

PROgrammez !

mensuel n°149 - février 2012

www.programmez.com

Au cœur de Windows 8



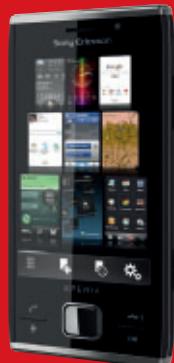
Nouveau
Le moteur
WinRT

Dossier spécial



Team Foundation Server Tout pour la qualité logicielle !

100% Geek



Prendre le contrôle
de son PC
avec un
smartphone

100% GNU



Créer des archives
de sources
avec autotools

M 04319 - 149 - F: 5,95 €



Printed in France - Imprimé en France - BELGIQUE 6,45 €
SUISSE 12 FS - LUXEMBOURG 6,45 € - DOM Surf 6,90 €
Canada 8,95 \$ CAN - TOM 940 XPF - MAROC 50 DH

**DÉVELOPPEZ
10 FOIS
PLUS VITE**

WINDEV®

- Windows 32 & 64 bits
- Linux
- Mac
- Internet
- Intranet
- Windows Mobile & CE
- Windows Phone
- Android 
- et maintenant*
- iPhone et iPad.

● Développez vos applications une fois pour toutes (les plateformes).
 Votre code, vos fenêtres, vos données, vos rapports,... sont compatibles.
 Déployez vos applications sur tous les systèmes et tous les matériels, dans tous les domaines, pour toutes les volumétries.
 Vous aussi, développez 10 fois plus vite, pour toutes les plateformes.

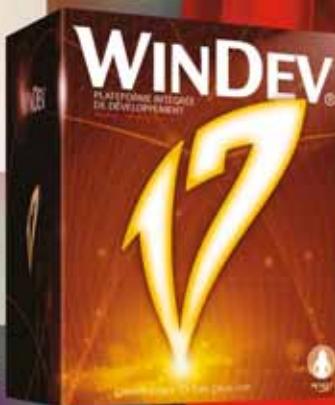
**VERSION
EXPRESS
GRATUITE**

Téléchargez-la !

Intégralement en français.
 Support Technique inclus.
 Ouvert à tous les standards,
 à toutes les bases de données.

Élu «Langage le plus productif du marché»

WINDEV, WEBDEV et WINDEV Mobile sont compatibles



Nouveau:
 créez des applications iOS (iPhone, iPad)



**EXIGEZ WINDEV 17
POUR LE DÉVELOPPEMENT
DE VOS APPLICATIONS**

Fournisseur Officiel de la
Préparation Olympique

► DEMANDEZ VOTRE DOSSIER GRATUIT

Dossier gratuit 260 pages sur simple demande. Tél: 04.67.032.032 info@pcsoft.fr



www.pcsoft.fr

Mise à jour automatique : le débat relancé

Depuis des années, la question se pose de savoir s'il faut imposer ou non la mise à jour automatique des logiciels sur les postes de travail.

En décembre dernier, Microsoft a lancé un nouveau pavé dans la mare aux logiciels : « *aujourd'hui, nous partageons notre plan de mise à jour automatique (pour les utilisateurs Windows) pour rendre disponible sur les PC la dernière version d'Internet Explorer* ». Microsoft prend exemple sur Google et Mozilla qui mettent à jour très régulièrement (trop ?) leur navigateur et imposent la mise à jour. Bien entendu, l'équipe Internet Explorer dit qu'il faut respecter le choix de l'utilisateur et son contrôle en lui donnant la possibilité de décocher l'automatisme et de revenir à une version n-1. Cependant, par défaut, la mise à jour sera automatique.

Ce mini buzz mérite tout de même toute notre vigilance. Car pour l'utilisateur, cette procédure est une perte de contrôle et de liberté sur son PC. Oui, la mise à jour est toujours fastidieuse et longue, mais l'utilisateur garde le choix et le contrôle de sa machine. Ce qui est le plus important. A lui de prendre ses responsabilités. Un éditeur n'a pas à imposer son choix. L'éditeur arguera qu'il laisse le choix, a posteriori. Et combien d'utilisateurs iront désactiver l'action et revenir à une version antérieure ?

Ce buzz n'est pas anodin. Il soulève une réelle question que chaque utilisateur devrait se poser : suis-je le maître de mon ordinateur ? La réponse demeure ambiguë.

POUR LE DÉVELOPPEUR, UN CASSE-TÊTE DE PLUS À GÉRER ?

Pour le développeur, l'automatisme des mises à jour pose de sérieuses questions. Car si cet automatisme concerne des éléments systèmes, une version majeure du navigateur, un player media, etc. il se doit de tester et vérifier le fonctionnement du code, de l'application, puis de proposer, un patch, une nouvelle version.

D'ores et déjà, les pré-requis techniques de certains IDE ou technologies contraintent à mettre à jour des éléments systèmes ou de bas niveau que l'utilisateur n'aura pas forcément. Or, combien de PC sont réellement à jour ? Il n'y a qu'à voir la résistance d'IE 6 dans le monde web, Microsoft a eu la plus grande difficulté à le « tuer », ce qui a empêché la vie des webmasters. Aujourd'hui, l'autre poison est la multiplication des navigateurs, mais cela est aussi une bonne chose pour notre liberté. Dans le cloud computing, la situation est encore plus floue.

Dans un PaaS, le fournisseur va déployer la mise à jour technique directement dans le PaaS et proposera un SDK adapté pour les nouvelles fonctions, les corrections. Or, la migration vers le nouveau SDK sera quasi obligatoire pour éviter des problèmes de compatibilité des applications. Mais le développeur devra alors requalifier le

code et l'application même si, théoriquement, le changement de SDK n'apportera pas de rupture de compatibilité. Cependant entre les API obsolètes, les nouvelles API, les API modifiées, nous ne savons jamais où se cache le problème. Et ni le développeur, ni l'utilisateur, n'ont le moindre contrôle sur la mise à jour du PaaS. C'est le principe même du cloud mais finalement, nous actions la mise à jour automatique... Et pour les outils de développement en mode SaaS, le constat est encore plus sévère, car là, nous ne pouvons strictement rien faire ! Or, pour un développeur, ne pas maîtriser son environnement, c'est perdre le contrôle et se fier à l'éditeur... Un autre problème est l'obsolescence imposée par les nouveaux outils et les technologies. Prenez l'exemple de iBooks Author d'Apple, qui officiellement ne fonctionne qu'avec OS X 10.7 alors qu'il peut parfaitement fonctionner sous un OS X 10.6.x. Or le développeur n'a pas forcément envie de migrer de système. Mais le cas d'Apple est loin d'être unique.

HTML 5 qui ne cesse de faire le buzz, pose lui aussi un problème car les spécifications finales du langage n'étaient pas définies que les éditeurs se sont empressés de proposer parfois du HTML 5, au risque de devoir tout casser au bout de 3 à 6 mois.

Vous l'aurez compris, notre buzzword du mois n'est peut être pas si buzz que cela mais il soulève de nombreuses questions pour le développeur et son travail. Dernier exemple, les évolutions d'interface graphique qui s'imposent à nous. Là encore le développeur n'a pas grand chose à dire, il suit.

Ce n'est pas un hasard si un grognement a eu lieu lors de la présentation de Windows 8 ou encore le revirement à 180° d'Adobe sur Flash Mobile et même Flex.

Car outre la mise à jour automatique, on constate que du jour au lendemain, une technologie peut passer à la trappe et alors, l'éditeur impose une migration vers autre chose.



Le top des téléchargements pour : Toutes les catégories Rechercher

Le Top des téléchargements du mois

- Pack de compatibilité Microsoft Office pour les formats de fichier Word, Excel et PowerPoint
- Visionneuse PowerPoint
- Microsoft .NET Framework 4 (programme d'installation Web)
- Composant d'exécution utilisateur final DirectX
- Conseiller de mise à niveau Windows 7
- Plus de téléchargements...

Et les nouveaux téléchargements

- Module d'intégration d'Office 365 pour Windows Small Business Server 2011 Essentials
- Test - French
- Fournisseur Microsoft Outlook Social Connector pour Facebook
- Extension de création de rapports Microsoft Dynamics CRM 2011
- Module linguistique Microsoft Dynamics CRM 2011
- Plus de nouveautés...

L'appli du jour

Application MyMemory Master Edition

Application Instant TV

Application UpdateMyFeed

Application MyQuiz Master

PROGRAMMEZ ! < 3 > FÉVRIER 2012



6



16

BUZZWORD

Mise à jour automatique : le débat relancé 3

ACTUS

En bref 6

OUTILS

Utiliser un bug tracker 12

MATÉRIEL 16

SÉCURITÉ

Sécuriser votre mot de passe 19

VEILLE TECHNO

au cœur De Windows 8 : WINRT 22

RESSOURCES

Notre sélection de livres 28

Agenda 29

Le Développeur du mois

De la chimie à l'open source 30



22

Team Foundation Server Tout pour la qualité logicielle



• GAMME

A vous de choisir ! 3

• Un ALM sur mesure 4

• GESTION

Cycle de vie des applications : pourquoi y aller ? 6

• AGILITE

Visual Studio et l'agilité 8

• QUALITE

- Collaboration, partage, communication : les clés d'une équipe agile et de qualité 11

- Tests unitaires, d'interface utilisateurs, de charge : une palette d'outils pour maîtriser votre qualité logicielle 12
- L'automatisation de tests sous Visual Studio 15
- Planifier et organiser les tests avec Microsoft 18
- Améliorer la qualité logicielle avec Team Foundation Server 2010 19
- Intégration continue et tests avec TFS Lab Management 20

• FUTUR

- Aperçu de la prochaine version 23
- Les principales nouveautés VS 11 sur le cycle de vie 24

Mars 2012 - n°150, parution le 29 février 2012

Un numéro à ne pas manquer !

Soyez fier d'être développeur !

Comment allons-nous développer dans 10 ans ?

Les Développeuses : la France toujours à la traîne

Hadoop : la techno incontournable du moment



88

CARRIÈRE

Les métiers de l'internet : une profonde évolution des profils et des compétences	32
Tendances de l'emploi informatique.....	34
Atos : 1000 recrutements en 2012	36

Je DÉBUTE

Introduction à l'architecture CUDA (3e partie).....	37
PhpStorm : accélérer les développements PHP.....	66
Bien débuter avec Visual C++	69

PRATIQUE

TestNG :	
le framework de tests nouvelle génération !	72
Prenez le contrôle de votre ordinateur à partir de votre Smartphone	76

cas D'USAGE

Personnaliser un thème sous Drupal 6	80
--	----



Construire une distribution de sources GNU

84

GAMING

Concevoir un design d'interaction sur Surface 2.0.....	88
--	----

WEBMASTER

Comment traquer (efficacement) les bugs de son site Web ?	92
---	----

BUGTRACK

Repérer une valeur NULL	98
-------------------------------	----



76

Editeur : Go-O2 srl, 21 rue de Fécamp 75012 Paris - diff@programmez.com.

Rédaction : redaction@programmez.com

Directeur de la Rédaction : Jean Kaminsky. **Rédacteur en Chef :** François Tonic - ftonic@programmez.com. **Ont collaboré à ce numéro :** F. Mazué, S. Saurel, Y. Grandmontagne, T. Lellouche. **Experts :** A. Lemar-Verrier, C. Villeneuve, J. Dollon, B. Boucard, G. Bougenière, C. Pichaud, V. Perdereau, J. Rowe, F. Gonzales

Illustrations couverture :

©Microsoft, D.R, FreeSoftware

Publicité : Régie publicitaire, K-Nova srl.

Pour la publicité uniquement : Tél. : 01 41 77 16 03 - diff@programmez.com. Dépot légal : à parution - Commission paritaire : 0712K78366 ISSN : 1627-0908. Imprimeur : S.A. Corelio Nevada Printing, 30 allée de la recherche, 1070 Bruxelles Belgique. **Directeur de la publication :** J-C Vaudecrane



Méga-sanction pour Megaupload

Nous avons décidé de revenir dans cet édito sur un phénomène de société qui nous concerne, qui vous concerne. Et il pose de vraies questions.

