples :

- GitHub

<https://github.com/SimonRit/RTK/>

- GitLab

<https://gitlab.in2p3.fr/explore/projects>

<https://forge.p2chpd.univ-lyon1.fr/explore/projects>

<https://gitlab.inria.fr/explore/projects>

- FusionForge

<https://sourcesup.renater.fr/> (accessible à tous les acteurs de l'enseignement supérieur et de la recherche)

- Redmine

<http://vip.creatis.insa-lyon.fr:9002/projects>

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Exemple : Projet RTK hébergé sur GitHub

SimonRit / RTK

3,832 commits 3 branches 7 releases 21 contributors Apache-2.0

Branch: master New pull request Create new file Upload files Find file Clone or download

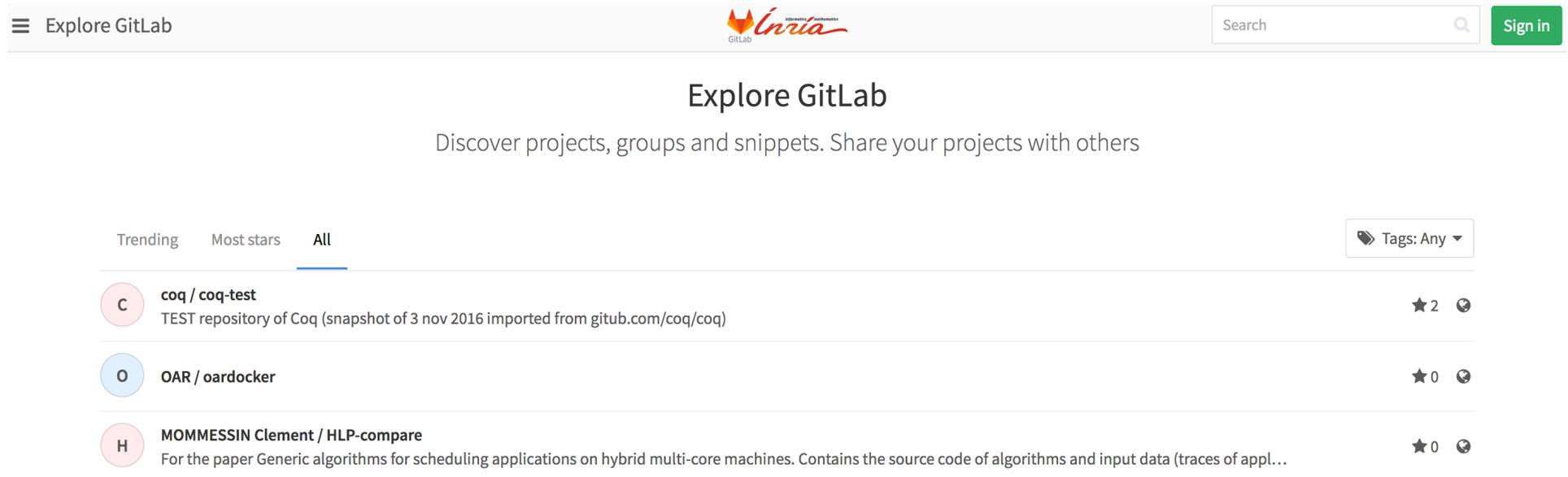
Commit	Description	Time
Simon Rit	Fix initialization of new scale and origin parameters	9 hours ago
applications	Fix mandory options for rtkdrawgeometricphantom	15 hours ago
cmake	COMP: Conform header include guard names to ITK style	3 months ago
code	Fix initialization of new scale and origin parameters	9 hours ago
documentation	Remove CMake-language block-end command arguments	4 months ago
examples	Convert CMake-language commands to lower case	4 months ago
testing	Fixed bug in rtkdecomposedualenergyprojections. Increased precision	11 days ago
utilities	COMP: Fixing compilation of SimpleRTK with BUILD_SHARED_LIBS enabled	15 days ago
.travis.yml	Do not run any test on travis if compiled with CUDA, only CUDA	6 months ago
CMakeLists.txt	Release v1.3	2 months ago

<https://github.com/SimonRit/RTK>

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Exemple : GitLab INRIA



The screenshot shows the GitLab INRIA explore page. At the top, there is a navigation bar with the GitLab logo, the INRIA logo, a search bar, and a 'Sign in' button. Below the navigation bar, the page title 'Explore GitLab' is centered, followed by the subtitle 'Discover projects, groups and snippets. Share your projects with others'. The main content area features a list of projects under the 'All' tab. The first project is 'coq / coq-test' with 2 stars. The second is 'OAR / oardocker' with 0 stars. The third is 'MOMMESSIN Clement / HLP-compare' with 0 stars. A 'Tags: Any' dropdown menu is visible in the top right of the project list area.

