

Développement , vers. l'usine logicielle

COLLABORATIF. Les plates-formes de 2005 industrialiseront le travail des développeurs. Un big-bang aussi important que l'arrivée des progiciels intégrés.

Le développement logiciel nécessite une parfaite cohésion entre les différents intervenants du projet. C'est à cette condition que délais, budget et niveau de qualité sont respectés. Mais les outils font encore défaut. « *Il n'existe aucune suite de développement logiciel intégrant des outils de collaboration et de gestion de projet* », déplore Franck Gonzales, fondateur du cabinet de conseil Osaxis. Difficile, dès lors, d'orchestrer le travail d'une équipe. D'autant que « *les environnements de développement intégrés ne couvrent pas tous les besoins, et que les outils isolés communiquent mal entre eux* », constate Carl Zetie, analyste chez Forrester Research. Chefs de projet, architectes et développeurs travaillent aujourd'hui chacun de leur côté. Ce qui

contraint les entreprises à s'organiser avec les moyens du bord : répertoires partagés pour échanger les documents d'analyse, envoi par courrier électronique des applications à déployer, liste de « à faire » transmise aux développeurs dans des fichiers Excel et à consolider ensuite pour le rapport d'activité, etc.

La communauté open source montre la voie

Cet artisanat se traduit par un taux d'échec record : seuls 16 % des projets tiennent leurs engagements, tandis que 37 % sont purement et simplement arrêtés en cours de route. Les 47 % restants dépassent les délais ou le budget initialement prévus. Telles sont les conclusions d'une étude menée par Borland auprès d'entreprises françaises.



La situation n'est pourtant pas désespérée. Contraintes par l'éloignement géographique et les barrières culturelles, les communautés open source ont commencé à industrialiser leurs projets logiciels. Basées sur des outils de collaboration, tel Sourceforge, associés à d'autres briques open source comme Mantis (gestion de bogues), CVS (gestion de configuration) ou Maven (gestion de builds), ces plates-formes de développement collaboratif ont fait leurs preuves sur plus de 70 000 projets. Elles sont encore imparfaitement intégrées aux outils des grands éditeurs. Néanmoins, elles rationalisent la collaboration des équipes. De grandes entreprises, telle BNP Paribas, s'en sont inspirées pour assembler des outils commerciaux et open source afin de construire leurs propres plates-formes de développement collaboratif. « *Nous utilisons le*



GLOSSAIRE

ALM/LCM (Application Lifecycle Management/Development Lifecycle Management)

Ces deux acronymes sont aussi parfois traduits par l'expression générique de développement collaboratif. Ils désignent les outils prenant en charge tout le cycle de développement d'une application et facilitent la collaboration entre les différents acteurs d'une équipe.

Build

Version compilée [binaires] du code source, prête à être déployée. Elle prend, par exemple, la forme d'un fichier exécutable [.exe] sous Windows.

IDE (Integrated Development Environment)

En français, environnement de développement intégré. Suite logicielle fournie par un même éditeur pour couvrir tous les besoins des développeurs. Ces outils s'ouvrent peu à peu aux solutions de partenaires via des extensions.

Intégration continue

Principe d'organisation des équipes projet autour d'une plate-forme de développement. L'unité de temps est la journée. Chaque nuit, la plate-forme (ou software factory) exécute des tâches automatiques : validation du code, compilation...

Software Factory

Automatisation d'une partie de la production d'une application. Lorsqu'une source est déposée sur la gestion de configuration, un robot tente de la compiler. Il effectue ensuite un ensemble de tests (unitaire, performance, conformité...). Puis il génère la documentation, compile le fichier binaire, et renvoie éventuellement les erreurs détectées aux développeurs incriminés.

La révolution industrielle

En rationalisant l'organisation des projets autour d'un socle logiciel unique, les éditeurs espèrent accélérer et fiabiliser les projets. A la clé, une meilleure productivité collective.