La sanction, aussi brutale qu'inattendue, est tombée le 19 janvier : la justice américaine a ordonné le coupure totale et immédiate des sites liés à Megaupload. Ce site de partage était mondialement connu pour être un lieu de partages de fichiers. On pouvait y trouver les films, logiciels, séries télés, de manière illégale. Les ayants droit américains ont appuyé dessus pour faire sanctionner au prix fort Megaupload.

Jusqu'à présent, le site avait réussi à échapper à tout problème en se protégeant derrière son statut d'hébergeur, non responsable des contenus stockés par les utilisateurs. Cependant, le document officiel de la justice américaine apporte quelques éléments troublants, qui montrent l'intérêt mercantile des fondateurs de megaupload : le chiffre d'affaires de la souscription Premium s'élèverait à 150 millions de dollars, sans compter la publication en ligne (25 millions). D'autres sites de stockage tels que Filesnic, ont suspendu la fonction de partage.

Dans la foulée de l'affaire Megaupload, les Anonymous ont immédiatement riposté en attaquant massivement des sites de l'industrie du divertissement et institutionnels (Universal Music, site de la justice américaine, hadopi.fr, le site de l'Elysée, etc.) et en dénonçant la censure sur Internet.

Mais cette affaire, qui n'est pas la première (rappelez-vous de Piratebay), ni la dernière, pose le problème fondamental de la nature même de ces sites. Le partage est légal mais pas le partage de contenu illicite. Mais après se posent des questions sur les offres numériques des productions de musique, vidéo, films, etc. Les gouvernements multiplient les lois et propositions contre le téléchargement illégal (Hadopi, Acta, Sopa, etc.). Quelle alternative plus attrayante pourrait-on voir fleurir de la part des acteurs du divertissement ? Face au « gratuit » ou presque, comment rester attractif ? Le phénomène est devenu une addiction pour certains et même un phénomène de société.

A l'heure où le stockage et le partage en ligne des documents et des contenus multimedia (musique, vidéo...) se multiplient (Google, Apple, Amazon pour ne citer que les principaux), une refonte du modèle économique et de la manière de consommer la culture est plus que jamais d'actualité. Et cela nous concerne tous. L'Histoire nous apprend que la prohibition totale n'a pas été une solution sur le long terme.

Alea jacta est.

François Tonic
Rédacteur en chef

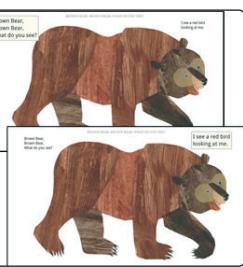
Abonnement : Programmez, 17, Chemin des Boulanger, 78926 Yvelines Cedex 9 - Tél. : 01 55 56 70 55 - abonnements.programmez@groupe-gli.com - Fax : 01 55 56 70 91 - du lundi au jeudi de 9h30 à 12h30 et de 13h30 à 17h00, le vendredi de 9h00 à 12h00 et de 14h00 à 16h30. **Tarifs abonnement (magazine seul) :** 1 an - 11 numéros France métropolitaine : 49 € - Etudiant : 39 € - CEE et Suisse : 55,82 € - Algérie, Maroc, Tunisie : 59,89 € Canada : 68,36 € - Tom : 83,65 € - Dom : 66,82 € - Autres pays : nous consulter. **PDF :** 30 € (Monde Entier) souscription exclusivement sur www.programmez.com



Amazon dévoile son KF8

Amazon complète sa gamme logicielle pour son Kindle. L'éditeur propose un nouveau format numérique dédié aux livres électroniques au format HTML / CSS : KF8. Mais au-delà du format (l'ambition est qu'il soit disponible partout), Amazon veut fournir les outils de conception. Pour cela, les auteurs disposent d'un logiciel auteur graphique pour voir le résultat des conceptions HTML 5 et CSS 3. L'éditeur propose depuis peu Kindle Previewer. Il s'agit d'un logiciel auteur graphique pour prévisualiser les livres Kindle. Cet outil permet de voir son œuvre sur Kindle et les applications dédiées. Mais

vous devrez toujours passer par un outil tiers pour créer votre livre numérique. Un plug-in pour InDesign permet de convertir les fichiers InDesign en fichiers KF8. Même si, comme toute conversion, le résultat n'est pas toujours au rendez-vous. Un travail d'adaptation sera sans doute nécessaire. Pour Amazon, KF8 est un bon format pour des mises en page dynamiques et complexes comme des BD, des manuels techniques. Selon l'éditeur, le format supporte



150 balises et fonctions HTML 5 et CSS3. Il est possible d'embarquer des polices. Les outils Amazon sont disponibles pour OS X et Windows.

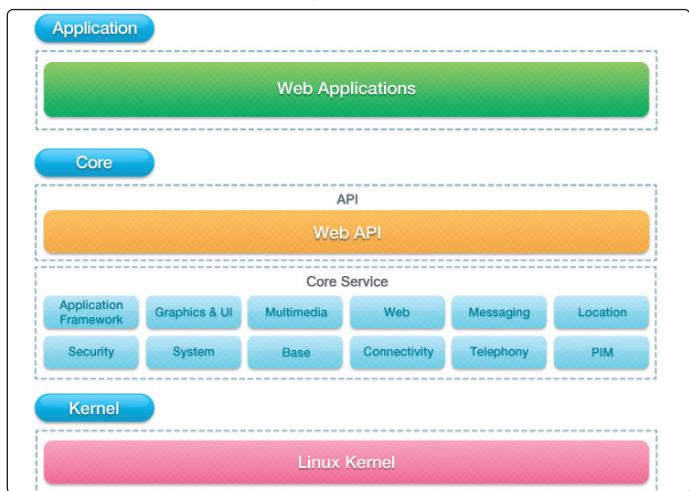
Site : http://www.amazon.com/gp/feature.html/ref=amb_link_359603402_2?ie=UTF8&docId=1000765261&pf_rd_m=ATVPDKIKXODER&pf_rd_s=right-4&pf_rd_r=ONXGQWJH8XQYCOKG4400&pf_rd_t=1401&pf_rd_p=1342417002&pf_rd_i=1000765211

RIM veut relancer Playbook avec la v2



RIM cherche à reprendre une bonne dynamique avec les tablettes. Si la première version a été un flop, le constructeur lance la v2 du matériel et du système. Cette version vise à améliorer le fonctionnement des applications actuelles, simplifier l'impression, offrir une meilleure compatibilité avec Office, la connexion à un téléviseur ou un vidéo projecteur. La partie développement a été elle aussi revue et corrigée. Le développement natif pour Playbook en C et C++ complète sa documentation et rajoute les extensions natives de Adobe AIR et un nouvel assistant de déploiement. Un nouveau SDK fait son apparition : Scoreloop. Il s'agit d'un kit dédié aux jeux. RIM propose aussi un outil pour repackager les applications Android en application Playbook. Cela pourrait permettre d'enrichir rapidement la boutique applicative. Les applications Android seront supportées dans la v2 du Playbook OS (attention à la compatibilité).

Intel et Samsung sortent Tizen



Intel n'est pas décidé à lâcher les systèmes mobiles malgré l'échec du projet commun avec Nokia. Cette fois-ci, Intel s'est associé à Samsung. Le nouveau système s'appelle Tizen et est disponible en open source. Cette annonce a laissé froid de nombreux acteurs du marché. Car aujourd'hui pour espérer faire décoller une plateforme mobile, il faut des constructeurs, des développeurs, et bien entendu, des millions d'utilisateurs. Nokia s'est cassé les dents, Intel aussi. RIM plonge, Microsoft a encore du mal à décoller. Et Samsung collabore à tout ce qui passe. Actuellement, une préversion des outils est disponible : les sources du système, les API, documentations et exemples.

L'architecture de Tizen s'appuie sur un noyau Linux. Les core services regroupent les services essentiels (interface, sécurité, web, réseau, etc.). Au-dessus, nous disposons d'API Web pour les applications s'exécutant sur le système. Ces API utilisent les spécifications W3C, WebGL. Sur l'architecture et l'interface, rien de bien nouveau, en revanche, notons la présence des technologies faisant parler d'elles : NFC, WPS (comme un GPS ou le mode Wi-Fi). Pour la téléphonie, là encore rien d'extraordinaire : UMTS, CDMA, 3G, VoIP. Les applications Tizen sont des widgets HTML/JavaScript/CSS. Le SDK est disponible sur Ubuntu uniquement mais des versions OS X et Windows sortiront sous peu. Et les pré-requis systèmes sont nombreux, espérons que les versions Windows et OS X seront plus « simples ».

A vous de voir maintenant : <https://www.tizen.org/>

Le bon design d'interface pour Android 4

Voilà un document que tout développeur Android devrait lire, relire, appliquer à la lettre : le guide officiel des interfaces Android 4.x. Le document en ligne explique les principes fondamentaux, ce qui est conseillé et déconseillé. Le développeur se verra expliquer les différents éléments d'interface, leur agencement dans une application et leur bon fonctionnement. Un document très complet !

site : <http://developer.android.com/design/index.html>



**ActiveReports 6**

à partir de € 520

GrapeCity

Dernière version du générateur de rapports .NET sans droits le plus vendu.

- Moteur de génération de rapports rapide, souple avec licence sans droits pour le Web/Windows
- Visualisation des données et contrôles de mise en forme: Chart, Barcode et Table Cross Section
- Prise en charge de Medium Trust dans l'environnement ASP.NET
- Vaste gamme de formats d'exportation/aperçus: Windows Forms Viewer, Web Viewer, Flash & PDF
- Contrôle utilisateur de conception avec API souple et personnalisation

**Janus WinForms Controls Suite V4.0**

à partir de € 689

Janus
systems

Ajoutez des interfaces de style Outlook à vos applications .NET.

- Vues ruban, grille, calendrier, et barres chronologique/raccourcis
- Nouveau – Style visuel Office 2010 pour tous les contrôles
- Nouveau – Support des profils client Visual Studio 2010 et .NET Framework
- Janus Ribbon ajoute Backstage Menus et la fonctionnalité onglet comme dans Office 2010
- Prend désormais en charge la sélection de cellules multiples

**DXperience DXv2**

à partir de € 1 116

DevExpress

Tous les outils DevExpress ASP.NET, WinForms, Silverlight, WPF et IDE Productivity en un.

- Abonnement de 12 mois pour tous les produits et mises à jour DevExpress et accès aux versions bêta en développement actif
- Exploitez vos compétences pour développer des applications tactiles sur toutes les plateformes
- Créez des applications métier avec l'esthétique Windows 8 Metro
- Inclut des fonctions d'aide décisionnelle, de rapport et d'analyse

**TX Text Control .NET for Windows Forms/WPF**

à partir de € 911

TX
CONTROL

Composants complets de traitement de texte pour Windows Forms et WPF.

