

RETROUVEZ L'INFORMATICIEN



CHAQUE MOIS

CHEZ VOTRE MARCHAND DE JOURNAUX

(Retrouvez les sommaires et les conditions
d'abonnement sur www.linformaticien.com)

L'INTÉGRATION CONTINUE

Automatiser pour une productivité durable

L'intégration continue est une méthode de développement logiciel qui consiste à mettre en place un processus automatisé de « constructions de versions » de manière continue. L'objectif de cette méthode est de déceler automatiquement et le plus rapidement possible les erreurs de développement et d'intégration de nouvelles parties du programme.

Selon une étude réalisée par le Standish Group l'année dernière, seuls 32% des projets informatiques – moins d'un tiers – sont livrés à temps, dans le budget et avec les fonctionnalités prévus. Le quart – 24% – sera annulé ou jamais terminé, et plus de 52% d'entre eux auront un coût quasiment doublé par rapport à l'estimation ini-

tiale! Quant au délai moyen de dépassement, il est estimé à... 230% en moyenne.

Les raisons de ce « chaos » sont diverses mais, le plus généralement, c'est le découpage du projet en modules, et leur affectation à différents développeurs ou groupes sans que la cohérence globale soit prise en compte, qui est la cause essentielle du dérapage lors de l'assemblage des modules.

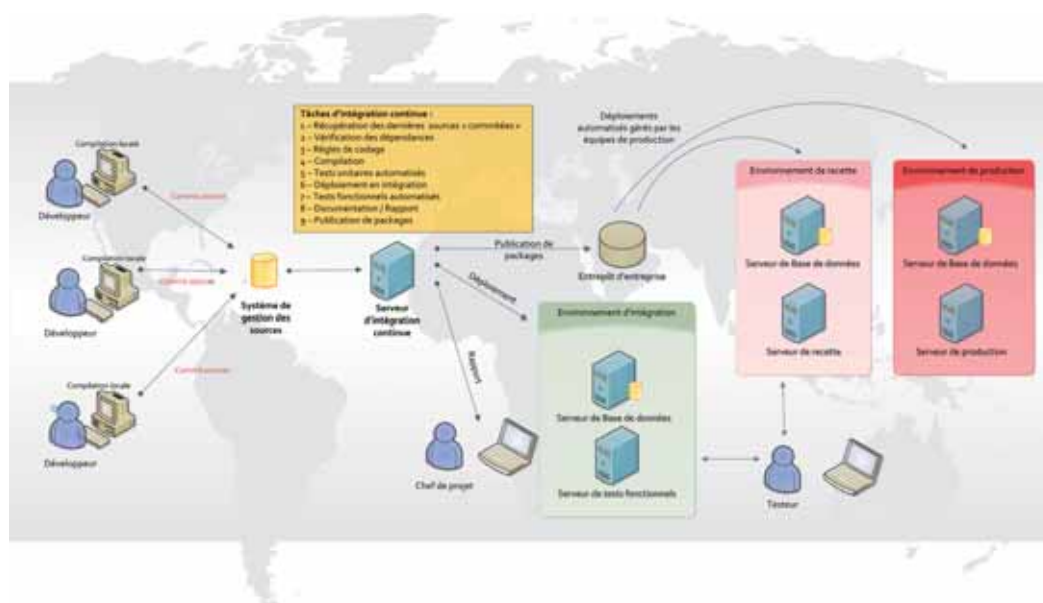
Un développement suivi à la trace

L'intégration continue permet aux équipes de développer des solutions cohésives plus rapidement en réduisant les problèmes d'intégration de nouveau code au sein d'une application. Notons que si cette technique constitue la pierre angulaire des méthodes qualifiées d'agiles, elle peut être mise en œuvre dans n'importe quel type de pro-

jet. La mise en place de cette méthode améliore la qualité globale des développements ; elle augmente la productivité, en particulier par le nombre de fonctionnalités sans erreurs ajoutées sur une période donnée. Cela contribue également à améliorer la traçabilité du développement, en particulier par la centralisation d'un référentiel unique. Pour les équipes, cela permet de plus facilement démontrer l'avancement du projet. Cette méthode permet également plus facilement de générer des versions de démonstration, car le suivi de processus permet à chaque moment de revenir en arrière si le développement s'avère non satisfaisant. En effet, la technique d'intégration continue permet de prendre en compte des changements à tout moment sans crainte que ces changements mettent en péril tout l'ensemble. Ce peut être ainsi résumé : fabriquer souvent, tester souvent et intégrer souvent, c'est-à-dire industrialiser le développement logiciel.

