

# Les outils de construction du logiciel (T4)

---

Cette thématique aborde quelques grandes étapes de la construction du logiciel sous l'angle des outils. Ceux mis en avant (arbitrairement, tant ils sont nombreux et couvrent de problématiques) concourent à donner au logiciel ses qualités dites non fonctionnelles : maintenabilité, robustesse, réutilisabilité, extensibilité.

Chaque atelier permettra de pratiquer un de ces outils, en vue de donner au participant la capacité de le mettre en œuvre et d'en tirer profit dans son propre contexte.

Les « groupes de travail » aborderont, sous forme de table ronde ouvertes aux participants, une des problématiques de la construction du logiciel en comparant les pratiques actuellement en cours dans notre communauté. Ils mettront en perspective les différents outils qui dépendent souvent de l'écosystème du langage de base dans lequel est écrit le logiciel, et interrogeront, toujours grâce aux retours d'expérience, l'utilisation d'une classe d'outils dans un contexte particulier (développement seul ou à plusieurs, durée de vie du logiciel, exigences d'exécution particulière comme le temps réel ou le web etc.).

Les présentations introduiront des concepts un peu plus « état de l'art », mais dont l'impact sur les méthodes et outils de développements pourrait se révéler majeur à l'avenir.

Tous les acteurs de la communauté ESR ayant à produire du code destiné à être utilisé au sein de cette communauté, ou qui influent sur la production de ce code, sont concernés par cette thématique. Elle est toutefois particulièrement ciblée vers les développeurs « isolés dans leur laboratoire ».

## Présentations

*Vendredi 9h00-12h30, Amphi Gay-Lussac*

---

### **T4.P1 L'intégration continue : pourquoi ? - *Intervenant* : Christophe Demarey (INRIA)**

L'intégration continue est une pratique de développement logiciel issue des méthodes agiles et qui s'impose aujourd'hui comme un des piliers des bonnes pratiques du développement logiciel. Cette présentation s'efforcera d'expliquer ce qu'est l'intégration continue et comment la mettre en pratique. Elle présentera également un retour sur la mise en place d'une plateforme d'intégration continue à Inria.

### **T4.P2 Gestion de la qualité en programmation graphique - *Intervenant* : Laurent Vaylet (National Instruments)**

La programmation graphique a de nombreux avantages, mais impose au développeur de changer ses pratiques et outils pour gérer le cycle de vie du logiciel : documentation, débogage, gestion de version du code, packaging etc. Nous verrons à l'aide de l'exemple LabView comment et dans quelle mesure ces difficultés peuvent être surmontées.

### **T4.P3 DevOps - *Intervenant* : Marc-Elian Begin (SixSq)**

DevOps exige la mise en place d'une chaîne d'outils intégrés pour implémenter le déploiement agile et continu de services logiciels complexes (genre plusieurs composants s'exécutant éventuellement sur des machines différentes). Cette présentation passera en revue les outils DevOps disponibles et montrera comment certaines caractéristiques du module SlipStream permettent une réelle mise en pratique des principes de DevOps.

### **T4.P4 Les enjeux de LLVM - *Intervenant* : Henri-Pierre Charles (CEA)**

La boîte à outils LLVM a été conçue pour permettre la construction de compilateurs à la fois efficaces, versatiles et permettant de garder la structuration du code source dans la génération d'exécutable. Au-delà du compilateur Clang, elle a engendré beaucoup d'outils annexes et un énorme intérêt scientifique, technologique et industriel. Nous verrons les raisons qui ont suscité cet intérêt et quelques-uns de ces outils.

### **T4.P5 La méthode Sqale pour gérer la dette technique - *Intervenant* : Jean-Louis Letouzey (Inspairit)**

La dette technique, c'est l'ensemble des compromis que vous avez faits chaque jour pour tenir vos délais de livraison. Prendre conscience de votre dette technique va vous permettre de la gérer explicitement et ainsi vous guider dans votre stratégie de conduite de votre développement. La méthode open source Sqale est un exemple typique d'outil qui peut aider à maîtriser cette dette.