Explore GitLab

Discover projects, groups and snippets. Share your projects with others

Trending Most stars All Tags: Any

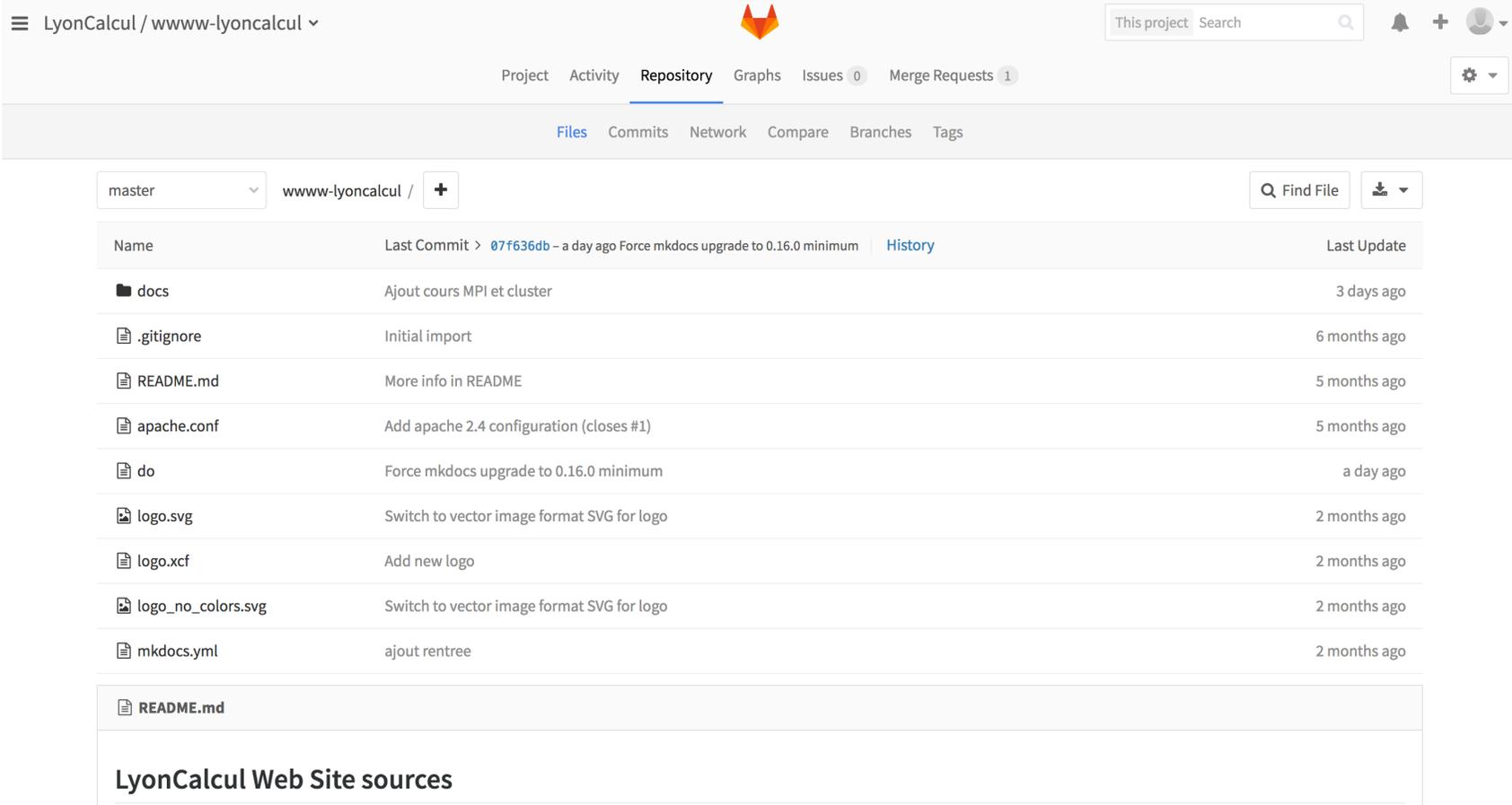
- C** coq / coq-test
TEST repository of Coq (snapshot of 3 nov 2016 imported from github.com/coq/coq) ★ 2
- O** OAR / oardocker ★ 0
- H** MOMMESSIN Clement / HLP-compare
For the paper Generic algorithms for scheduling applications on hybrid multi-core machines. Contains the source code of algorithms and input data (traces of appl...)

<https://gitlab.inria.fr/explore/projects/>

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Exemple : GitLab Mésocentre Lyon – Projet Lyon Calcul



The screenshot shows the GitLab interface for the repository 'LyonCalcul / www-lyoncalcul'. The 'Repository' tab is active, displaying a file list for the 'master' branch. The table below summarizes the files shown in the screenshot.

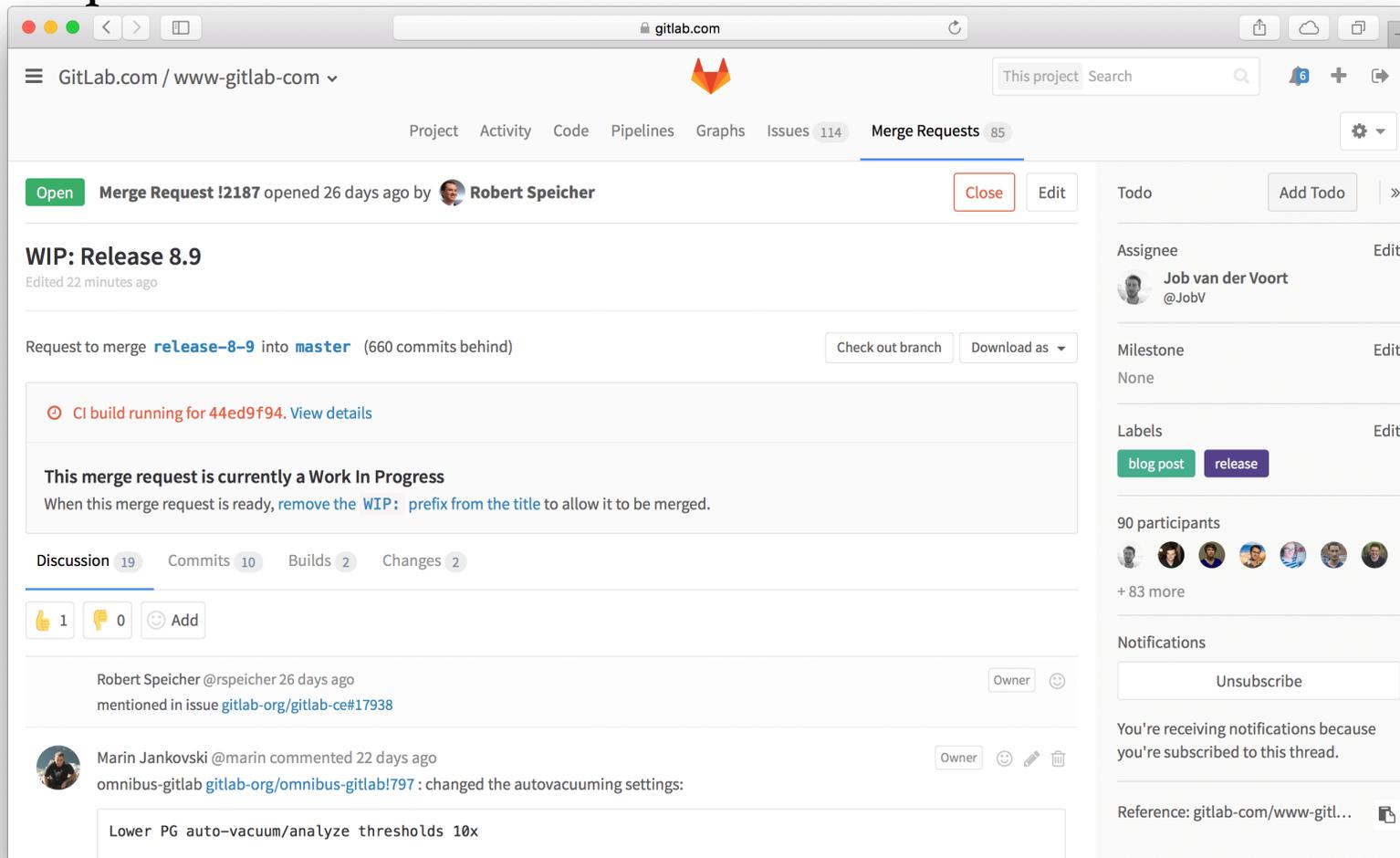
Name	Last Commit	Last Update
docs	Ajout cours MPI et cluster	3 days ago
.gitignore	Initial import	6 months ago
README.md	More info in README	5 months ago
apache.conf	Add apache 2.4 configuration (closes #1)	5 months ago
do	Force mkdocs upgrade to 0.16.0 minimum	a day ago
logo.svg	Switch to vector image format SVG for logo	2 months ago
logo.xcf	Add new logo	2 months ago
logo_no_colors.svg	Switch to vector image format SVG for logo	2 months ago
mkdocs.yml	ajout rentree	2 months ago