- Le traitement de texte professionnel pour vos applications
- Chargez, modifiez et enregistrez au format MS Word (DOCX, DOC, RTF)
- WYSIWYG, tableaux imbriqués, cadres, en-têtes, pieds de pages, images, puces, listes numérotées, zoom, sauts de section, etc.
- Exploitez la liaison de données XAML pour lier les barres d'outils et de ruban

1&1 HÉBERGEMENT WEB



« Quelles que soient les perspectives de développement de votre activité, avec 1&1 vous êtes toujours entre de bonnes mains. Aujourd'hui comme demain. »

Stefan Mink,
Responsable de centres de données 1&1

Le choix qui s'impose pour votre réussite en ligne :

- ✓ Disponibilité maximale :
Hébergement simultané dans 2 centres de données distincts
- ✓ Rapidité exceptionnelle :
Connectivité de 275 Gbits/s
- ✓ Hébergement vert :
Energie renouvelable
- ✓ Innovation permanente :
1000 développeurs en interne

The screenshot shows a web browser displaying the 1&1 Dual Hosting landing page. The page features a large yellow '&' symbol and the text '1&1 DUAL HOSTING'. Below this, it says 'Votre site doublement sécurisé grâce à un hébergement dans 2 centres de données distincts.' On the left, there's a sidebar for selecting a platform: 'Linux' (selected) or 'Windows'. The main area shows four hosting plans:

- 1&1 DUAL ESSENTIEL**: 6 MOIS GRATUITS*, 2 Go d'espace disque, 100 Go trafic, 1 base de données MySQL, 25 applications Click & Build, 10 Outils de statistiques et de référencement, PHP5, Perl, Python, Zend Framework.
- 1&1 DUAL CLASSIQUE**: 6 MOIS GRATUITS*, 10 Go d'espace disque, 100 Go trafic, 10 bases de données MySQL (1 Go), 65 applications Click & Build, 10 Outils de statistiques et de référencement.
- 1&1 DUAL AVANCE**: 3 MOIS GRATUITS*, 20 Go d'espace disque, 200 Go trafic, 20 bases de données MySQL (1 Go), 100 applications Click & Build, 20 Outils de statistiques et de référencement.
- 1&1 DUAL ILLIMITÉ**: 3 MOIS GRATUITS*, 50 Go d'espace disque, 500 Go trafic, 50 bases de données MySQL (1 Go), 200 applications Click & Build, 50 Outils de statistiques et de référencement.

1&1 DOMAINES

- 5 Mo d'espace disque
- 1 compte email de 2 Go
- 1 site de 3 pages
- Gestion DNS
- 1000 alias email
- Transfert de domaine

à partir de
.fr **3,99**
.eu
.com

€ HT/an
(4,77 € TTC/an)
la première année*

1&1 DUAL CLASSIQUE

- 2 noms de domaine inclus
- 100 Go d'espace disque
- Trafic ILLIMITÉ
- 10 bases de données MySQL (1 Go)
- 65 applications Click & Build
- Outils de statistiques et de référencement
- PHP5, Perl, Python, Zend Framework

6 MOIS GRATUITS

puis 4,99 € HT/mois
(5,97 € TTC/mois)*



* Offre 6 mois gratuits : engagement de 12 mois. À l'issue des 6 premiers mois, le pack Dual Classique et l'e-Boutique Classique sont à leur prix habituel respectif de 4,99 € HT/mois (5,97 € TTC/mois) et 19,99 € HT/mois (23,91 € TTC/mois). Frais de mise en service respectifs : 4,99 € HT (5,97 € TTC) et 9,99 € HT (11,95 € TTC). Offres sans durée minimum d'engagement également disponibles. Offres domaines : pendant la 1ère année, les noms de domaine (.fr, .eu, .com) sont à partir de 3,99 € HT/an (4,77 € TTC/an) au lieu de leur prix habituel 6,99 € HT/an (8,36 € TTC/an). Offres à durée limitée, conditions détaillées sur 1and1.fr.

HÉBERGEMENT HAUTE FIABILITÉ

6 MOIS GRATUITS*

VALABLE JUSQU'AU 29/02/12 !

1&1 E-BOUTIQUE CLASSIQUE

- Votre boutique en ligne rapidement
- 1000 articles, 100 catégories
- Vos produits sur eBay et Kelkoo
- Design : 145 modèles personnalisables
- Outils de statistiques et de référencement
- Paiement sécurisé PayPal
- Gestion simple des commandes et livraisons

**6 MOIS
GRATUITS**

puis 19,99 € HT/mois
(23,91 € TTC/mois)*

1&1, fournisseur de solutions Web :

- ✓ Domaines
- ✓ Hébergement
- ✓ Sites Web
- ✓ Serveurs
- ✓ e-Commerce
- ✓ Solutions email
- ... et bien plus encore !

1&1

Appelez le **0970 808 911** (non surtaxé) ou consultez notre site Web

www.1and1.fr



» Mozilla lance la version Firefox ESR (Extended Support Release). Cette version se dédie aux entreprises ayant une gestion centralisée de Firefox. La fondation annonce que cette version sera mise à jour une fois par an. Elle sera plus stricte sur les évolutions pour ne pas bouleverser les habitudes et la compatibilité en entreprise.

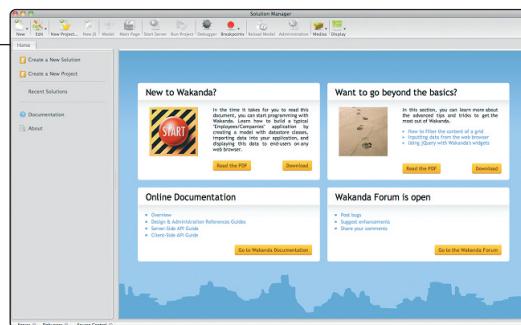
Annonce officielle :
<http://blog.mozilla.com/blog/2012/01/10/delivering-a-mozilla-firefox-extended-support-release/>



» Amazon Web Services

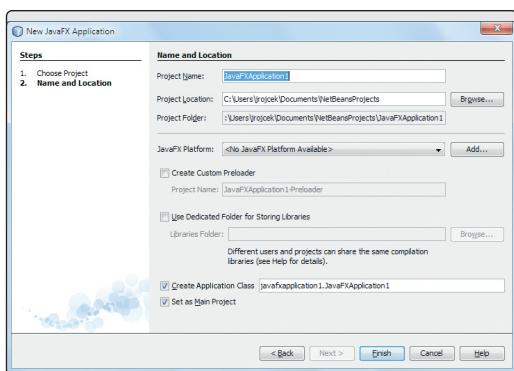
annonce la disponibilité des instances sous Windows Server en usage libre. Cette fonction était jusqu'à présent limitée à Linux. Les utilisateurs des applications Windows Server comme les développeurs et les entreprises bénéficient ainsi d'une utilisation annuelle gratuite de micro-instance de 750 heures du service

Amazon Elastic Compute Cloud (Amazon EC2) par mois. Grâce à cette extension du module d'usage libre d'AWS, les développeurs de Windows ont désormais accès aux mêmes ressources libres sur AWS que les développeurs de Linux, et ont ainsi la possibilité de développer de nouvelles applications, de tester les applications existantes dans le nuage, ou d'acquérir une expérience pratique avec AWS. C'est une bonne nouvelle. Car AWS avait l'habitude de faire payer plus cher Windows Server que Linux. Pour en savoir plus : <http://aws.amazon.com/windows/>



» 4D investit HTML 5 avec son outil Wakanda qui est disponible en préversion depuis fin décembre dernier. Il s'agit d'une plateforme open source « end-to-end » de développement d'applications professionnelles pour le Web, le cloud et les plateformes mobiles, entièrement en HTML5 et en JavaScript. Une application Wakanda peut être déployée sur un cloud. Les principales fonctionnalités sont : orienté modèle, sup-

port développement web et mobile, NoSQL, environnement visuel, open source et extensible. Site : <http://www.wakanda.org/>



» Netbeans 7.1 est disponible depuis début janvier. Pour rappel cette version supporte JavaFX 2, un nouveau debug, CSS 3, meilleur support de Glassfish, nouvelles API Windows, etc. La version 7.1.1 (maintenance) sera disponible en mars prochain. La prochaine version majeure apparaîtra en juin/juillet. Cette version se focalisera sur les performances (éditeurs, refactoring). Tous les détails ne sont pas encore dévoilés. Site : http://wiki.netbeans.org/NetBeans_72

» Micro Focus dévoile Borland StarTeam 12. StarTeam est l'environnement de gestion de cycle de vie des applications de Micro Focus. Borland StarTeam 12.0 connecte automatiquement les développeurs et leur code à n'importe quel outil de gestion des changements, afin que les équipes puissent collaborer totalement au processus de livraison logiciel destiné aux utilisateurs finaux. Parmi les nouveautés, notons : synchronisation avec Eclipse Mylyn, traçabilité améliorée des éléments projets, des workflows développés plus visibles. Site : <http://www.borland.com/us/products/starteam/>

» Eclipse va-t-il devenir l'IDE le plus hétéroclite du monde pour les langages ? Possible. En effet, durant la prochaine EclipseCon, il sera beaucoup question de l'intérêt et de la possibilité de supporter les nouveaux langages tels que Dart, Lua, Ceylon, Xtend. La question centrale est : pourquoi supporter ces langages ? EclipseCon se déroulera du 26 au 29 mars prochain à Reston (USA).

» Oracle dévoile une mise à jour de sécurité très complète (CPU) pour janvier 2012 (disponible) : presque 80 failles corrigées. Tous les produits Oracle sont concernés, dont VM VirtualBox, MySQL, Database 11g, WebLogic... Rien que sur MySQL, ce sont 27 fixes de sécurité. Sur Java, Oracle a sorti une mise à jour de JavaFX (2.0.2). Site : <http://www.oracle.com/technetwork/topics/security/alerts-086861.html>

» VLC arrive en version 2.0. Anciennement numérotée 1.2, la v2 rajoute beaucoup d'améliorations et de modifications : un cœur vidéo optimisé et plus performant sur la 3D, le traitement des shaders. Sur l'affichage 3D des vidéos, ce ne sont que les premiers pas. De nouveaux filtres vidéos sont disponibles ainsi que de nouveaux codecs et formats (blu-ray, protocoles HLS et DASH). Sur les codecs, le lecteur supporte le décodage multi-coeur. Bonne nouvelle aussi sur l'interface qui subit une modification assez profonde sur la partie OS X et web. On parle déjà d'une v3 courant 2013. A suivre... 

» **Google** a sorti App Engine (PaaS) de sa longue période bêta. App Engine passe donc en production officielle chez Google. Actuellement, nous sommes en version 1.6.x. Cette version introduit plusieurs changements : classes instances, outil de réPLICATION pour les dataSTORE, API de conversion de format, API de lecture des logs. App Engine supporte toujours Java et Python mais on dispose désormais d'un SDK Go (langage maison). Côté tarification, App Engine s'aligne sur les grilles AWS et Microsoft : paiement par forfait, formule gratuite, paiement selon les API utilisées et les fonctions. Ainsi, côté facturation, il faudra faire attention aux coûts des instances, du stockage, lecture / écriture, etc. (<http://code.google.com/intl/en/appengine/docs/billing.html>). L'offre Free reste gratuite avec les quotas à surveiller. L'offre Paid revient à 9 \$ par mois / application et les quotas sont identiques à ceux de l'offre Free, mais on y gagne le SLA, le support. L'offre Premier revient à 500 \$ par compte et par mois. SLA : d'une manière générale le taux de disponibilité est de 99,95 % [global] en niveau premiumN (<http://code.google.com/intl/en/appengine/sla.html>). Sur les comptes premiers : support premium avec SLA, pas de limite du nombre d'applications, pas de coût minimum, paiement mensuel par facturation. Reste à voir comment Google App Engine va pouvoir capter du marché car les PaaS Java sont assez nombreux : VMware, Salesforce.com, Windows Azure pour citer les principaux.