# Ateliers

---

## **T4.A1 Gérer son workflow de développement avec un DVCS, exemples avec Git et Hg**

*Intervenants* : Frédéric Magniette (LLR), Antoine Pérus (LAL), Fabrice James (LPC)

Comment gérer son processus de développement en s'appuyant sur Git/Mercurial et en s'inspirant du modèle de gestion de branches 'gitflow'/'hgflow'. L'atelier permettra également de se familiariser avec la pratique de Git/Mercurial et GitHub/Bitbucket.

## **T4.A2 Eclipse et sa galaxie pour gérer les étapes du développement**

*Intervenants* : Frank Buloup (Cnrs, ISM), Julien Nauroy (Inria, LRI)

Eclipse est à la fois un environnement de développement intégré et un atelier de génie logiciel extensible par un mécanisme de greffon (plug-in). Dans cet atelier, nous vous proposons d'utiliser Eclipse sur un cas pratique afin de produire un petit logiciel.

## **T4.A3 Génération de documentation avec Sphinx**

*Intervenants* : Marc Poinot (ONERA), Antoine Pérus (LAL).

L'outil Sphinx est présenté et appliqué au problème de la documentation du logiciel, doc de référence extraite des sources, doc utilisateur, doc d'installation, doc de release etc... Nous verrons un exemple de production de documents web et PDF, sur du code Python, C++ et Fortran depuis la spécification de l'interface jusqu'au manuel utilisateur.

## **T4.A4 Déployer sur le cloud : l'exemple SlipStream**

*Intervenants* : Marc-Elian Begin (SixSq), Charles Loomis(LAL).

SlipStream permet aux équipes DevOps de définir les composants individuels d'un service global, d'organiser et valider le déploiement de ces composants. Les participants à cet atelier l'utiliseront pour définir les composants d'un service multimachines, définir les tests de validation, et de l'utiliser comme un moteur de déploiement

## **T4.A5/T6.A3 Tracer et améliorer la qualité de son code avec Sonar, gérer sa dette technique avec SqaLe**

*Intervenants* : Olivier Gaudin (CEO SonarSource), Jean-Louis Letouzey (Inspearit), Michel Ilkiewicz (Inspearit)

Un module de Sonar permet d'évaluer un indicateur de la dette technique et mesure l'effort de remédiation qui reste encore à fournir. Cet atelier vous permettra d'appliquer la méthode SqaLe pour évaluer votre code par rapport à ce critère.

## **T4.A6 L'outil de reconstruction/déploiement Waf**

*Intervenant* : Maude Le Jeune (IN2P3, APC)

Waf est un outil de construction de logiciel, dans la lignée des Autotools (configure, make), CMake, SCons. Les participants à l'atelier seront guidés dans l'écriture d'un script simple de construction incluant la configuration (mise en place d'options), la compilation (utilisant les fonctionnalités de compilation C/C++, Fortran et Python) et l'installation, l'utilisation de Waf dans le cadre d'une application (performances et gestion des dépendances), et l'écriture d'extensions.

# Groupes de travail

---

## **T4.GT1 Pourquoi, comment et où déposer mon code ?**

*Animateurs* : Janet Berthot (INRIA, Nice), Sébastien Médard (Renater), Charles Loomis (IN2P3, LAL), Jean-René Rouet (IN2P3,CCIN2P3).

## **T4.GT2 La programmation graphique (Labview et co), forces et faiblesses**

*Animateurs* : Karin Dassas (CNRS, IAS), Laurent Vaylet (National Instruments), David Rey (INRIA, Num3sis), Christian Fournier (INRIA, OpenAlea), Xavier Maître (NeticLab)

## **T4.GT3 DevOps, Intégration continue et ALM**

*Animateurs* : Charles Loomis (IN2P3, LAL), Christophe Demarey (INRIA), Marc-Elian Begin (SixSq), Olivier Gaudin (SonarSource (CEO)), Jean-Louis Letouzey (Inspearit)