LE CONSTAT

- 30 % des difficultés sont liées à une mauvaise intégration entre les outils.
- 47 % des équipes informatiques attendent davantage de communication.
- 37 % des entreprises veulent accélérer le processus de développement.

LES PROMESSES

- Toutes les données du projet sont fédérées au sein d'un référentiel ouvert.
- Des outils de collaboration facilitent le partage des informations.
- Les tâches rébarbatives sont automatisées et fiabilisent le projet.

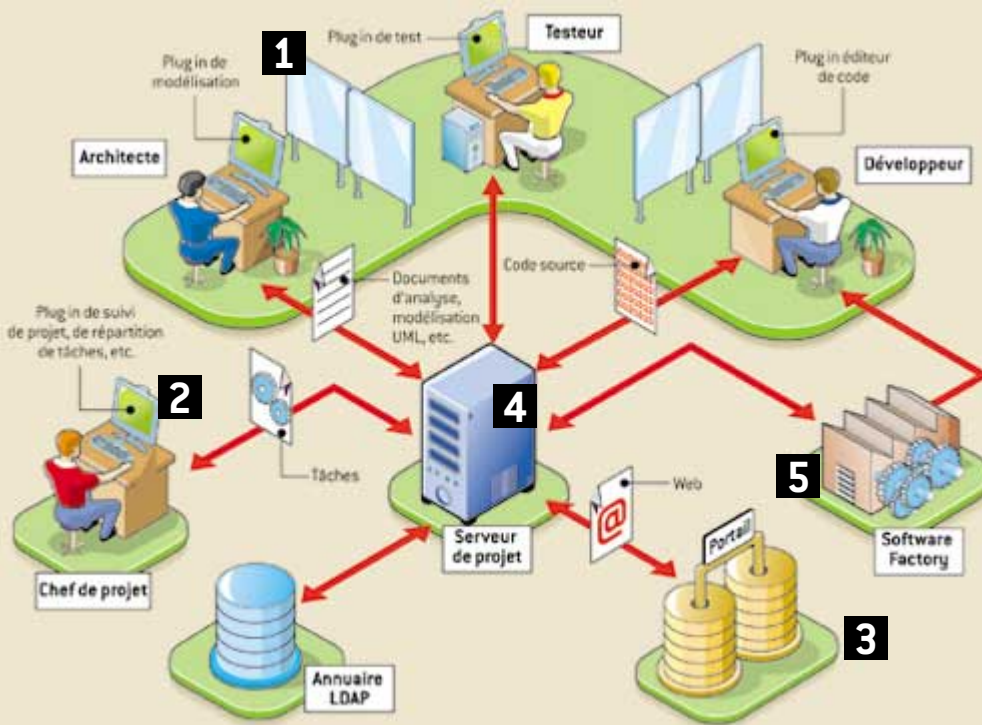
gestionnaire de configuration Starteam de Borland comme référentiel pour stocker tout notre code, mais aussi les documents d'analyse, décrit Pascal Jarry, responsable architecture technique, BNP Paribas Securities Services. L'information est partagée par le biais d'un intranet et de listes de diffusion. Nous utilisons aussi les logiciels libres Maven et Jelly pour documenter et

automatiser les builds, ainsi que JUnit et Clover pour les tests unitaires. »

Les entreprises sont cependant nombreuses à n'avoir ni les compétences techniques ni le temps pour assembler leur propre plate-forme, brique par brique. Conscients de cette lacune, Microsoft, IBM et Borland ont tous annoncé de nouvelles plates-formes de développement ●●●

La plate-forme de demain : intégrée, mais modulaire

Au centre des nouvelles plates-formes : un serveur de projet avec sa base de données. Des extensions viennent s'y greffer pour fournir les outils nécessaires à chacune des populations.