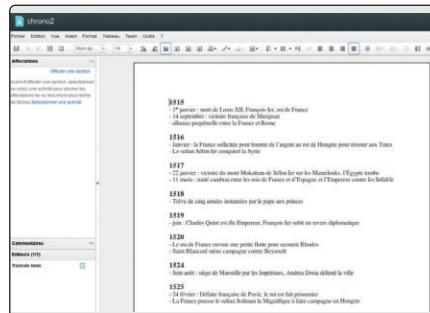
» Si **Microsoft** travaille activement à Hadoop pour Azure, une équipe de Microsoft Research travaille sur un autre projet, tout aussi important : Daytona. Il s'agit d'une implémentation, sous forme d'un runtime à déployer sous Azure, de MapReduce. MapReduce est à la base de Hadoop et est capable de distribuer les données sur des centaines de serveurs pour traiter, analyser les données. Ce projet rentre dans le cadre d'un groupe de



PROGRAMMEZ ! < 11 > FÉVRIER 2012

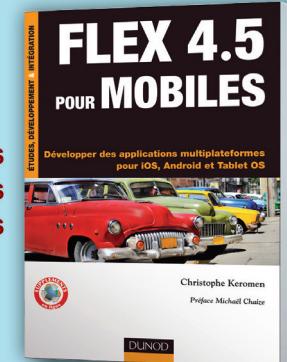
travail sur le cloud interne à Research. Il est possible de développer ses propres algorithmes de calculs qui seront ensuite injectés à MapReduce. Aujourd'hui, on peut utiliser ces algos dans des applications clientes simples, en attendant une intégration aux services BI (par exemple). Les priorités de Daytona sont : - conçu pour le cloud aussi bien que pour un IaaS ou un PaaS. Le but est de garantir une bonne utilisation des machines virtuelles / instances
- utiliser au mieux les ressources de stockage tout en minimisant les effets du overhead
- utiliser en temps réel les nouvelles ressources provisionnées pour Daytona
L'équipe Daytona espère que ce projet sera mis en production et proposé sur Windows Azure. Site : <http://research.microsoft.com/en-us/projects/daytona>

» **IBM** veut s'imposer comme éditeur SaaS. Si l'éditeur possédait déjà des



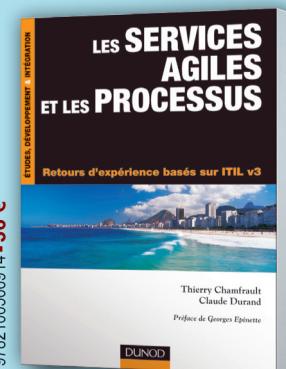
services de collaboration et communication avec Lotus Live, l'éditeur passe à une autre dimension avec IBM Docs. Docs regroupe le traitement de texte, le tableur, un outil de présentation, des fonctions de partage, d'équipes et de commentaires. Clairement, il s'agit d'une sorte de suite bureautique Symphony en mode SaaS pour concurrencer Office 365 et Google Docs, même si sur la partie communication, IBM Docs n'est pas aussi complet. Le projet était connu sous le nom "Lotus Live Symphony". L'interface se veut sobre mais complète. Le service est disponible sur le portail des services SaaS d'IBM : greenhouse. Actuellement en préversion, rien n'est connu sur la disponibilité finale et la tarification. La suite supporte les fichiers Microsoft Office mais attention, certaines fonctions et enrichissements ne sont pas supportés par IBM Docs. Inscription obligatoire. Site : <https://greenhouse.lotus.com/>

DÉVELOPPEZ L'AGILITÉ ET LA MOBILITÉ



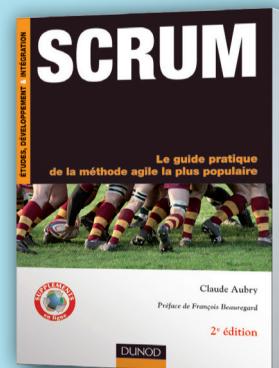
Créez des applications multiplateformes

9782100567065 • 29 €



9782100566914 • 36 €

La boîte à outils pour une gestion des services efficace



Le guide pratique de la méthode agile la plus populaire

9782100563203 • 29,50 €



Concevez des applications web, logicielles, mobiles et tactiles

Réalisation MATEO

Retrouvez-nous du 7 au 9 février sur les **tech.days**

www.dunod.com





Utiliser un bug tracker

Un bug tracker est un logiciel qui permet de tracer les anomalies d'un programme, afin d'améliorer la qualité du développement. Les erreurs identifiées sont qualifiées (bloquant, mineur...) et suivent un cycle de vie qui détermine si l'erreur va être corrigée bientôt (ou pas).

Le bug tracker accompagne le logiciel dès les premières livraisons, et ce quel que soit l'environnement, aussi bien dans les phases de test, recette, que de production. Avant que le logiciel ne soit en production, les développeurs et les testeurs peuvent tracer les anomalies dans le bug tracker. Après la livraison, la hotline peut s'en servir pour recenser les anomalies et suivre leur évolution.

> Qui est intéressé ?

Tous les acteurs du projet peuvent utiliser l'outil de suivi des bugs.

- Le testeur / le service de Hotline :
 - Il crée les anomalies dans l'outil, suit leur évolution, donne des compléments d'information.
- Le développeur :
 - Il sait sur quelles anomalies travailler et dispose de toutes les informations nécessaires pour reproduire le bug.
 - Il ajoute la solution trouvée et les composants techniques impactés par la correction.
 - Inversement, dans son code, il peut indiquer le numéro du bug qui a nécessité la modification.
 - Il peut retrouver une anomalie qui avait déjà été traitée, et ainsi voir quelle solution avait été mise en place.
- Le chef de projet :
 - Il peut effectuer un reporting sur les erreurs corrigées, celles qui sont en cours de correction ou celles qui seront corrigées pour la prochaine version.
 - Il peut en déduire un état d'avancement de la recette ou définir des marqueurs de

The screenshot shows a detailed view of a bug report in Bugzilla. The URL in the address bar is https://bugzilla.mozilla.org/show_bug.cgi?id=169837. The page title is "Bug 169837 - scroll bookmarks but not other menu items in bookmarks menu - Mozilla". The main form fields include:

- Bug ID:** 169837 alias: []
- Product:** Firefox
- Component:** Bookmarks
- Status:** NEW
- Resolution:** []
- Assigned To:** Vladimir Vukcevic (Bookmarks Bugs Only) <vladimir+bm@vad1.com>
- QA Contact:** mconnor@steelfiglyphon.com
- URL:** []
- Summary:** scroll bookmarks but not other menu items in bookmarks menu
- Status Whiteboard:** []
- Keywords:** []
- Hardware:** PC
- OS:** All
- Version:** unspecified
- Priority:** P4
- Severity:** normal
- Reporter:** Asa Dotzler <asa@mozilla.org>
- CC:** bugs@bengodger.com, bugzilla3@yahoo.gr, bugzilla@waruno.com, bugzilla@spray.se, chris.biore@gmail.com
- Last modified:** 2005-05-29 11:57 PDT
- Flags:** blocking1.8b4, blocking1.9a1, asa: blocking-aviary1.0, blocking-aviary1.0.6, mconnor: blocking-aviary1.1, blocking-aviary2.0

Below the form, there's a table for attachments, a dependency graph, and a section for additional comments.

qualité du logiciel. Par exemple : en comptabilisant le nombre de bugs bloquants ou majeurs, le chef de projet peut déterminer si le logiciel est prêt à être livré.

- Il peut aussi suivre les anomalies sur lesquelles travaillent les membres de son équipe et définir ainsi des tâches à effectuer, avec des notions de priorités.

> Quels outils ?

La liste des logiciels est longue, l'objectif n'est pas ici de comparer l'ensemble des outils de suivi d'anomalies du marché. Le choix d'un outil dépend des caractéristiques et des contraintes du projet, des normes de qualité de l'entreprise, de la volonté d'un outil unique ... Ci-dessous, quelques pistes de réflexion pour choisir un outil avec des exemples associés.

- **Le choix de l'outil dépend du budget :** par exemple, Mantis Bug Tracker (<http://www.mantisbt.org/>) est gratuit, Bugzilla (<http://www.mozilla-europe.org/fr/products/bugzilla/>) aussi. JIRA est un outil gratuit pour les projets open source approuvés, payant sinon (<http://www.atlassian.com/fr/software/jira/overview>).
- **Le choix de l'outil dépend de contraintes techniques,** telles que : la base de données où sont stockés les bugs (MySQL, Oracle ...), l'identification des intervenants via un annuaire d'entreprise, l'envoi de

mails en automatique. L'outil Redmine (<http://www.redmine.org/>) permet par exemple un accès à un LDAP, et des notifications automatiques par mail.

- **L'outil doit permettre de gérer d'autres éléments clés du projet en plus du suivi d'anomalies :** comme les plans de tests, des tableaux de bord ... 2 plateformes correspondent parfaitement à ce besoin :
 - Codendi (<http://www.codendi.com/>) est un outil collaboratif open source entièrement paramétrable, il dispose des fonctionnalités de gestion de projet, de workflow de documents, des outils de collaboration (forums, flux RSS...).
 - HP Quality Center (<http://www8.hp.com/fr/fr/software/software-product.html?compURI=tcm:113-937045&pageTitle=quality-center>) propose de tracer les exigences, les plans de tests, de lier les anomalies aux tests et à la gestion des versions.

> Que saisir dans une anomalie ?

Voici des informations qu'on retrouve dans un bug tracker :

- **Le titre :** pour résumer l'anomalie, il doit être explicite.
- **L'auteur :** normalement renseigné en automatique et non modifiable, il permet de savoir à qui s'adresser s'il y a des questions sur l'anomalie.

- La date** : pour éventuellement corrélérer l'anomalie à un problème de plateforme ou de version
- L'affectation** : il peut s'agir d'un service, ou d'une personne. Ce champ correspond à l'acteur qui est en charge de l'anomalie, il sera modifié à chaque nouvelle étape du cycle de vie.
- La sévérité** : bloquant si une correction rapide est à programmer; majeur si l'anomalie peut attendre la prochaine livraison, mineur, cosmétique ... Indispensable pour prioriser la résolution de l'anomalie.
- Le statut** : bug, évolution, doublon. Ce statut peut être modifié au cours de l'étude de l'anomalie. Quand l'anomalie est créée, le testeur pense qu'il s'agit d'un bug. Mais, s'il s'avère finalement que ce point n'a pas été spécifié complètement, l'anomalie pourra être passée en statut « Evolution ». Une anomalie peut aussi avoir été détectée plusieurs fois, dans ce cas le statut « Doublon » pourra être utilisé.
- La version de détection** : pour donner la version du logiciel sur laquelle l'anomalie a été décelée.
- La version planifiée** : pour définir dans quelle version sera la correction. Ce champ peut être utilisé pour filtrer toutes les anomalies corrigées dans une version.
- Le cycle de vie d'une anomalie** : Nouveau,

Ouvert, Fermé, Rouvert, Annulé, Corrigé mais non livré ... Un exemple sera détaillé plus bas.

- L'environnement** : pour savoir dans quel environnement est apparue l'anomalie (en production, en recette ...)

- La description détaillée** : pour saisir l'enchaînement des actions qui ont conduit à l'erreur, noter les paramètres, le message d'erreur, le cas de test associé, tout élément permettant de reproduire l'anomalie.
- Les fichiers attachés** : une copie d'écran de l'erreur, des fichiers de données d'entrée, de logs ...

Les logiciels peuvent disposer d'autres champs en standard, ou bien proposer de configurer les champs. Ces champs spécifiques peuvent être utiles pour générer des statistiques propres à l'entreprise, pour cibler les anomalies sur un domaine fonctionnel, sur une version...

Suivons le cycle de vie d'une anomalie :

- Un testeur découvre un bug, il crée une nouvelle anomalie, avec un état Nouveau et affecte l'anomalie au service de développement.
- Le chef de projet affecte l'anomalie à un développeur de son équipe, et passe le statut à Ouvert.
- Le développeur corrige l'anomalie, il positionne l'état à résolu, mais non livré.

Et voilà, l'anomalie est résolue mais non livrée.

- La correction est déployée sur l'environnement de test, l'état peut être positionné à Livré.

- Le testeur peut alors valider la correction. Si le test est concluant, l'anomalie sera fermée, sinon elle pourra être ré-ouverte et le cycle de vie recommencera.

Chaque outil dispose de son propre cycle de vie. Certains logiciels ont la possibilité de le paramétriser pour qu'il coïncide parfaitement avec la gestion de l'entreprise.

Par exemple, le cycle de vie peut intégrer la livraison dans chaque environnement : livré en environnement de développement, en recette métier, en production ...

Les changements clés de l'anomalie peuvent être his-

torisés (le changement d'affectation, de cycle de vie) afin de la suivre au mieux.

> Quelques pièges ...

Le bug non détaillé

« ERREUR FATALE, CA NE MARCHE PAS !!! »

Oui, c'est sûr, on est énervé parce que l'application plante, mais il faut penser au développeur qui essaye de reproduire le bug, et qui a (vraiment) la volonté de le corriger.