<https://forge.p2chpd.univ-lyon1.fr/LyonCalcul/www-lyoncalcul/tree/master>

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Exemple : GitLab



<https://about.gitlab.com/features/>

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Exemple : GitLab – Résolution de conflit de merge

The screenshot shows a GitLab Merge Request for the project 'GitLab.org / GitLab Community Edition'. The merge request is titled 'Touch project when toggling stars to update cache' and was opened by Eliza Burr. The description states: 'This is a fix for gitlab-org/gitlab-ce#3513.' The interface shows a conflict in the file 'app/models/application_setting.rb'. The conflict is between the 'HEAD//our changes' and 'origin//their changes'. The 'our changes' version has two lines for 'CACHE_KEY': 'application_setting.last' and 'application_settings.last'. The 'their changes' version has two lines for 'CACHE_KEY': 'applications_setting.last' and 'applications_setting.last'. The interface provides buttons to 'Use ours' or 'Use theirs' for each conflicting section. The file content is as follows:

```
... @@ -30,6 +30,8 @@
30 30 #
31 31
32 32 class ApplicationSetting < ActiveRecord::Base
    HEAD//our changes Use ours
33     CACHE_KEY = 'application_setting.last'
34     CACHE_KEY = 'application_settings.last'
    origin//their changes Use theirs
35 35     serialize :restricted_visibility_levels
36 36     serialize :import_sources
37 37     serialize :restricted_signup_domains, Array
... @@ -703,6 +703,7 @@ class User < ActiveRecord::Base
700 700     else
701 701     UsersStartProject.create!(project: project, user: self)
702 702     end
    HEAD//our changes Use ours
703     project.touch #update project cache
```

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Exemple : GitLab – Gestion des tâches / bugs (« Issues »)

GitLab.org / GitLab Community Edition

Project Activity Repository Pipelines Graphs Issues 5,224 Merge Requests 399 Snippets

Open 5,157 Closed 9,331 All 14,488

Filter by name ... New Issue

Author Assignee Milestone Labels Weight Last created

- Make hover/focus/active states consistent for dropdowns**
#24150 · opened 14 minutes ago by Chris Peressini 8.14 UX updated 12 minutes ago
- Better hover/focus states for navigation bar buttons**
#24149 · opened 20 minutes ago by Chris Peressini 8.14 UX updated 12 minutes ago
- Record img height when uploading so issue page doesn't jump on loading**
#24148 · opened 21 minutes ago by Régis Freyd (GitLab) Frontend updated 21 minutes ago
- Delete environments**
#24147 · opened 39 minutes ago by Dave updated 39 minutes ago
- Add blue border to buttons on focus**
#24146 · opened 41 minutes ago by Chris Peressini 8.14 Frontend updated 12 minutes ago
- Make hover/focus/active states more noticeable for buttons**
#24145 · opened about an hour ago by Chris Peressini 8.14 UX updated 12 minutes ago
- Make sure all buttons turn dark on focus**
#24144 · opened about 2 hours ago by Chris Peressini 8.14 Frontend UX updated 12 minutes ago
- Shared CI secure variables across the projects**
#24143 · opened about 2 hours ago by Vasily Bezruchkin updated about 2 hours ago

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Exemple : GitLab – Intégration continue

The screenshot shows the GitLab interface for a project. At the top, there is a navigation bar with the GitLab logo and a search bar. Below the navigation bar, there are tabs for Project, Activity, Repository, Pipelines, Graphs, Issues (5,225), Merge Requests (398), and Snippets. The Pipelines tab is selected, and there are sub-tabs for Pipelines, Environments, and Cycle Analytics.

