
	Titre du document : Sélection de l'outil d'intégration continue et spécification de son interface avec la forge
	Référence : L4.4.1
	Version 1.0 du 16/11/2010

Sélection de l'outil d'intégration continue et spécification de son interface avec la forge


Livrable du au titre du projet	COCLICO
Lot	Sous-projet 4 : Traçabilité et intégration continue
Tache	4 : Interface avec le système d'intégration continue
Livrable	L4.4.1 Sélection de l'outil d'intégration continue et spécification de son interface avec la forge

Rédacteur(s)	Vérificateur(s)	Approbateur(s)
Marc Nazarian	Nicolas Guérin	

	Titre du document : Sélection de l'outil d'intégration continue et spécification de son interface avec la forge
	Référence : L4.4.1
	Version 1.0 du 16/11/2010


Documents applicables
Annexe technique au projet COCLICO

Documents de références (pour information)
http://confluence.public.thoughtworks.org/display/CC/CI+Feature+Matrix
http://www.google.com/insights/search/#cat=5&q=udson%2Ccruise%20control%2Ccontinuum%2Ccruisecontrol%2Ctinderbox&cmpt=q
http://lijinoseji.wordpress.com/2008/05/10/comparing-two-continous-build-tools-cruise-control-and-hudson/
http://globalgateway.wordpress.com/2007/12/17/cruisecontrol-vs-hudson/
http://hudson-ci.org/
http://wiki.hudson-ci.org/display/HUDSON/Remote+access+API

	Titre du document : Sélection de l'outil d'intégration continue et spécification de son interface avec la forge
	Référence : L4.4.1
	Version 1.0 du 16/11/2010

Sommaire

1	Introduction.....	5
1.1	Définition.....	5
1.2	Vocabulaire spécifique.....	6
2	Outils existants.....	7
2.1	Cruise Control.....	7
2.2	Hudson.....	7
2.3	Continuum.....	8
2.4	Autres (TinderBox, Gump, etc.).....	8
3	Outil choisi.....	9
4	Spécification de l'interface avec la forge.....	10

	Titre du document : Sélection de l'outil d'intégration continue et spécification de son interface avec la forge
	Référence : L4.4.1
	Version 1.0 du 16/11/2010

1 Introduction

1.1 Définition

L'intégration continue est un ensemble de bonnes pratiques utilisées en génie logiciel. Ces bonnes pratiques visent à vérifier qu'une modification de code source n'entraîne pas de régression de l'application en cours de développement. Cette vérification est en générale effectuée sur une autre machine que votre machine de développement (serveur d'intégration); et cette vérification est effectuée assez fréquemment, d'où le nom d'Intégration Continue.


L'outil qui va se charger d'effectuer cette vérification est un outil d'intégration continue tel que CruiseControl ou Hudson. Cette étape de vérification est appelée **Build**. Un build pourra correspondre, selon votre projet, en une succession d'étapes, comme par exemple :

- Le déploiement,
- La compilation,
- La génération de la documentation (javadoc par exemple),
- L'exécution des tests unitaires,
- Une analyse de qualimétrie sur le code source (respect des conventions de codage, pourcentage de commentaires, métriques sur le code, ...),
- La génération d'un exécutable.

L'outil d'intégration continue n'effectue pas lui-même le build, mais le lance à intervalle régulier, affiche le résultat du build, et est en général capable de notifier les membres du projet si une modification a engendré une quelconque régression.

Cette technique de développement, initiée au départ par la communauté de l'Extreme Programming et adoptée par les méthodes Agiles apporte un certain nombre d'avantages. Parmi ceux-ci, nous pouvons citer :

- Les tests sont immédiatement exécutés après chaque modification (étape souvent négligée par les développeurs),
- Les problèmes d'intégration sont détectés en continue, permettant de les réparer au plus vite,
- Il existe toujours une version des livrables opérationnelle et disponible pour test, démonstration ou distribution.