Avec si peu d'information, c'est un vrai défi ! Et il ne faut pas oublier non plus le chef de projet qui désespère de montrer dans ses reporting la (les) fameuse(s) erreur(s) « ça ne marche pas » dont le titre n'est vraiment pas significatif.

Le bug tracker donne la possibilité de détailler l'anomalie, de mettre une description précise, d'ajouter une copie d'écran.

Celui qui crée l'anomalie doit être suffisamment précis, ajouter l'enchaînement des actions, les données en entrée pour que celui qui corrige l'anomalie la reproduise aisément. Mettre le libellé exact de l'erreur peut par exemple permettre au développeur de trouver l'emplacement exact dans le code. De même, le titre de l'anomalie doit être précis et pertinent pour aider à la recherche, et pour avoir une vraie signification dans un reporting.

Le bug aléatoire

Cette fois c'est bon, le bug est détaillé, l'enchaînement des actions est précisé, et on a même ajouté une copie d'écran quand l'erreur apparaît. Parfait. Mais non, le développeur a beau répéter le scénario décrit, il ne reproduit pas l'erreur. Que faire ? Supprimer le bug ? Surtout pas, le problème n'est pas résolu, il peut réapparaître, il serait dommage de perdre cette description si détaillée ... Fermer le bug ? Oui, si on peut le rouvrir ensuite. Il se peut que le bug apparaisse plus fréquemment, qu'on comprenne un détail supplémentaire qui permettra sa résolution. Ou au contraire le bug correspond peut-être à une erreur temporaire d'environnement, ou à un problème d'installation résolu depuis. Dans ce cas, le bug restera fermé et pourra être définitivement oublié.

Le bug oublié

Pour que les choses soient claires : au début, celui qui le crée ne veut pas qu'il soit oublié ! C'est un bug souvent mineur, il n'a pas été affecté à une version, ou à un service, il est seul. Si on oublie de renseigner la version dans laquelle le bug doit être corrigé,

The screenshot shows a detailed view of a bug entry in the Codendi Partners system. The task is titled "Administration: account creation". Key details include:

- Description:** "Administration account creation" (with a note: "Provide administrative interface for account creation. This should only be accessible to platform administrators")
- Status:** "Ouvert" (Open)
- Follow-ups:** A list of comments from users Bob Brown and John Smith, with their names and timestamps.
- Attachments:** A section for uploaded files, currently empty.
- File Description:** A table showing file details like name, type, size, and upload date.
- Dependencies:** A list of dependencies with their status.
- References:** A list of artifacts referenced by the bug.
- Change History:** A history of changes made to the bug.



The screenshot shows the Mantis Bug Tracker interface. At the top, there's a banner for 'FastBugTrack' with the text 'Start Tracking in 5 Minutes! Free Demo Here!' and a link to 'www.fastbugtrack.com'. Below the banner, the main menu includes 'Anonymous', 'Login', 'Signup for a new account', 'Main', 'My View', 'View Issues', 'Change Log', 'Summary', and 'Docs'. A search bar and a 'Jump' button are also present.

The main content area displays 'Viewing Issue Advanced Details' for issue ID 0005926. The table shows the following information:

ID	Category	Severity	Reproducibility	Date Submitted	Last Update
0005926	[mantisbt] administration	minor	always	07-12-05 12:46	07-12-05 12:46
Reporter	Paul_Berg	View Status		public	
Assigned To					
Priority	normal	Resolution	open	Platform	
Status	new			OS	
Projection	none			OS Version	
ETA	none	Fixed in Version		Product Version	1.0.0a3
				Product Build	

Summary: 0005926: Filter on checkbox field
Description: I have created a custom field of the type "checkbox" with 3 possible values A, B and C. When I filter on this field Mantis gives me 5 options to choose from: any, none, A, B, C. If I choose "none" I'd expect Mantis to show me all issues for which none of the 3 checkboxes are set, instead it never shows me any issues.
 Am I doing something wrong, or is this a bug?
 Paul

Steps To Reproduce:
Additional Information:
Fixed in Release:
Attached Files:

Relationships:

There are no notes attached to this issue.

ou la personne en charge de bug, il se peut très bien que le bug n'apparaisse pas dans les filtres des chefs de projets ou des développeurs. Il est très important de renseigner les champs de paramétrage utilisés par les chefs de projets, et les développeurs pour qu'ils soient correctement pris en compte. Parfois, au bout de mois et de mois d'attente, le bug n'est toujours pas corrigé. Il faut sans doute s'interroger sur la pertinence de ce bug : « *est-ce qu'il est vraiment important, est-ce que beaucoup d'utilisateurs l'ont signalé ou est-ce qu'il ne dérange que moi ?* ». En fonction des réponses à ces questions, le bug pourra être affecté à une priorité plus forte ou au contraire être annulé, et cette fois il sera oublié pour de bonnes raisons.

Le bug répété, mais pas par la même personne

C'est un bug qui embête tout le monde, les testeurs, les clients. Chacun ajoute son anomalie dans le logiciel, mais finalement le problème est unique. Il y a deux points de vue à concilier : celui du chef de projet « *il n'y a qu'un SEUL bug à corriger* » et celui qui a créé l'anomalie « *je veux être averti de la résolution de MON bug* ». Au niveau du reporting, il est essentiel de montrer qu'il s'agit d'une unique correction, mais qu'elle impacte beaucoup d'utilisateurs. Elle doit donc être positionnée avec une priorité forte. Quand le programme sera corrigé, chacun devra être averti, il ne faut donc pas

fermer tous les bugs pour la simple raison : « *quelqu'un d'autre l'a trouvé avant toi !!!* ». Si ce sont les membres d'une même équipe qui ont créé chacun une anomalie, il est possible de ne conserver qu'un seul bug, en communiquant sur le bug de référence. Par contre, s'il s'agit de clients différents, la fermeture de l'anomalie devra attendre la correction pour que chaque client puisse être averti personnellement.

Il peut exister plusieurs solutions à ce type de problème, qui dépendent du logiciel de suivi de bug. Par exemple : positionner un paramètre « *Doublon* », établir un lien entre tous les bugs vers un bug de référence ...

Le bug corrigé à moitié

Le bug est noté corrigé, la livraison est effectuée, mais on a beau tester, seul le premier point décrit dans le bug est corrigé et pas les suivants. La gestion de l'anomalie peut alors devenir complexe : le testeur n'est pas d'accord pour fermer ce bug qui n'est pas complètement corrigé, le développeur n'est pas d'accord pour rouvrir le bug car il en a corrigé une partie ... Mais comment en est-on arrivé là ? Le développeur a pu oublier de corriger certaines anomalies (cachées en bas de la page), ou n'a eu le temps nécessaire que pour corriger le premier point et les autres points ont été planifiés pour une future version. Pour éviter ce genre de problèmes, l'idée est de ne pas décrire trop de points dans un même bug et de préférer

créer plusieurs bugs. Chaque anomalie aura ainsi son propre cycle de vie et pourra être affectée à la bonne version.

Les mails automatiques

Les outils de bug tracking peuvent envoyer des mails automatiques lors des modifications de l'anomalie. Si le mail est envoyé à la moindre modification de l'anomalie, chacun est vite surchargé par les mails. On corrige une faute d'orthographe, un mail part, on avait juste oublié de saisir le numéro de version, un autre mail est envoyé, on ajoute un paramètre, et voilà déjà un troisième mail pour une unique anomalie. Finalement quand arrive le mail automatique décisif « *alerte rouge, l'anomalie est devenue bloquante* », il est noyé parmi tous les mails automatiques du bug tracker. Il peut même être classé automatiquement dans un dossier « *les mails du bug tracker* » avec une règle de gestion de la messagerie. Le bénéfice de l'envoi automatique est alors perdu. Si l'envoi en automatique est paramétrable, il est nécessaire de bien définir les critères d'envoi du mail pour ne pas surcharger les utilisateurs de mails (changement de priorité, d'affectation...). S'il n'est pas paramétrable, en cas de modification majeure, il faudra souvent communiquer en plus pour alerter du changement (en pratique : envoyer un nouveau mail...).

> Finalement ?

Le bug tracker est un outil **indispensable** dans la gestion de projets informatiques pour assurer un logiciel de qualité : il permet de suivre les anomalies avant la livraison, et les corrections à effectuer après la mise en production. Chaque acteur du projet contribue à l'enrichissement des données, et bénéficie de l'outil pour s'organiser ou connaître l'état d'avancement de ses demandes.

Chacun peut avoir, à la fois une vision détaillée des anomalies qui le concernent et une vision d'ensemble de l'état d'aboutissement du logiciel. Le choix du logiciel est déterminé en fonction des volontés ou des contraintes de l'entreprise, l'important est de pouvoir tracer et suivre l'anomalie tout au long de son cycle de vie. Les pièges indiqués sont juste anecdotiques, clin d'œil à des projets sur lesquels l'outil de bug tracking était un véritable atout qualité...



Aude Lemar-Verrier
*Pôle Assistance
 à Maîtrise d'Ouvrage
 Objet Direct*
www.objetdirect.com

1&1 SERVEURS DERNIÈRE GÉNÉRATION



**EXCLUSIVITÉ 1&1 :
2 x 16 CŒURS**
PROCESSEUR AMD OPTERON™ 6272

✓ **Fiabilité :**
Centres de données ultra-modernes, disponibilité de plus de 99,9 %

✓ **Tout inclus :**
Parallels® Plesk Panel 10.4 illimité, SSL, nom de domaine

✓ **Flexibilité :**
Systèmes d'exploitation et fonctionnalités au choix

✓ **Rapidité :**
Trafic illimité et connectivité de plus de 275 Gbits/s

✓ **Des questions ?**
Hotline non surtaxée et assistance par email

Découvrez tous les serveurs 1&1 sur notre site Web.

NOUVELLE GAMME DE SERVEURS DÉDIÉS 1&1 :

NOUVEAU : SERVEUR 1&1
AVEC PROCESSEUR INTEL® !

SERVEUR 4i



- Intel® Xeon® E3-1220
- 4 cœurs jusqu'à 3,4 GHz
- 12 Go ECC RAM
- 1000 Go RAID 1 avec 2 x 1000 SATA HDD

69,99 €
HT/mois*
(83,71 € TTC/mois)

VALIDABLE JUSQU'AU 29/02/12 !

SERVEUR XL 6



- AMD Hexa-Core
- 6 cœurs jusqu'à 3,3 GHz
- 16 Go ECC RAM
- 1000 Go RAID 1 avec 2 x 1000 SATA HDD

**3 MOIS À
0€***

99,99 €
HT/mois*
(113,59 € TTC/mois)

EXCLUSIVITÉ 1&1 :
LE SERVEUR LE PLUS PUISSANT !

SERVEUR XXL 32 CORE



- 2 x AMD Opteron™ 6272 (Interlagos)
- 2 x 16 cœurs jusqu'à 3,0 GHz
- 64 Go ECC RAM
- 2400 Go RAID 6 avec 6 x 600 SAS HDD

399,99 €
HT/mois*
(478,39 € TTC/mois)

1&1



Contactez-nous au **0970 808 911** (non surtaxé) ou consultez notre site Web

www.1and1.fr

* Offre Serveur XL 6 : 3 mois à 0 € puis 99,99 € HT/mois (113,59 € TTC/mois). Offre soumise à un engagement de 12 mois et à des frais de mise en service de 99 € HT (118,40 € TTC). Offre sans durée minimum d'engagement également disponible. Serveurs 4i et XXL 32 Core : frais de mise en service de 49 € HT (58,60 € TTC) pour le Serveur 4i et de 99 € HT (118,40 € TTC) pour le Serveur XXL 32 Core. Pas de frais de mise en service en cas d'engagement de 12 mois. Conditions détaillées sur 1and1.fr.