Pipeline #4860322 with 46 builds for `show-status-from-branch` (queued for 2 minutes 55 seconds)

running

Authored by **Lin Jen-Shin (godfat)** 17 minutes ago

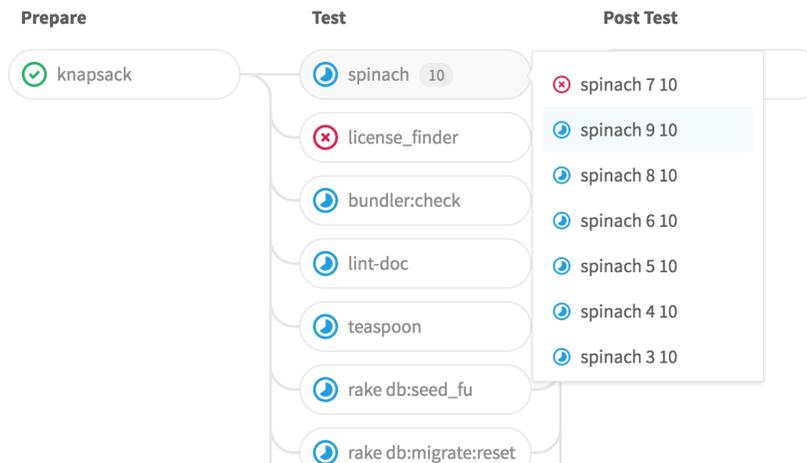
Commit `ce1dc4c25d3464b7a9a1b21d93157c9fed98f705`

Update for CHANGELOG

Hide pipeline graph

Retry failed

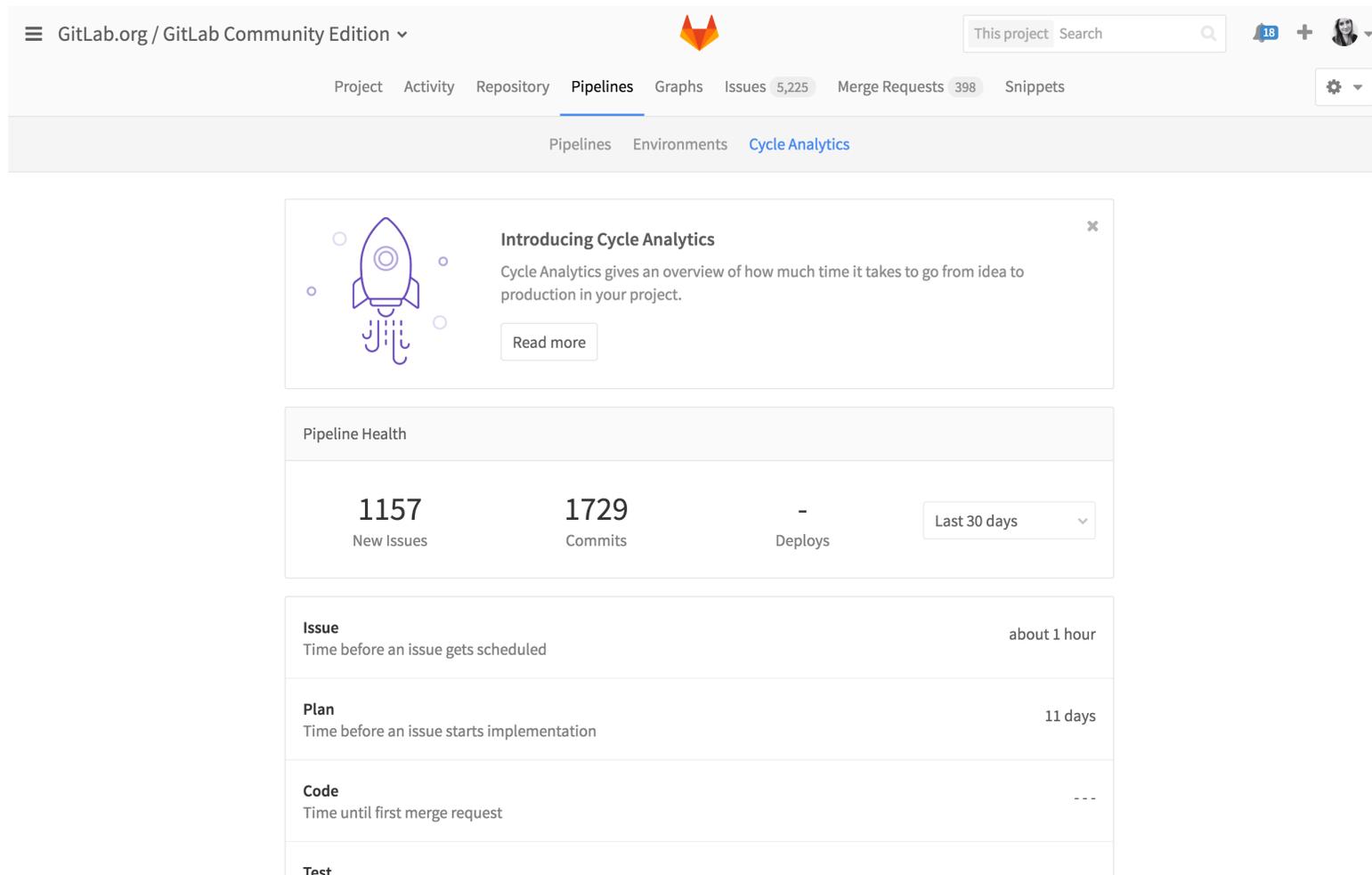
Cancel running



IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Exemple : GitLab – Analyse d'un cycle de développement



GitLab.org / GitLab Community Edition

This project Search

Project Activity Repository Pipelines Graphs Issues 5,225 Merge Requests 398 Snippets

Pipelines Environments Cycle Analytics

Introducing Cycle Analytics

Cycle Analytics gives an overview of how much time it takes to go from idea to production in your project.