Les treize étapes d'un projet

- 1/ Création du projet.
- 2/ Définition des exigences.
- 3/ Modélisation fonctionnelle UML.
- 4/ Modélisation technique UML.
- 5/ Génération des squelettes de classes.
- 6/ Développement.
- 7/ Tests et profiling.
- 8/ Publication du code source.
- 9/ Compilation et construction de l'application (build).
- 10/ Révision automatique du code source.
- 11/ Génération automatique de la documentation.
- 12/ Packaging et déploiement.
- 13/ Gestion des changements.

1 Un outil personnalisé

Les participants utilisent le même environnement de développement personnalisé par des plug in fournis par l'éditeur ou ses partenaires. Ils s'interfaçent avec le serveur de projet et couvrent tout le cycle de développement.

2 La gestion de projet

Le chef de projet dispose d'outils pour distribuer les tâches aux développeurs et suivre le déroulement du projet. Il peut automatiser une partie du suivi à l'aide d'indicateurs clés et rendre certaines étapes obligatoires.

3 Une interface web

Team Foundation, de Microsoft, est doté d'interfaces de programmation pour créer ses propres applications web. L'entreprise expose les données du projet pour construire un tableau de bord ou mieux suivre un projet offshore.

4 Un référentiel unique

Il fédère toutes les données du projet dans une base ou dans une gestion de configuration, favorisant ainsi la collaboration. Chaque événement (dépose d'un code source, résultats de tests...) peut déclencher une action.

5 La Software Factory

Cette usine logicielle automatise la fin du cycle de développement. Ce dernier comprend l'intégration du code, la compilation, les tests en tout genre, et le déploiement des builds.

●●● collaboratif pour l'année prochaine. Après avoir amélioré la productivité individuelle, les éditeurs cherchent désormais à associer leurs outils et à compléter leur couverture fonctionnelle pour couvrir tout le cycle de vie d'un projet. « C'est la seule façon d'améliorer la qualité et de gagner en productivité », estime Bruno de Combiens, responsable marketing produits chez Borland.

Pour tenir cette promesse, IBM et Microsoft ont découpé leurs ateliers de

Les éditeurs veulent associer leurs outils et couvrir tout le cycle de vie d'un projet

développement en un ensemble de modules. Ceux-ci sont fédérés, côté serveur, par un socle d'intégration unique et, côté client, par un environnement de développement personnalisable.

Concevoir des solutions indépendantes

Team Foundation, le socle d'intégration de Microsoft, concentre dans une base SQL Server toutes les données du projet :

exigences, diagrammes UML, code source, résultats des tests, etc. Ces données sont ensuite exposées au travers d'une interface de programmation XML, accessible directement au sein de Visual Studio Team System ou via un portail. Grâce à ce couplage lâche, cette architecture dissocie l'évolution des outils de développement de celle des données des projets. Celle-ci offre aussi une plus grande ouverture. Editeurs et entreprises peuvent développer des solutions indépendantes de Visual Studio : rapports directement accessibles au sein d'un portail, compilations lancées à l'aide de scripts sans passer par l'environnement de développement, etc. Moins avancé sur cette partie serveur, IBM s'appuie, lui, sur sa gestion de configuration, dans laquelle il stocke les données du projet à l'aide de fichiers XML.

Un suivi automatique de l'avancement du projet

Avec l'apparition du référentiel projet, les outils de développement côté client ont, eux aussi, subi un sérieux lifting. Pour répondre aux besoins de l'ensemble des acteurs d'un projet, sont apparus des modules de test, de modélisation et de suivi de projet. Borland a récemment racheté Estimate pour la gestion de projet. Atlantic, d'IBM, et Visual Studio 2005 Team System, de Microsoft, permettent au chef de projet d'affecter des tâches à des développeurs et de suivre automatiquement le déroulement du projet. Celui-ci peut, par exemple, recevoir chaque matin un bilan complet sur l'état d'avancement des développements. Notamment le nombre de classes développées et testées, les statistiques du projet par lot, par développeur ou par indicateur (comme le pourcentage de bogues). Il peut aussi être alerté en fonction d'événements paramétrables.