Spécial CES 2012

Le futur de l'électronique et de l'informatique

Comme chaque année, le CES, le plus grand salon électronique du monde, a été durant quelques jours le lieu où il fallait être. Microsoft a fait sa dernière prestation officielle, sans aucune annonce, et dans un show oublié dès sa fin. Apple était présent sans l'être, comme chaque année. Que retenir du salon : des tablettes, des smartphones, des ultrabooks et le nouveau gadget à la mode : la télévision connectée !



Le futur de la télévision dès maintenant, ou presque

Depuis quelques mois, la rumeur n'en finit plus et ne risque pas de se dégonfler : Apple prépare une télévision nouvelle génération. Mais d'autres ne veulent pas attendre.



Google veut relancer son système Google TV, avec une v2 plus aboutie. Parmi les constructeurs, citons des écrans signés Sony, LG. Samsung a fait sensation avec une télévision que l'on peut piloter à la voix, par les gestes. Elle intègre aussi la reconnaissance faciale. Mais le constructeur ne s'arrête pas là en intégrant sur d'autres modèles des processeurs double cœur ou encore des dalles oled. Mais la plupart de ces modèles ne seront pas disponibles avant plusieurs mois et tous les prix ne sont pas

encore connus. L'interface est un des défis les plus importants de cette télévision d'où la volonté d'intégrer des technologies gestuelles et vocales, reste à voir leur comportement réel dans une vraie utilisation. Ensuite, la connectivité doit être permanente avec Internet et les sites pour accéder aux fonctionnalités et surtout aux contenus qui demeurent le véritable nerf de la guerre. Sans contenu, aucun modèle ne pourra se vendre !

Ubuntu est aussi dans une logique tous azimuts pour imposer sa distribution et son modèle système avec Ubuntu TV. Il s'agit d'un système embarqué spécifique aux téléviseurs.

Le principal argument : l'open source, bref avoir un système ouvert (ARM et x86). Mais ce n'est pas un argument suffisant pour le grand public. L'important est la fluidité, l'ergonomie et surtout le contenu. Ubuntu TV sera proposé en OEM aux constructeurs. Mais finalement, tout le monde attend

Apple et son hypothétique iTV...

Mini-serveur pour maxi capacités

Le constructeur Xi3 dévoilait ses nouvelles machines ultra-compactes : à peine 10 cm de côté ! Xi3 est spécialisé dans les ordinateurs minuscules et aussi les serveurs. Les Z3RO sont des serveurs consommant 1 watt capables de faire tourner Windows et tout autre système x86. Ils possèdent des ports USB, un port vidéo et audio. Le pro-



cesseur standard est un Athlon Dual Core, avec des options Athlon 3100 et 2000. Le disque dur est un disque SSD (16 Go par défaut). Chaque module coûte 250 dollars. Site : <http://xi3.org>



Les nouveautés d'Intel

Intel a profité du CES pour dévoiler plusieurs prototypes et parler de sa vision. Le fondeur mise sur les smartphones, tablettes et ultrabooks. Sur les deux premiers marchés, il faut reconnaître qu'Intel ne brille pas beaucoup. Sur les ultrabooks, le fondeur fournit des processeurs mais il voudrait imposer son architecture type, alors que les constructeurs PC suivent leurs propres idées pour concurrencer le Macbook Air d'Apple, référence du marché. Pour ces modèles, Intel pousse Ivy Bridge qui arrivera cette année avec en point de mire, Windows 8. Intel a voulu aussi mettre en avant sa boutique d'applications : App-Up. D'autre part, le fondeur a montré un prototype d'ultrabook équipé d'un repose-main tactile pouvant servir d'écran. Cela rappelle la fonction Slideshow démontrée avec Windows Vista qui fut un monumental bide. Si l'idée peut paraître intéressante reste à voir son intégration sur les machines, la consommation de cet écran et surtout son intérêt pour l'utilisateur.

Microsoft®

tech·days

7, 8 et 9 février 2012

**SOYEZ AU CŒUR DU
PLUS GRAND ÉVÉNEMENT
INFORMATIQUE EUROPÉEN !**



→ **6^{ème} édition**
Palais des Congrès Paris

- Pour les professionnels de l'Informatique.
- Pour les décideurs IT.
- Pour les développeurs.

AU PROGRAMME :

Participez à des sessions sur les enjeux actuels de l'informatique, pour tous les métiers : **Cloud Computing**, Intégration, Technologies de développement, Virtualisation, **Mobilité**, Sécurité, **Management**, Décisionnel, ...

3 conférences plénaires
300 sessions techniques
17000 visiteurs
140 exposants
50 communautés
35 000 web-spectateurs
TechDays TV



Inscrivez-vous
et suivez-nous
sur **www.mstechdays.fr**

En partenariat avec : **PROgrammez!**



matériel

Une tablette spéciale gamer

Razer a dévoilé un prototype d'une tablette dédiée aux joueurs, le projet Fiona. Pour ce faire, elle dispose de deux manettes de part et d'autre de l'écran. La tablette utilise un Intel Core i7 pour délivrer la puissance nécessaire. Les jeux PC peuvent fonctionner directement dessus pour éviter de devoir redévelopper des jeux. « Le Projet Fiona est la combinaison de matériels haute performance réalisés par Intel pour le jeu vidéo et d'une tablette spécialement étudiée pour les joueurs sur PC. Cela va permettre aux

joueurs PC d'apprécier les jeux qu'ils préfèrent et les futurs hits à venir sur un appareil nomade très excitant, une tablette, » explique Brad Graff, Director of Partner Marketing, Netbook & Tablet Group chez Intel. Sur le gameplay, les manettes sont utiles pour le reste. Il n'y a plus qu'à convaincre les joueurs de l'intérêt d'une telle machine car les iPad et autres tablettes possèdent déjà de nombreux jeux. Surtout, Fiona ne sera pas disponible avant le 4e trimestre 2012 pour un prix estimé à 900 - 1000 dollars...



Le smartphone devient un APN

Ce n'est plus un mystère, les smartphones sont devenus de véritables APN que des photographes de renom utilisent au quotidien. Et les derniers modèles le démontrent encore. Olloclip dévoile un objectif très complet pour iPhone. On dispose ainsi d'une macro et d'un grand angle. La 3e fonction est l'œil de poisson, le fisheye qui permet de prendre des photos à 180°. Avantage : il se fixe directement sur le bord du téléphone. Bonne idée. L'optique est vendue à 70 \$, un peu cher !

Kinect sur PC !

Microsoft a confirmé l'arrivée sur PC / Windows de Kinect jusqu'à présent limité à la Xbox. Elle sera disponible dès début février. L'orientation pourrait rapidement se faire vers les professionnels mais aussi chez le grand public. Cependant, le prix risque d'être élevé, 250 dollars ! Cette version gère ce que l'on appelle le mode proche car difficile d'être à plusieurs mètres quand on travaille sur PC. Par contre, il semblerait que cette Kinect Windows ne soit pas compatible avec la Xbox...

USA : deux systèmes dominent le marché du smartphone

La dernière analyse du marché des smartphones aux Etats-Unis menée par NPD Group est sans appel : plus que jamais deux systèmes mobiles dominent, Android et iOS. La période octobre - novembre 2011 a été marquée par deux faits : la

remontée spectaculaire des terminaux iOS passant de 26 à 43 % (effet iPhone 4S) et Android qui passe de 60 à 47 %. Ces deux plateformes pèsent 90 % du marché US des smartphones. La concurrence ? Elle est inexistante : Blackberry continue sa descente aux enfers avec 6 %, Windows Mobile demeure inexistant. L'autre enseignement est la part de plus en plus importante des smartphones sur le marché mobile (67 % du total). Reste à voir comment les deux premiers ont agi pour les fêtes de fin d'année et si Windows Phone va enfin redresser la tête.



Un bug pour Crucial M4

Le constructeur Crucial a découvert un bug gênant et étonnant. Son excellent SSD M4 a un problème. Après 5200 heures d'utilisation, il se bloque, nécessitant un redémarrage. Il semblerait que le bug se soit niché dans le firmware du disque. Crucial a sorti une mise à jour du firmware pour supprimer ce vilain bug. Comme quoi, le point faible du SSD n'est peut être pas les modules mémoires mais la partie purement logicielle...

Et la souris devient un cube...

Logitech invente la souris Cube. Ce cube est d'un format assez petit et sans fil. Il peut être souris, surface tactile et pointeur. La forme peut gêner plus d'un utilisateur mais pourquoi pas. Elle fonctionne avec Windows et OS X. Par contre, son prix pourra paraître élevé (70 euros) même si le côté multifonction rend la facture moins douloureuse. Si vous l'avez utilisée, faites-nous le savoir ! Pour plus d'informations : <http://www.logitech.com/fr-fr/mice-pointers/mice/devices/9472>



Le lapinou Nabaztag revit !



Depuis le 24 décembre minuit, votre lapin favori revit enfin ! Nabalive l'a reconnecté grâce à la communauté les lapins. Il s'appuie sur le projet Open Jab Nab. Les applications et services sont hébergés sur de nouveaux serveurs. Des problèmes de connexion sont encore existants pour la première génération. Plusieurs milliers de lapins se connectent chaque jour... site : <http://www.nabalive.com/#home>

Le retour du PC à 100 dollars... sous forme de tablette



Le projet OLPC n'est pas mort. Après avoir longtemps attendu le PC à 100 dollars, qui ne fut pas à la hauteur des attentes, le projet revient cette fois-ci sous forme d'une tablette, la OLPC XO-3. L'idée est d'aboutir à une tablette flexible. Un panneau solaire permettra de l'alimenter et bénéficiera d'un écran pour lire directement les livres, manuels scolaires. Plusieurs systèmes pourront être utilisés (SugarOS ou Android). Un port USB sera disponible ainsi que des entrées / sorties audio. Le cœur sera animé par un Marvell Amarda et bénéficiera d'une mémoire de 512 Mo et d'un stockage de 4 Go.

Sécuriser votre mot de passe

Quel que soit le type de site web proposant un espace privé ou personnel comme les sites de e-commerces, réseaux sociaux, messageries, jeux...), il est important de sensibiliser les possesseurs des comptes aux risques encourus au niveau des données et par conséquent à la seule porte d'accès intéressante : le mot de passe.

On entend régulièrement que des millions de comptes ont été piratés. Les cibles actuelles des pirates concernent les sites comme Facebook, Twitter, Hotmail, Google, Yahoo... Mais ces sites mettent de nombreux moyens pour enrayer le phénomène. Les personnes réalisant ces « exploits », ont généralement de mauvaises intentions en se servant du contenu des données récupérées. Ces intentions sont diverses, car en plus de ce vol de données privées, elles peuvent connaître vos messages, photos, vie privée... Mais aussi usurper à votre insu votre identité et donc se faire passer pour vous. Les mots de passe peuvent vous permettre de garder un certain niveau de confidentialité, mais tout dépend de la complexité de ceux-ci. C'est pourquoi, de nombreux sites web mettent en place des procédures pour aider leurs membres à sécuriser leur mot de passe.

LES ERREURS

Les erreurs autour du mot de passe sont très variées, tant au niveau du choix, que de la façon de se faire abuser

> Le choix du mot de passe

La sélection de votre mot de passe est un choix important, car les études ont démontré que certains mots de passe sont faciles à retenir, et par conséquent, sont des clés très recherchées par de nombreuses personnes.

Le TOP 3 des mots de passe les plus répandus sont :

- 123456
- password
- 12345678

De nombreux sites proposent régulièrement un classement des mots de passes les plus répandus.

> Les points à éviter

Il faut aussi éviter les mots de passe trop évidents, comme par exemple :

- Les mots venant du dictionnaire
- Les prénoms
- Les noms
- Les dates de naissances

L'utilisation visuelle du clavier, c'est-à-dire 123 ou AZERTY est fortement déconseillée. De la même façon si vous retenez une suite de lettres ou de chiffres ou la répétition d'une séquence. Il est préférable d'utiliser un mot de passe différent entre vos comptes de messageries et vos comptes de sites web. Par ailleurs, vous ne devez pas mettre en place un mot de passe de moins de 5 caractères.