[Read more](#)

Pipeline Health

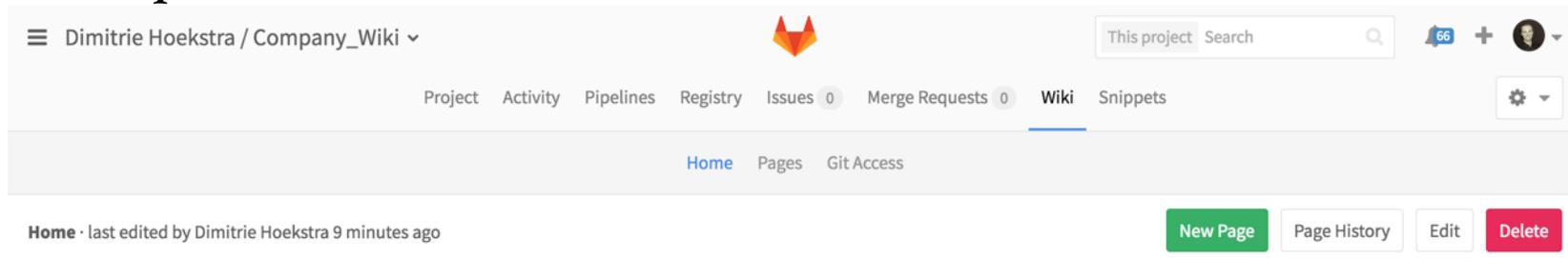
1157 New Issues	1729 Commits	- Deploys	Last 30 days
--------------------	-----------------	--------------	--------------

Issue Time before an issue gets scheduled	about 1 hour
Plan Time before an issue starts implementation	11 days
Code Time until first merge request	---
Test	

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Exemple : GitLab – Wiki



The screenshot shows the GitLab Wiki interface for a project named 'Company_Wiki'. The breadcrumb trail is 'Dimitrie Hoekstra / Company_Wiki'. The navigation menu includes 'Project', 'Activity', 'Pipelines', 'Registry', 'Issues 0', 'Merge Requests 0', 'Wiki', and 'Snippets'. The 'Wiki' tab is active, showing a 'Home' page. The page title is 'Home' and it indicates it was last edited by Dimitrie Hoekstra 9 minutes ago. Action buttons for 'New Page', 'Page History', 'Edit', and 'Delete' are visible.

Wiki: Project / Company Values

We value results, transparency, sharing, freedom, efficiency, frugality, collaboration, directness, kindness, diversity, boring solutions, and quirkiness:

- Results:** We care about what you achieve; the code you shipped, the user you made happy, and the team member you helped. Do not compete by proclaiming how many hours you worked yesterday because we don't want someone who took the afternoon off to feel like they did something wrong. Instead, celebrate the achievements of yourself and your teammates. We want people to have the desire to ship.
- Transparency:** Be open about as many things as possible. By making information public we can reduce the threshold to contribution and make collaboration easier. An example is the [public repository of this website](#) that also contains this [company handbook](#). Everything we do is public by default, for example, the [GitLab CE](#) and [GitLab EE](#) issue trackers, but also [marketing](#) and [infrastructure](#). Transparency creates awareness for GitLab, allows us to recruit people that care about our culture, it gets us more and faster feedback from people outside the company, and makes it easier collaborate with them. There are exceptions, material that is not public by default is documented in the [general guidelines](#). On a personal level, you should tell it like it is instead of putting up a poker face. Don't be afraid to admit you made a mistake or were wrong. When something went wrong it is a great opportunity to say "What's the [kaizen](#) moment here?" and find a better way without hurt feelings.
- Sharing:** We care about giving great software, documentation, examples, lessons, and processes to the world. An example is the MIT licensed [GitLab CE](#). We believe that open source creates more value than it captures. We are grateful to our customers, users, partners, investors, and the open source ecosystem.
- Freedom:** You should have clear objectives and the freedom to work on them as you see fit. Any instructions are open to discussion. You don't have to defend how you spend your day. We trust team members to do the right thing instead of having rigid rules.
- Efficiency:** We care about working on the right things, not doing more than needed, and not duplicating work. This enables us to achieve more progress with fewer people and makes our work more fulfilling. We think of how we can make the company better instead of being territorial or defensive.
- Frugality:** [Amazon states it best](#) with: *Accomplish more with less. Constraints breed resourcefulness, self-sufficiency and invention. There are no extra points for growing headcount, budget size or fixed expense.*
- Collaboration:** Helping others is a priority, even when it is not related to the goals that you are trying to achieve. You are expected to ask others for help and advice. Anyone can chime in on any subject, including people who don't work at GitLab. The person who has to do the work decides how to do it but you should always take the suggestions seriously and try to respond and explain.

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Exemple : FusionForge SourceSup par Renater

SourceSup
par RENATER

La forge Enseignement supérieur et Recherche

Vous êtes déjà authentifié. [Continuer.](#)

Projets Rechercher

Accueil Ma page Projets Échantillons de code Postes ouverts du projet

Bienvenue sur SourceSup

SourceSup est un service opéré par [RENATER](#)

SourceSup est une plateforme web de gestion de projets à destination de l'Enseignement Supérieur et des laboratoires de Recherche Français. Tous les membres de la communauté peuvent créer un projet sur SourceSup. Les Projets sont, par défaut, privés, mais ils peuvent être rendus publics. De nombreux outils sont disponibles et peuvent être activés pour chaque projet dans l'onglet administration.

- [Documentation](#) est disponible ici
- [Adresse de support](#) pour envoyer un mail à notre support.