Outre l'extension fonctionnelle, ces environnements de développement reposent sur une nouvelle architecture technique, liée à l'apparition du serveur et à la volonté des éditeurs de fournir un seul outil client. Architecte, développeur et chef de projet partagent désormais le même environnement personnalisé à l'aide de plug in, en fonction de leur rôle : module de modélisation pour l'architecte, éditeur de code pour le développeur, tableau de bord pour le chef de projet. N'apparaissent à l'écran que les menus et fonctions auxquels l'utilisateur a le droit

Sur le terrain oui...

Bruno Boucard,
architecte à la Société Générale

« Le chef de projet est enfin pris en compte »

« La première qualité des nouvelles plates-formes est l'intégration collaborative. Affecter, par exemple, des tâches à des développeurs directement dans l'environnement de développement soulagera le chef de projet. Aujourd'hui, tout le monde fonctionne par e-mail, avec des feuilles Excel. Cela ne pose pas de problème pour une petite équipe de cinq personnes. Mais avec cent cinquante développeurs, cela devient autrement plus compliqué à gérer. Dans le cas de Microsoft, le socle d'intégration sera, à terme, très intéressant pour intégrer des équipes n'utilisant pas les mêmes outils. Mais il faudra d'abord que les éditeurs de ces outils s'interfacent avec l'API de Microsoft. »



F. SUTTER/AREA

...mais

Laurent Kempé, architecte
chez Helvetia Patria Group

« L'industrialisation à 100 % est impossible »

« Les prochains environnements permettront au développeur de se concentrer sur l'essentiel de son travail : coder. Ils lui apporteront aussi plus de rigueur en l'aidant à respecter une méthodologie de projet. L'adhésion à ces nouvelles méthodes dépendra beaucoup de la capacité des outils à intégrer directement ces contraintes méthodologiques dans leur fonctionnement. Elle sera aussi conditionnée au degré de liberté laissé au développeur. En effet, la créativité ne peut être industrialisée à 100 %. Enfin, ce n'est qu'une fois mesurés des gains concrets sur plusieurs projets et en respectant une certaine déontologie que ces outils seront totalement acceptés. »



KEYSTONE

d'accéder. Si bien que chaque profil possède un poste de travail taillé sur mesure et entièrement personnalisable. Microsoft suit la même démarche. Visual Studio 2005 est un socle modulaire, sur lequel les entreprises connectent les plug in issus d'éditeurs spécialisés.

Au-delà d'une meilleure intégration, ces outils industrialisent le cycle de développement en intégrant une dimension méthodologique et en automatisant la chaîne de production des applications. La personnalisation de l'environnement de développement sert à respecter une métho-

dologie précise. Cela en obligeant certains profils à effectuer des tâches dans un ordre donné ou en les contraignant à respecter des étapes de validation. Pour ajouter une classe dans la gestion de configuration, le développeur devra, par exemple, l'avoir préalablement testée ou validée auprès de son responsable technique. Cette gestion de processus s'appuie sur les Enterprise Templates chez Microsoft, et sur les Perspectives d'Eclipse chez IBM. Pour être efficace, cette démarche impose cependant de formaliser les processus et le rôle de chaque individu. Forrester ●●●

IBM et Microsoft imposent leur vision face à Borland

Les deux éditeurs proposeront, dès l'an prochain, des plates-formes à la fois modulaires et intégrées, basées sur une vision PGI du développement collaboratif. Borland, lui, reste sur une vision fédératrice