	Titre du document : Sélection de l'outil d'intégration continue et spécification de son interface avec la forge
	Référence : L4.4.1
	Version 1.0 du 16/11/2010

1.2 Vocabulaire spécifique

L'intégration continue, possède un certains nombre de termes spécifiques. Pour clarifier la suite du document, en voici la définition :

Terme	Définition
Job	La notion de Job peut être mise en correspondance avec la notion de projet. C'est le Job qui va exécuter le build. Mais il ne fait pas que cela. En effet, le Job va mettre en place l'environnement nécessaire à l'exécution du Build (mise à jour du code source), exécuter le build, puis effectuer un certain nombre d'opération après celui-ci (comme publier la documentation générée, publier le résultats des tests, notifier du résultats du build, ...)
Build	Littéralement "construction" de votre projet logiciel. Processus constitué de plusieurs étapes exécutées périodiquement sur un serveur d'intégration continue.
Artéfact	Item généré lors du build, et mis à disposition par l'outil d'intégration continue.
Espace de travail (workspace)	Répertoire dans lequel le projet va être déployé pour pouvoir y effectuer le build, et éventuellement y construire les artefacts.
Statut (du build)	État du build. Peut prendre diverses valeurs selon l'outil. Hudson propose 4 états : <ul style="list-style-type: none"> • Succès : tout s'est bien passé, aucun test n'est en échec, • Instable : le build s'est achevé, mais il est instable (des tests sont en échecs par exemple), • Échec : le build est en échec • Inactif : aucun build n'a encore été lancé ou le build est désactivé.
Tendance (des builds)	Tendance du résultat des builds basé sur les 5 derniers builds. Cette tendance est représentée par une image empruntée à la météo (soleil, orage, etc) pour signifier que la tendance est au beau fixe ou pas.

Tableau 1. Glossaire des termes spécifiques à l'Intégration Continue

	Titre du document : Sélection de l'outil d'intégration continue et spécification de son interface avec la forge
	Référence : L4.4.1
	Version 1.0 du 16/11/2010

2 Outils existants

Une matrice assez complète présentant un état des lieux des outils d'intégration continue ainsi que leur fonctionnalités disponibles actuellement est consultable à l'adresse suivante : <http://confluence.public.thoughtworks.org/display/CC/CI+Feature+Matrix>

Si nous restreignons la liste aux outils open source supportant à la fois CVS et SVN, et n'étant pas basés sur les technologies .NET, les outils restant sont :

- Cruise Control
- Hudson
- Continuum
- TinderBox
- Gump

2.1 Cruise Control

Il s'agit de l'acteur historique dans le domaine de l'intégration continue. C'est probablement l'outil le plus déployé. Il propose le support de nombreux gestionnaires de code source et de langages.

Points positifs :

- l'outil est très utilisé,
- beaucoup d'information/documentation disponible

Points négatifs :


- la configuration est souvent considérée comme difficile (modifications de fichiers XML)
- l'outil commence à être ancien, la communauté est peu active

2.2 Hudson

C'est actuellement le projet le plus actif. Le projet rattrape rapidement son retard en terme de fonctionnalités par rapport à CruiseControl.

Points positifs :

- projet très actif,

	Titre du document : Sélection de l'outil d'intégration continue et spécification de son interface avec la forge
	Référence : L4.4.1
	Version 1.0 du 16/11/2010

- beaucoup de plugins disponibles,
- paramétrage réalisé dans l'interface web de manière graphique, avec beaucoup d'aide contextuelle

Points négatifs :

- moins utilisé (pour l'instant) que CruiseControl, mais la tendance semble s'inverser (voir documents de références)

2.3 Continuum

Outil d'intégration continue très bien intégré avec Maven

Points positifs :


- facile à utiliser si on utilise déjà Maven

Points négatifs :

- beaucoup moins adapté si l'on n'utilise pas Maven

2.4 Autres (TinderBox, Gump, etc.)

Les autres acteurs sont plus petits, plus anciens, avec une communauté moins active, et semblent moins pertinents que les précédents.

	Titre du document : Sélection de l'outil d'intégration continue et spécification de son interface avec la forge
	Référence : L4.4.1
	Version 1.0 du 16/11/2010

3 Outil choisi

La plupart des articles comparant Hudson et CruiseControl sont en faveur d'Hudson (voir documents de référence).