Dernières annonces

Release of FullSWOF_UI 1.02.00
Christian Laguerre - 13/05/2016 13:43 - [FullSWOF_UI](#)
The FullSWOF development team is pleased to announce the release of FullSWOF_UI 1.02.00.
1 Commentaire [Lire la suite/Commenter](#)

Release of FullSWOF_2D 1.07.00
Christian Laguerre - 14/03/2016 21:17 - [FullSWOF_2D](#)
The FullSWOF development team is pleased to announce
1 Commentaire [Lire la suite/Commenter](#)

Nouvelle version 3.0 Commandant Poulard !!!
Gérard Milhaud - 17/02/2016 08:01 - [QoQ-CoT](#)
Pré-scriptum : désolé pour l'hypercentex pas très hyper, mais pas possible de faire figurer des liens dans les annonces de SourceSup, même si on peut les rédiger avec fckEditor... :-(
1 Commentaire [Lire la suite/Commenter](#)

Nuage de mots-clés
C++ ESUP Java MATLAB PHP Perl Python R XML XSLT alignment bioinformatics
calendrier data analysis database django esup formations java javascript mass
spectrometry monitoring pedagogie protein proteomics python quooxdoo read
security signal processing simulation web

Statistiques de SourceSup

Projets hébergés : **2,226**
Utilisateurs inscrits : **6,702**

Projets les plus téléchargés

183,008	AGATTE
130,780	DataJiver
81,095	ScientificPython
41,967	eXtensible Metagrammar
41,168	BibliOpera
30,580	Lodel
20,370	Molecular Modelling Toolkit
15,810	Développement de WIMS

<https://sourcesup.renater.fr/>

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Exemple : FusionForge SourceSup par Renater – Projet AGATTE

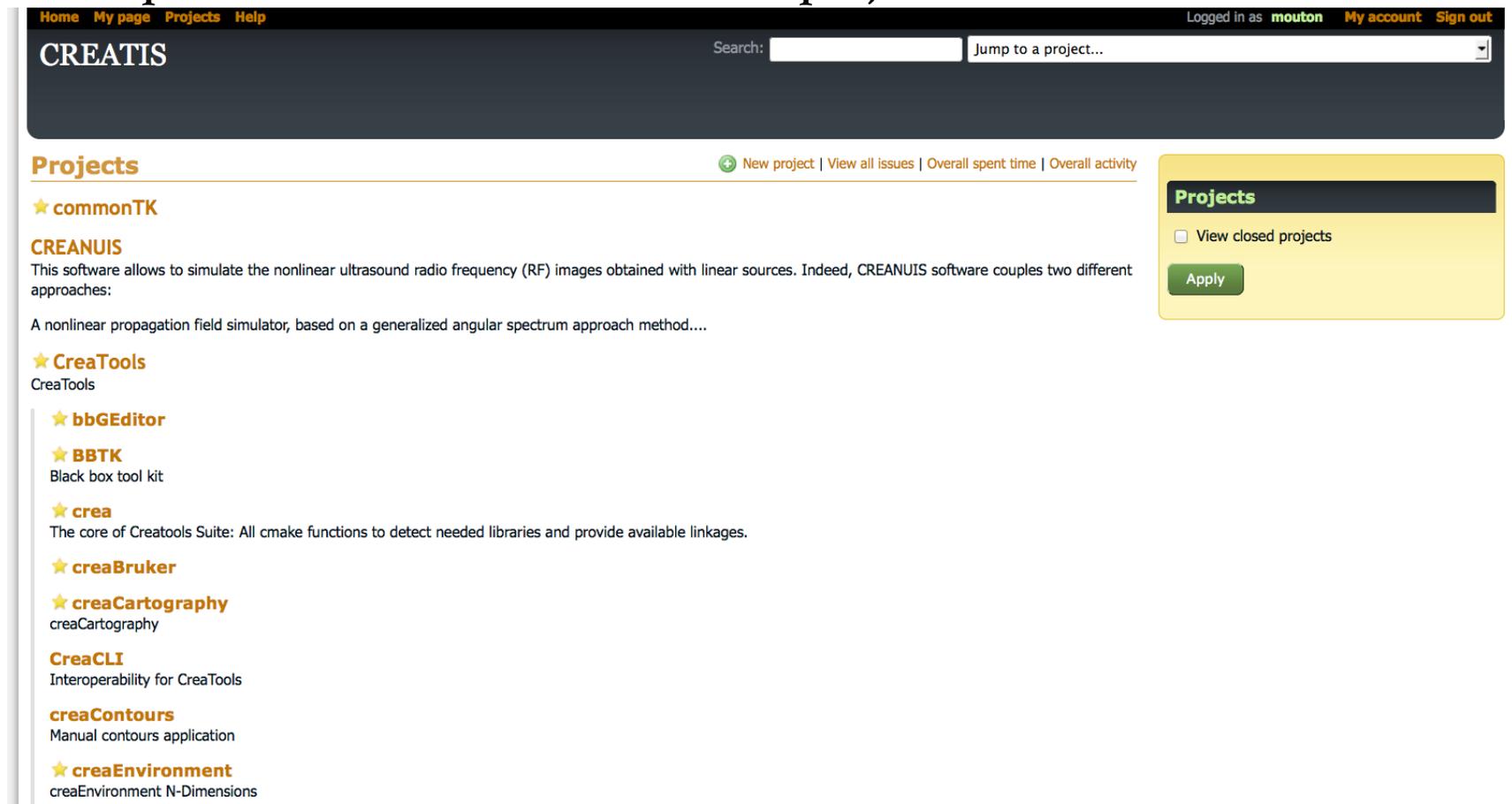
The screenshot displays the SourceSup project page for 'AGATTE'. The header includes the SourceSup logo (par RENATER) and the text 'La forge Enseignement supérieur et Recherche'. A search bar is present with the text 'Chercher dans le projet entier' and a 'Rechercher' button. The main navigation bar includes 'Accueil', 'Ma page', 'Projets', 'Échantillons de code', 'Postes ouverts du projet', and 'AGATTE'. Below this, there are sub-navigation tabs: 'Résumé', 'Activité', 'Outil de suivi', 'Tâches', 'Documents', 'Annonces', 'Sources', 'Fichiers', and 'Listes Sympa'. The 'Description du projet' section states: 'Logiciel WEB de gestion du Temps de travail à l'Université : gestion des pointages et des congés'. The 'Information du projet' section lists various metadata: Development Status (5 - Production/Stable), Environment (Web Environment, Win32 (MS Windows)), Intended Audience (End Users/Desktop), License (Other/Proprietary License), Natural Language (French), Operating System (Microsoft: Windows, POSIX: Linux), Programming Language (Delphi/Kylix, Java, PL/SQL), and Topic (Office/Business: Scheduling). The 'Membres du projet' section lists administrators (Kevin Hergalant, Olivier ZILLER, Cédric Champmartin) and members (Jéréme GUTIERREZ, Matthieu Manginot, Matthieu Bildstein), with a link to 'Voir les 6 membre(s)'. The 'Dernières annonces' section shows three recent announcements by Olivier ZILLER, each with a 'Lire la suite/Commenter' link.