> Type d'attaque

L'attaque la plus répandue sur le mot de passe concerne le phishing, c'est-à-dire par l'intermédiaire de votre messagerie, vous recevez un message pour vous demander de vous identifier à nouveau. En fait, il

s'agit d'un site web identique à celui que vous utilisez régulièrement, seulement un ou 2 petits détails montrent qu'il s'agit d'une supercherie. Le repère vient souvent de l'URL car même si le site propose un accès sécurité avec un en-tête identique à l'original, vous pouvez avoir une URL qui pointe ailleurs, comme un serveur pirate.

LES BONS CHOIX

Pour choisir le « bon » mot de passe, il suffit juste de respecter certaines règles.

> Composition

Un mot de passe, pour être considéré d'un très bon niveau, doit se composer d'une ou plusieurs lettres en Majuscules, avec un ou plusieurs chiffres aléatoires. La longueur totale du mot de passe est aussi un critère important, car plus celui-ci est long, mieux c'est pour vous.

> Vérifier le niveau de sécurité

Lorsque vous avez choisi votre mot de passe, il faut en vérifier son niveau de pertinence. Ce niveau se mesure de différentes manières :

- Soit par l'intermédiaire d'un site web (voir encadré), qui attribue une note avec un résultat allant de faible à très élevé.
- Soit directement par le site qui vous propose un compte personnel. Bien entendu, vous pouvez réaliser l'opération par vous-même en quelques lignes de codes PHP.

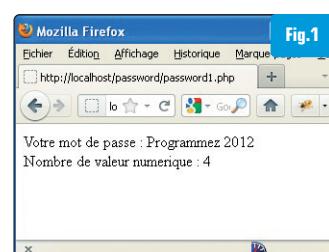
Nous validons le mot de passe, si celui-ci possède au moins une valeur numérique.

```
<?php
$motDePasse = 'Programmez 2012';
$resultat=0;
for ($i=0; $i <= strlen ($motDePasse); $i++)
{
    if (is_numeric($motDePasse[$i])==1)
        $resultat++;
}

echo "Votre mot de passe : ".$motDePasse."<br>";
echo "Nombre de valeur numérique : ".$resultat;
?>
```

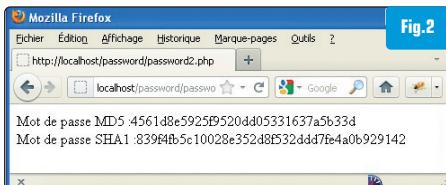
pour obtenir le résultat suivant :
[Fig.1]

Bien sûr, vous pouvez contrôler la pertinence de votre mot de passe, en déterminant des critères supplémentaires comme la détection des caractères majuscules, des symboles, etc.



Ensuite, au lieu de choisir un mot de passe, vous pouvez appuyer sur les fonctions MD5 ou SHA1, qui sont des algorithmes mathématiques.

```
<?php
echo 'Mot de passe MD5 :'.md5 ('Programmez')."<br>";
echo 'Mot de passe SHA1 :'.sha1 ('Programmez')."<br>";
?>
```



Générateur

Cependant, vous pouvez vous appuyer sur un générateur de mots de passe qui vous proposera directement un mot de passe suivant une longueur que vous aurez définie.

```
<?php
function creation_password($longueur=10)
{
    $alphabet = "abcdefghijklmnopqrstuvwxyz";
    $alphabet .= "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    $alphabet .= "1234567890";
    $i = 0;
    $pass = "";

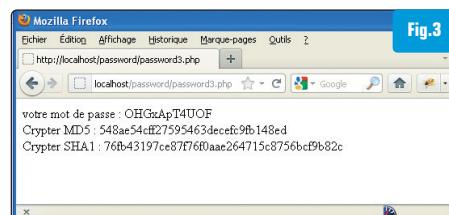
    srand((double)microtime()*1000000);
    while ($i<=$longueur)
    {
        $valcara = rand(0, strlen($alphabet));
        $pass .= substr("$alphabet",$valcara,1);
        $i++;
    }
    return $pass;
}
?>
```

Cette fonction va fournir un mot de passe que vous pourrez utiliser et enregistrer dans ce nouveau site. Nous considérons que le mot de passe sera d'une taille de 10 caractères.

Son contenu est défini par un nombre aléatoire en milliseconde par rapport à l'horloge du serveur. Cette valeur permet de sélectionner une lettre par rapport à la chaîne proposée.

Après avoir conçu le mot de passe, vous pouvez appliquer un cryptage supplémentaire avec un préfixe et un suffixe.

```
<?php
$password=creation_password();
echo "votre mot de passe : ".$password."<br>";
$md5=md5($password);
echo "Crypter MD5 : ".$prefixe.''.$md5.''.$suffixe."<br>";
$sha1=sha1($password);
echo "Crypter SHA1 : ".$prefixe.''.$sha1.''.$suffixe."<br>";
?>
```



Voici le résultat [Fig.3]. L'exemple ne montre pas comment réaliser le préfixe et suffixe car différentes manières existent et elles sont propres à chacun. Bien entendu, vous pouvez appliquer les mêmes règles pour la conception du mot avec un type de codage différent.

SOLUTION

Il n'existe pas de solutions miracles, seulement quelques conseils que vous pouvez suivre. Tout d'abord, ne pas faire confiance aux sites web, car même s'ils annoncent le 100 % sécurité au niveau de votre compte, il peut y avoir un risque provenant d'un autre compte associé ou d'une nouvelle technique de pénétration.

Mais la solution la moins risquée, c'est de réaliser certaines opérations manuellement :

- Il est important d'utiliser des mots de passe différents entre vos divers comptes (messageries, services bancaires, réseaux sociaux...)
- Vous devez penser à changer vos mots de passe régulièrement, principalement les plus importants.
- Se déconnecter après votre session quand vous utilisez un ordinateur qui ne vous appartient pas.
- Fermer vos sessions si vous n'êtes pas le seul à utiliser votre ordinateur, ce qui permet de rester le plus prudent possible.
- Un autre point important, consiste de ne pas répondre aux emails (phishing) qui vous demandent de vous identifier à nouveau pour vérifier que les serveurs sont à jour, même si cela vous semble plausible.
- Prévoir un mot de passe assez complexe.

Par ailleurs, il est important de ne pas laisser visible un fichier texte dans votre ordinateur, surtout si celui-ci possède la liste de vos mots de passe et est accessible par différentes personnes.

CONCLUSION

De nombreuses sociétés de messagerie ou de sites possédant des millions de comptes, mettent régulièrement des nouveaux systèmes pour que vous puissiez récupérer un nouveau mot de passe. Ces systèmes sont variés, ce peut-être aussi bien une question cachée ou un numéro de téléphone pour recevoir un SMS qu'un compte de messagerie différent de celui que vous avez enregistré. Bien entendu, même si vous appuyez sur ces différents services, il faut aussi, de votre côté mettre un mot de passe assez conséquent.

Ressources

- Password checker (<http://www.microsoft.com/protect/yourself/password/checker.mspx>)
- password meter (<http://www.passwordmeter.com/>)
- Tester son mot de passe (<http://cs76.free.fr/test-mot-passe.php>)
- Security stats (<http://www.securitystats.com/tools/password.php>)
- Javascript Password strength Meter (<http://www.geekwisdom.com/dyn/passwdmeter>)
- Password strength checker (<http://tools.devshed.com/webmaster-tools/password-strength/>)

Christophe Villeneuve

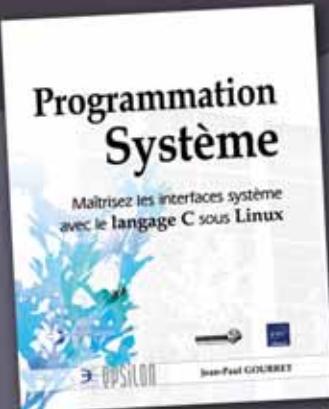
Consultant pour Alter Way solutions, auteur du livre « PHP & MySQL-MySQLi-PDO, Construisez votre application », aux Éditions ENI. Rédacteur pour nexen.net, membre des Teams DrupalFR, AFUP, LeMug.fr, PHPTV..

Windows Azure Platform

Développez, déployez et administrez pour le Cloud Microsoft

Windows Azure Platform
Développez, déployez et administrez pour le Cloud Microsoft

version
numérique
OFFERTE
pour l'achat
d'un livre imprimé



Programmation Système
Maîtrisez les interfaces système avec le langage C sous Linux

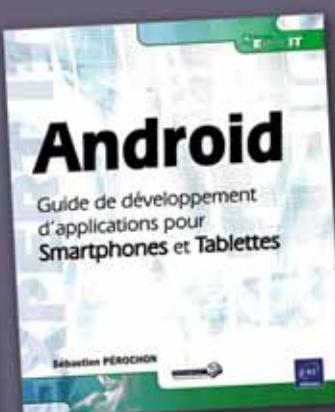


C#4
Les fondamentaux du langage
Développer avec Visual Studio 2010

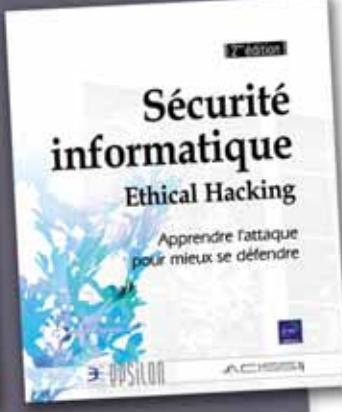
Windows Azure Platform

Développez, déployez et administrez pour le **Cloud Microsoft**

Florent SANTIN



Android
Guide de développement d'applications pour Smartphones et Tablettes



Sécurité informatique
Ethical Hacking

www.editions-eni.fr

eni
éditions

Au cœur de Windows 8 : WinRT

Depuis la BUILD de septembre 2011, beaucoup de questions se posent aux développeurs autour du nouveau modèle applicatif de Windows 8 : WinRT. Celui-ci impacte directement le développement et le modèle de programmation propre à Windows. La première version bêta de Win8 devrait permettre de clarifier certaines interrogations. Plongeons-nous dès maintenant dans WinRT.

Le challenge pour Microsoft est grand, les ventes de Windows 7 sont à faire pâlir n'importe quel autre OS du marché, l'adoption est rapide et elle continue de se réaliser chez les particuliers et les entreprises. Sortir Windows 8, dans la même lignée que Windows 7 c'est être certain de faire un "bide". Après Windows 7, pourquoi un utilisateur, qui vient juste de changer de système, irait migrer vers une nouvelle ver-



sion ? Il fallait donc innover, ou au moins changer profondément les fondations, tout en répondant à la problématique des tablettes. Voilà plusieurs années que l'on attendait un OS ayant une capacité tactile. Pendant un temps nous avions pensé à un éventuel portage de Windows Phone 7 vers un OS tablette mais déjà à la PDC 2010 le mot de Microsoft était "Non, pas de WP7 sur tablette" (malgré des rumeurs et des employés de l'éditeur exprimant le contraire). Windows 8 est donc, à mon avis, la meilleure façon de répondre au marché actuel (de Microsoft) :

- Avoir un système unique PC/Tablette. Certains diront que Microsoft prend de l'avance sur Apple ou Google dans ce domaine, mais les approches Apple et Google ne sont pas identiques. Pour iOS rappelons qu'il s'agit d'un dérivé de OS X, commun aux terminaux mobiles (smartphone, tablette, iPod), Google a pris le même type d'uniformité avec Android 4. Par contre Apple a décidé avec OS X Lion de reprendre quelques principes iOS dans le système desktop. L'avenir dira quelle est la meilleure solution.
- Pouvoir créer un nouveau besoin. Aujourd'hui les tablettes sont dominées par Google et Apple mais pas les "Hybrids". Un hybrid c'est à la fois un PC et une tablette. Un peu comme Apple a réussi à changer les mœurs avec un téléphone tactile, Microsoft veut changer les mœurs des PC en poussant le tactile sur écran, qu'il s'agisse ou non d'une tablette. Microsoft sera l'acteur, le moteur d'un nouveau cou-

rant et si cela fonctionne, cela boostera indirectement les ventes de ses tablettes (en espérant une définition plus claire que celle des TabletPC, ndlr).