Même si ces chiffres sont à prendre avec des pincettes, les tendances de recherche du moteur de recherche Google montrent très clairement que Hudson est l'outil "en vogue" à l'heure actuelle :

Google Tendances des recherches


Comparer par	Termes de recherche	Filtre
<input checked="" type="radio"/> Termes de recherche <input type="radio"/> Lieux <input type="radio"/> Périodes	Conseil : Pour comparer des éléments, ajoutez une virgule (tennis, squash). <input type="text" value="hudson"/> <input type="text" value="cruise control"/> <input type="text" value="continuum"/> <input type="text" value="cruisecontrol"/> <input type="text" value="tinderbox"/>	<input type="text" value="Recherche sur le Web"/> <input type="text" value="Dans tous les pays"/> <input type="text" value="De 2004 à ce jour"/> <input type="text" value="Informatique et électronique"/>
		<input type="button" value="Rechercher"/>

Recherche sur le Web : intérêt pour hudson, cruise control, continuum, cruisecontrol, tinderbox

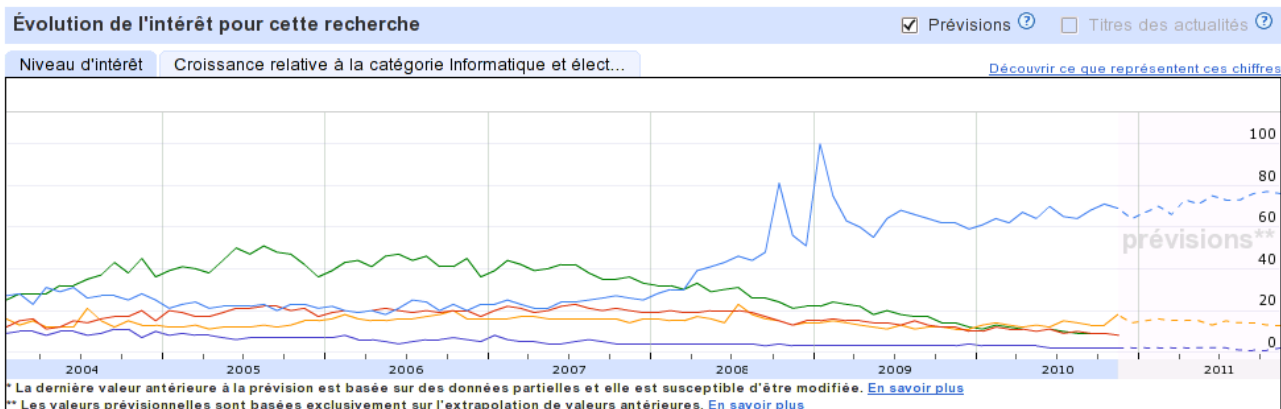
Dans tous les pays, De 2004 à ce jour

[Toutes les catégories](#) > Informatique et électronique

Sous-catégories : [Logiciels](#), [Programmation](#), [Matériel](#), [Réseau](#), [Électronique grand public](#), [autres...](#)


 Moins de 10 % des recherches contenant vos termes de recherche appartiennent à la catégorie Informatique et électronique. [En savoir plus](#)

Totaux	
hudson	43
cruise control	17
continuum	14
cruisecontrol	31
tinderbox	5



Hudson est l'outil déjà choisi et mis en place parmi certains partenaires du projet Coclico parmi lesquels Orange Labs, Objet Direct, Xerox.

Pour ces raisons, nous avons choisi l'outil d'intégration continue Hudson.

	Titre du document : Sélection de l'outil d'intégration continue et spécification de son interface avec la forge
	Référence : L4.4.1
	Version 1.0 du 16/11/2010

4 Spécification de l'interface avec la forge

Le système d'intégration continue devra pouvoir piloter les outils de la forge et inversement.

Pour ce faire, les outils utiliseront les API d'accès proposées par les différentes forges (API web service) et les API proposées par Hudson. Les API proposées par Hudson sont de style REST.

Voir <http://wiki.hudson-ci.org/display/HUDSON/Remote+access+API> pour plus de détails.

Hudson propose trois type d'API accessibles à distantes :

- XML
- JSON avec support JSONP
- Python

Ces trois API permettent toutes de faire les actions suivantes :

- récupérer de manière programmatique les informations de Hudson,
- Lancer un nouveau build,
- créer/copier des jobs.

L'API XML, de par son interopérabilité et sa facilité de manipulation semble être un bon candidat pour les besoins du projet Coclico. Si les partenaires rencontrent des difficultés ou ont des besoins spécifiques, ils pourront utiliser une autre API.

Notons qu'il est également possible de développer un plugin Hudson pour intéragir avec la forge. Si le besoin s'en fait sentir, nous pourrons étudier cette option.