Le service d'intégration continue orchestre un ensemble d'outils dédiés : gestionnaire de code source, tests unitaires, validateurs (règles de codage, couverture de code...), compilateur, tests fonctionnels, tests d'interface,... Lors de la mise en place d'un tel système, il convient d'instaurer une véritable culture projet pour l'ensemble de l'équipe, y compris le management, de sorte que cela devienne une priorité pour l'ensemble des membres. Les équipes de production doivent être étroitement associées dans les processus de livraison/déploiement des applications. ■

Jean-Baptiste Marcé



↑ Principe du service d'intégration continue : automatiser le processus de construction d'une application.

Les tâches d'intégration automatique

Lors d'un processus d'intégration continue, voici les principales tâches qui sont effectuées afin de s'assurer que les modifications intervenues dans le code sont correctement prises en compte.

- 1 – Récupération des dernières sources « commitées ».
- 2 – Vérification des dépendances.
- 3 – Règles de codage (inspection / revue de code).
- 4 – Compilation.
- 5 – Tests unitaires automatisés.
- 6 – Déploiement en intégration.
- 7 – Tests fonctionnels automatisés.
- 8 – Documentation / rapport : notification.
- 9 – Publication de packages pour déploiement en production.

Le déclenchement de l'intégration peut s'accomplir chaque fois qu'un changement est enregistré, toutes les nuits ou encore à la demande. De manière pratique, il est recommandé de procéder à chaque fois qu'une modification est intervenue, afin de vérifier que les nouveautés introduites s'accordent avec le projet. Dans le cas contraire, il est plus difficile de détecter l'origine du problème.

Les outils

Il existe de très nombreux logiciels qui permettent la mise en œuvre d'un projet d'intégration continue. Nous listons ci-dessous les principaux produits en termes de gestionnaires de code source, outils de tests et orchestrateurs d'intégration continue.

Services d'intégration continue :

- Anthill Pro
- Atlassian Bamboo
- Build Forge
- Cruise Control et Cruise Control .NET
- Apache Continuum : l'outil « officiel » de la communauté Maven
- Hudson
- Luntbuild (et Luntbuild pro)
- JetBrains TeamCity
- Microsoft Team Foundation Server

Source Control Systems :

AccuRev, CVS, Subversion SVN, Visual Source Safe, Perforce, Rational ClearCase, SourceGear Vault, TeamSystem

Build tools :

Ant, Nant, Visual Studio .NET, MSBuild, FinalBuilder, Javac

Test, Coverage and Metric tools :

Junit, Nunit, MSTest, FxCop, Ncover, VIL, Simian, MBUnit, FitNesse, Ndepend, Gendarme, Selenium, Green pepper
CppUnit (informal)

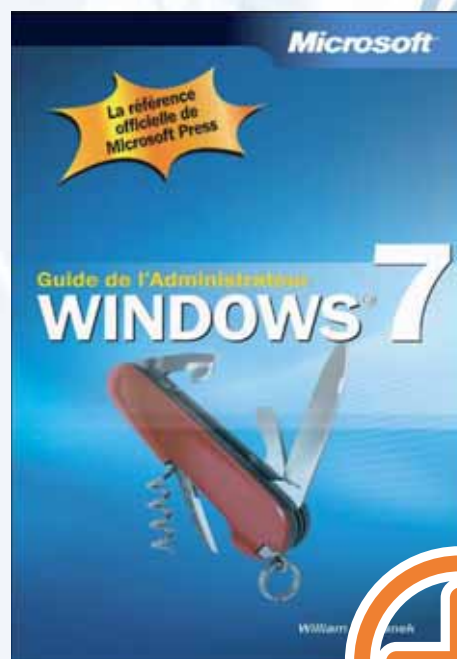
À propos de l'auteur

Jean-Baptiste Marcé est le co-fondateur et le directeur technique de la société nLiive. Il totalise plus de vingt ans d'expérience dans la conception et l'industrialisation du développement logiciel. Le cabinet de conseil nLiive est spécialisé dans les technologies Microsoft .NET et la plate-forme de Cloud-computing de Microsoft : Azure. nLiive est également éditeur de solutions distribuées en mode SaaS (Software as a Service). Plus d'infos sur www.nliive.com

ABONNEMENT

Offert : **1 AN**

Le guide de l'administrateur WINDOWS 7



59 € Seulement

Abonnez-vous sur
www.linformaticien.com