<https://sourcesup.renater.fr/projects/agatte/>

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Exemple : Redmine à CREATIS – Les projets



The screenshot displays the CREATIS Redmine interface. At the top, there is a navigation bar with links for Home, My page, Projects, and Help. The user is logged in as 'mouton' and can access My account or Sign out. A search bar and a 'Jump to a project...' dropdown are also present. The main content area is titled 'Projects' and includes links for 'New project', 'View all issues', 'Overall spent time', and 'Overall activity'. A list of projects is shown, including 'commonTK', 'CREANUIS', and 'CreaTools'. The 'CREANUIS' project is expanded, showing a description: 'This software allows to simulate the nonlinear ultrasound radio frequency (RF) images obtained with linear sources. Indeed, CREANUIS software couples two different approaches: A nonlinear propagation field simulator, based on a generalized angular spectrum approach method....'. Below this, a list of sub-projects is displayed, each with a star icon and a brief description: 'bbGEditor', 'BBTK' (Black box tool kit), 'crea' (The core of Creatools Suite: All cmake functions to detect needed libraries and provide available linkages.), 'creaBruker', 'creaCartography', 'CreaCLI' (Interoperability for CreaTools), 'creaContours' (Manual contours application), and 'creaEnvironment' (creaEnvironment N-Dimensions). On the right side, a sidebar titled 'Projects' contains a checkbox for 'View closed projects' and an 'Apply' button.

<http://vip.creatis.insa-lyon.fr:9002/projects>

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Exemple : Redmine à CREATIS – Le projet OsiriX Plug-in >> inTag

The screenshot displays the Redmine interface for the 'inTag' project. The top navigation bar includes 'Home', 'My page', 'Projects', and 'Help'. The user is logged in as 'mouton'. The project title 'OsiriX Plug-in » inTag' is prominently displayed. Below the title is a search bar and a navigation menu with tabs for Overview, Activity, Roadmap, Issues, New issue, Gantt, Calendar, News, Documents, Wiki, Files, Repository, and Settings. The 'Overview' tab is active, showing a description of the inTag software, a list of team members with their roles, and an issue tracking summary. The 'Spent time' section shows 0.00 hours. The 'Manager' section lists team members with their profile pictures and roles. A 'Download' button is visible at the bottom right of the page.

Home My page Projects Help

Logged in as **mouton** My account Sign out

OsiriX Plug-in » inTag

Search: » inTag

Overview Activity Roadmap Issues New issue Gantt Calendar News Documents Wiki Files Repository Settings

Overview [New subproject](#)

inTag plug-in for OsiriX, is a software to calculate, display and analyze myocardial strains and intra-myocardial mechanics from cardiac MR images with a tagging pattern.

inTag brings processing and analysis of cardiac tagged MR sequences to most in the clinical environment. One of the main reasons for MR tagging not to be more frequently used in preclinical of clinical research, is the lack of widely available processing tools.

inTag offers a fast and integrated process that bring quantitative strain analysis to most scientists, or physicians in a matter of minutes.

Further information inTag plug-in wiki

- Homepage: <http://www.creatis.insa-lyon.fr/intag/>

Issue tracking

- Bug: 2 open / 2
- Feature: 4 open / 4
- Support: 0 open / 0
- Research: 0 open / 0
- Integration: 0 open / 0
- Deployment: 0 open / 0
- Test: 2 open / 2

[View all issues](#) | [Calendar](#) | [Gantt](#)

Members

Manager: Claire Mouton, DAVILA Eduardo, Maciej Orkisz, Patrick Clarysse, pierre croisille, William A. Romero R.

Developer: Claire Mouton, DAVILA Eduardo, Patrick Clarysse, pierre croisille, William A. Romero R.

Reporter: Claire Mouton, DAVILA Eduardo, Patrick Clarysse, pierre croisille, William A. Romero R.

Viewer: Claire Mouton, coralie vandroux, DAVILA Eduardo, Frédéric Cervenansky, Maciej Orkisz, Patrick Clarysse, pierre croisille, William A. Romero R.

Spent time

0.00 hour

[Log time](#) | [Details](#) | [Report](#)

Manager

Claire Mouton DAVILA Eduardo

Maciej Orkisz Patrick Clarysse

pierre croisille William A. Romero R. CNRS Research Engineer

Download
inTag inTag v1.2-1B

Powered by Redmine © 2006-2013 Jean-Phillippe Lang

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Exemple : Redmine à CREATIS – Le projet OsiriX Plug-in >> inTag/Issues

The screenshot shows the Redmine web interface for the 'OsiriX Plug-in >> inTag' project. The top navigation bar includes 'Home', 'My page', 'Projects', and 'Help'. The user is logged in as 'mouton' with options for 'My account' and 'Sign out'. The search bar contains 'inTag'. The main navigation menu includes 'Overview', 'Activity', 'Roadmap', 'Issues', 'New issue', 'Gantt', 'Calendar', 'News', 'Documents', 'Wiki', 'Files', 'Repository', and 'Settings'. The 'Issues' section is active, showing a list of issues with filters for 'Status' (set to 'open') and 'Options'. The issues table has columns for '#', 'Project', 'Tracker', 'Status', 'Priority', 'Subject', and 'Updated'. The right sidebar contains 'Issues' (View all issues, Summary) and 'Graphs' (Open aging issues, Total issues over time, Total bugs over time, Calendar, Gantt).