EDITEUR, PRODUIT	Borland, Themis	IBM, Atlantic	Microsoft, Visual Studio 2005 Team System
LANGAGES	• Java/J2EE	• Tous	• C#, VB.Net, etc.
SOCLE TECHNIQUE	• Primetime	• Eclipse 3.0 (open source)	• .Net Framework 2.0
SERVEUR DE PROJET	• A terme	• Partiel	• Oui
UN SEUL SOCLE CLIENT	• Non	• Oui	• Oui
COUVERTURE FONCTIONNELLE NATIVE	• Gestion de projet, outils de collaboration, gestion de configuration, gestion des exigences, gestion de builds, IDE et modélisation UML	• Gestion de projet, outils de collaboration, gestion de configuration, gestion des exigences, gestion de builds, outils de test, profiler, suivi des bogues, IDE, modélisation UML	• Gestion de projet, outils de collaboration, gestion de configuration, gestion de builds, outils de test, profiler, suivi des bogues, IDE, modélisation UML (partielle)
REMARQUES	• Pas de socle d'intégration commun, mais une association entre les outils	• Distribution d'Eclipse 3.0 enrichie des outils de Rational et de fonctions de collaboration	• L'ensemble des données du projet est stocké dans une base unique

●●● Research estime ainsi que, comme pour les progiciels de gestion intégrés, le paramétrage initial des premiers projets nécessitera un important investissement. Microsoft propose donc des modèles de projets préconfigurés (XP, CMMI...) et adaptables pour accélérer le déploiement de Visual Studio 2005 Team System.

Ces plates-formes fiabilisent et accélèrent également le projet en automatisant la fin du cycle de développement au sein d'une usine logicielle (Software Factory). Lorsqu'un code source est dépo-

sé sur la gestion de configuration, le serveur prend le relais. Entre autres, ce dernier assemble et compile le code fourni par plusieurs développeurs, effectue des tests, et publie les builds dans le référentiel. Un travail méticuleux, parfaitement adapté au potentiel d'un robot, et qui, en outre, décharge les développeurs. Lesquels peuvent alors se consacrer à des tâches à plus valorisantes.

Même s'ils font gagner du temps aux développeurs, ces outils sont contraignants. Par ailleurs, de par leur capacité de suivi,

ils peuvent être perçus comme des instruments de contrôle. Pour obtenir l'adhésion, « il faut donc former et sensibiliser les acteurs du projet à ces nouvelles dimensions », conseille Jean-François Vallet, consultant chez Osaxis. Souvent aussi, les équipes auront à réorganiser leur cycle de développement à la journée pour être capables d'automatiser la production des builds dans une logique de Software Factory. L'impact de ces outils n'est donc pas neutre en termes d'organisation et de conduite du changement.



AVIS D'EXPERT

Sami Jaber, architecte chez Valtech

La SSII Valtech est réputée pour ses compétences Java. Sami Jaber connaît également très bien l'architecture Microsoft. Il a fondé la communauté www.netguru.org.

« Atlantic et Visual Studio 2005 bâtissent une Software Factory.

Le grand mérite de ces deux offres est d'automatiser les tâches journalières rébarbatives. On entre dans une logique d'intégration continue, où l'unité de temps devient la journée. Chaque soir, les développeurs publient leurs travaux sur le serveur. »

« Les développeurs passent le relais au serveur.

Ce dernier effectue la revue du code [vérification des règles de nommage, détection préventive des failles de sécurité, analyse statique des goulets d'étranglement], la compilation, le lancement de tests unitaires module par module, la génération de la documentation, etc. »

« Un travail qu'un homme seul a du mal à réaliser.

Quand la Software Factory détecte une erreur, elle envoie une alerte au développeur directement dans son environnement de développement. Sa précision chirurgicale lui donne la possibilité d'indiquer la ligne exacte incriminée et la règle non vérifiée. »

Une migration progressive

Dans tous les cas, la migration risque d'être progressive. « La possibilité de n'utiliser que les modules désirés au fur et à mesure de l'évolution de nos besoins sera un facteur déterminant », assure Olivier Bourassé, directeur architecture du système d'information de Radio France International. De plus, les outils de développement actuellement utilisés en entreprise n'ont que très rarement été conçus pour dialoguer avec les nouvelles API d'Eclipse ou de Visual Studio Team System Foundation. Les entreprises risquent donc d'attendre, avant de migrer progressivement, que les outils qu'elles utilisent aujourd'hui s'intègrent à ces nouvelles plates-formes de développement collaboratif. ●

FREDERIC BORDAGE