Home My page Projects Help Logged in as **mouton** My account Sign out

OsiriX Plug-in » inTag Search: » inTag

Overview Activity Roadmap **Issues** New issue Gantt Calendar News Documents Wiki Files Repository Settings

Issues

Filters

Status open Add filter

Options

Apply Clear Save

#	Project	Tracker	Status	Priority	Subject	Updated	
<input type="checkbox"/> 2012	2012	inTag	Bug	New	Normal	Ambiguity with parameter identifier in xls files	05/28/2013 04:56 pm
<input type="checkbox"/> 1962	1962	inTag	Feature	New	Normal	Alternatives Motion estimators	03/29/2013 09:55 am
<input type="checkbox"/> 1937	1937	inTag	Bug	New	Normal	Limited access to sample data	03/18/2013 05:02 pm
<input type="checkbox"/> 1936	1936	inTag	Feature	In Progress	Normal	Contour interactive correction	08/26/2013 07:11 pm
<input type="checkbox"/> 1935	1935	inTag	Test	New	Normal	RefIntag Test	03/18/2013 05:03 pm
<input type="checkbox"/> 1934	1934	inTag	Test	New	Normal	inTag Evaluation	03/18/2013 05:04 pm
<input type="checkbox"/> 1933	1933	inTag	Feature	New	Normal	Point value picking	03/18/2013 05:04 pm
<input type="checkbox"/> 1931	1931	inTag	Feature	In Progress	Normal	Strain rates, peak strain rates computation	03/18/2013 05:17 pm

(1-8/8)

Also available in: Atom | CSV | PDF

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Exemple : Redmine à CREATIS – Le projet OsiriX Plug-in >> inTag/Wiki

The screenshot shows a Redmine Wiki page for 'inTag'. The page title is '[OsiriX Plug-in] inTag WIKI'. The main content area is titled 'Content' and lists four items: 1. Software development road map, 2. Developer Guide, 3. Meetings log, and 4. Recommended links and resources. A sidebar on the left also lists these items under the heading 'Content'. Below the content list is an 'Overview' section with text describing the software's purpose and a list of activities. At the bottom, there is a UML diagram showing a 'User' actor and an 'inTag' use case. The diagram includes a start node (black circle) pointing to a rounded rectangle labeled 'Start inTag', which then points to a horizontal line representing the system boundary. On the right side of the page, there are two profile pictures: one for 'pierre croisille' and one for 'William A. Romero R. CNRS Research Engineer'. Below the profiles is a 'Download inTag inTag v1.2-1B' button.

IV. Partager, travailler en équipe

Travailler en équipe : Les Forges

Exemple : Redmine à CREATIS – Le projet OsiriX Plug-in >> inTag/Dépôt

The screenshot shows a Redmine interface for a repository named 'OsiriX Plug-in » inTag'. The user is logged in as 'mouton'. The interface includes a search bar, navigation tabs (Overview, Activity, Roadmap, Issues, New issue, Gantt, Calendar, News, Documents, Wiki, Files, Repository, Settings), and a file browser showing the repository structure. Below the file browser is a table of 'Latest revisions' with columns for revision number, date, author, and comment.

root @ master Statistics | Branch: master | Tag: | Revision:

Name	Size
OsiriXPlugin	
wxInTag.skel	
.gitignore	162 Bytes
AxinoeLogo.xpm	23 KB
CMakeLists.txt	2.98 KB
CreatisLogo.png	4.99 KB
CreatisLogo.xpm	22.9 KB
README	10 KB
builtwithwx.png	2.7 KB

Latest revisions

#	Date	Author	Comment
a8cb985e	05/24/2012 02:18 am	Jean-Charles BERTIN	Added parameters to debug save panel.
ee1a20c5	05/24/2012 02:17 am	Jean-Charles BERTIN	Added inTag to saved parameters.
3f71b14d	05/24/2012 01:45 am	Jean-Charles BERTIN	Make wxInTag target optional.
c5f77ff4	05/24/2012 01:44 am	Jean-Charles BERTIN	Fixed compilation warning.
8214dea9	05/23/2012 08:44 pm	Jean-Charles BERTIN	Use Alternate key to access debug save panel.
153fa0ba	05/23/2012 08:30 pm	Jean-Charles BERTIN	Minor cleanup.
30343549	05/23/2012 08:27 pm	Jean-Charles BERTIN	Added Wait.h file.
37e03b5f	05/23/2012 08:25 pm	Jean-Charles BERTIN	Added slices versioning.
2d74de58	05/23/2012 08:25 pm	Jean-Charles BERTIN	First import.
83305de5	05/23/2012 08:25 pm	Jean-Charles BERTIN	Added recalculation of initial results with warping algorithm.

[View differences](#)

[View all revisions](#) | [View revisions](#)

Also available in: [Atom](#)

Powered by [Redmine](#) © 2006-2013 Jean-Philippe Lang

Les réseaux d'aide entre développeurs de l'enseignement supérieur et de la recherche

- **Développement logiciel**
 - Devlog : réseau national
devlog@services.cnrs.fr <http://devlog.cnrs.fr/>
 - Aramis : réseau régional sur Lyon
aramis@listes.resinfo.org <http://aramis.resinfo.org/>
- **Calcul**
 - Groupe Calcul : réseau national
calcul@listes.math.cnrs.fr <http://calcul.math.cnrs.fr/>
 - Lyon Calcul : réseau régional
lyoncalcul@sympa.rocad.fr <http://lyoncalcul.univ-lyon1.fr/>
- **CLUB des Développeurs : rendez-vous informel mensuel**
Inscription auprès de claire.mouton@creatis.insa-lyon.fr

Fin du cours...

À vous la main!

Des questions avant de plonger dans le code?