- Pouvoir proposer un système différent, sans pour autant oublier toutes les nouveautés de Windows 7.
- Avoir des fonctions « cloud » intégrées au système (sans être aussi radical que ChromeOS, ndlr). En effet, si on ne prend que la partie métro, les applications sont toutes des applications pouvant à l'avenir être hébergées dans le cloud. Windows ne servant que de "pont" entre le cloud et le end-user. On notera d'ailleurs qu'un espace cloud est réservé pour les applications métro pour enregistrer les paramètres de l'utilisateur [Fig.1].

Alors qui dit tablette, dit forcément processeur ARM (je parle des vraies tablettes qui n'ont pas 4H d'autonomie). J'imagine bien qu'il existera plusieurs versions de Windows 8. Une version allégée, c'est-à-dire juste avec l'interface Métro que l'on peut voir en photo. C'est d'ailleurs à mon avis pour cette raison que l'internet explorer de Métro ne possède aucun plug in comme Flash et Silverlight, afin de ne pas avoir à les porter sur la version allégée pour tablette.

Puis une version pour tablette avec processeur Intel et PC, qui proposera également une version "Windows 7 améliorée" comme nous le trouvons actuellement avec le support de Silverlight et .NET.

Le seul reproche que l'on pourrait faire à Microsoft est que la version PC avec souris (pour nous simple mortel non doté d'écran



tactile) n'est pas franchement agréable à utiliser. Mais je fais confiance aux ergonomes de Microsoft. Je crois qu'on parle d'un choix d'écran au démarrage.

En tout cas, ce qui est agréable c'est de voir que Microsoft est capable de prendre un tournant assez radical. Nous constaterons ou non si dans l'avenir tout cela va payer.

A noter que le dernier tournant que j'ai eu la chance de connaître, fut le passage de COM à .NET. Nous en sommes au même point, et le passé nous a démontré que si Microsoft n'avait pas fait ces choix par le passé, nous aurions perdu la guerre de la compétitivité technologique.

On peut le voir comme une machine arrière, c'est-à-dire après avoir vécu le COM->.NET, on fait le .NET->COM. Mais avec tous les avantages des deux technologies.

Windows 8 arrive avec boutique d'application. Totalement copié sur la philosophie Apple (App Store), certes, mais tellement indispensable de nos jours. Aujourd'hui, un développeur d'application Windows qui veut vendre son produit passe plus de la moitié de son temps à gérer tout l'aspect sécurité/sérial/vente de son produit. C'est totalement ingérable. Sans même parler de tous les sites qui pullulent sur le net tel que Clubic, Telecharger.com, etc.

Là nous avons un centre d'application, qui va rendre la publication facile pour un développeur et qui va permettre à l'utilisateur d'avoir des applications sûres. Cela aura certainement un aspect positif sur les ventes d'applications: histoire de redonner goût aux gens d'acheter des choses pour leurs PC, à l'image de leurs téléphones.

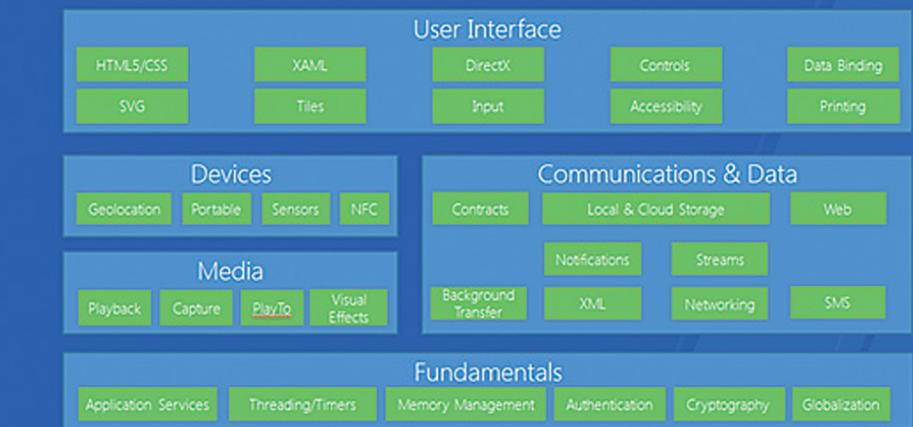
Là où Microsoft réussit son pari, c'est de proposer un parc informatique le plus gros au monde avec des langages de développement, accessible à tous. En effet, nous allons voir par la suite que Microsoft ouvre le développement d'applications Métro à HTML 5 et JavaScript.

> Une autre façon de concevoir des applications pour Windows

Enfin, quand je dis Windows, je parle des applications dites "Métro". C'est-à-dire, une application qui tournera en environnement Métro pour une utilisation tactile ou non, compatible tablettes, PC, etc. On parle aussi de "immersive apps". Il restera (ouf !) toujours les applications desktops type Win32, Silverlight, .NET qui fonctionneront sur le bureau Windows 8 (et sans modification par

Metro style application APIs

Fig.2



rapport à Windows 7, hormis les nouvelles API). Microsoft nous propose un nouvel environnement sandboxé pour développer: WinRT pour Windows RunTime.

De quoi s'agit-il ? [Fig.2]

C'est tout simplement un framework allégé de .NET (sans ASP.NET par exemple), ce n'est certainement pas un remplaçant mais une version qui va permettre de créer des applications Métro.

Au même titre que Silverlight ou Silverlight for Windows Phone sont deux environnements de développements allégés (mais le principe reste le même et proche de .NET). Pour info, WinRT possède même moins d'assemblées que Silverlight. Ne soyons pas perturbés par ce nouveau venu, il utilise les langages que nous connaissons.

WinRT est inspiré de .NET, avec un gros ménage réalisé à l'intérieur et des nouveautés permettant de faire des applications métro (comme l'utilisation de tiles par exemple). Des API inutiles, dangereuses, dupliquées, obsolètes n'ont pas été intégrées à WinRT.

Pour utiliser WinRT, il faudra soit utiliser le bon vieux couple VB/C++/C# (rayez l'option inutile) avec XAML ou bien HTML/CSS/JavaScript. Que ce soit du JavaScript ou du C# (5 avec les nouveautés type await), ces langages vont être capables de discuter avec les API du Windows Runtime. A noter que l'utilisation du JavaScript et HTML n'est pas exactement la même que pour des sites web et qu'il n'y a pas l'objectif de portage de code entre .NET->métro et HTML web->Html pour métro.

Un objet dans WinRT n'est pas du managé mais peut être utilisé par du C# ou du JavaScript grâce au WinRT Object Manager et la

projection du type vers le langage souhaité (un wrapper en quelque sorte!).

Un objet WinRT n'est donc pas un System.Object mais bien un Objet différent (implémentant de nouvelles interfaces, des métadatas pour Windows pour faire du versionning...).

WinRT n'est pas un environnement managé (mais peut être utilisé par une CLR .NET utilisant C# grâce à des wrappers). C'est en gros la nouvelle version de la bonne vieille API Win32 de Windows qui datait de 1995. Si vous comprenez bien, le JavaScript peut faire des appels COM: ceci est une révolution! L'avantage des performances est encore du côté de C++ puisque que comparé à C# et JavaScript qui nécessitent une CLR, le C++ fait des appels non managés directement (quoique, je vous conseille de lire le livre « Ecrire des applications performantes en .NET » aux éditions ENI). L'API WinRT possède des wrappers managés qui sont automatiquement référencés lors de la création d'un nouveau projet type Metro App (même si elles ne sont pas affichées dans l'EDI dans le dossier Références).

Voici quelques exemples de différences entre WinRT et .NET (pour un portage de code, nous pouvons utiliser les régions #if !WINRT #else) :

- En Silverlight ou .NET, on déclare l'accès à une librairie avec un XML namespace comme ceci:

```
1 xmlns:local="clr-namespace:MyAssembly.  
MyNamespace;assembly:MyAssembly"
```

en WinRT, on se simplifie la vie et on retire l'assembly :

```
1 xmlns:local="using:MyAssembly.MyNamespace"
```

- Des namespaces d'assemblies ont changé (System.Windows devient Windows.UI...)
- L'écriture dans un fichier, l'utilisation d'une webrequest etc. beaucoup d'API ont été renouvelées

> Création d'une application pour Windows 8 avec C#/XAML

Etant un peu égocentrique, l'objectif est de créer l'application permettant de lire les news de mon blog. Pour cela, j'ai repris les éléments fondamentaux exposés par Microsoft :

• Création d'une identité graphique (et bases de l'ergonomie sur tablette)

Je ne vais pas encore vous décrire Métro, l'intérêt d'une bonne typographie, etc. L'objectif va être pour moi de créer un design sobre. Je vais opter pour les couleurs noir et blanc en jouant avec mon logo [Fig.3].

• Création du projet

N'étant pas un grand fan de HTML5/JavaScript, je décide de partir en premier sur un projet C# : [Fig.4]

Plusieurs templates sont proposés, nous optons pour le template le plus minimalistique possible : Application.

La première chose que l'on remarque, est l'apparition d'un nouveau fichier dans le Solution Explorer: Package.appxmanifest.

Ce fichier XML est le fichier manifest de votre application. Il permet d'initialiser le splashscreen, de gérer les capacités utilisées par l'application etc.

Avant de configurer ce fichier en mode graphique, intéressons-nous au cycle de création d'une application avec Visual Studio. Nous avons un code C#, qui comme à son habitude, va être compilé en MSIL avec CSC.exe.

Suite à ce traitement, l'exécutable et ses ressources passent dans un logiciel permettant de "packager" l'application (MakeAppx.exe). C'est là que notre fichier XML rentre en jeu. Le résultat est un bon fichier ZIP avec une extension .Appx.

Une fois le package généré, il faut le signer pour pouvoir l'installer sur une machine (grâce à signtool.exe) puis côté client, on peut l'installer avec une cmdlet PowerShell

 (Add-AppxPackage).

Tout ceci est automatique avec Visual Studio 2011. Revenons à nos moutons, configurons ce fameux fichier Package.appxmanifest : [Fig.5].

On nous demande une suite de choses :

- Titre, description et point d'entrée dans l'application
- 3 images dont 2 pour la tile de l'application et une pour sa représentation dans des listes (recherche, partage). Les images

peuvent être transparentes.

- Personnalisation du texte sur la tile et sa couleur. Personnalisation de la couleur de fond de la tile
- La personnalisation des toasts (alertes) de notre application

Après cette configuration, voilà ce que cela nous donne : [Fig.6].

Dans les autres onglets de cet assistant, nous retrouvons les capacités requises par notre application.

Nous pouvons demander que notre application puisse accéder aux ressources réseaux, aux documents de la machine, l'utilisation de port spécifique, la géolocalisation, la gestion de la webcam etc [Fig.7].

L'onglet suivant est l'onglet Declarations. Il va permettre de s'abonner à des "contrats" et proposer une meilleure intégration dans le système.

Mon application, va par exemple proposer des contacts (le mien et celui de mes blogs rolls par exemple), elle va aussi permettre la recherche dans mes billets [Fig.8].

Ceci va permettre ce genre d'intégration lorsque l'utilisateur cherchera quelque chose : [Fig.9].

Le dernier onglet permet de préparer l'application au déploiement [Fig.10].

Juste pour le test, et avant même de vouloir coder quoi que ce soit, je décide d'ajouter un

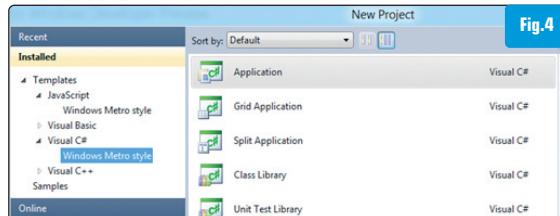


Fig.4

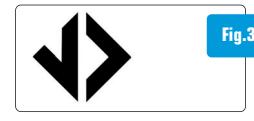


Fig.3

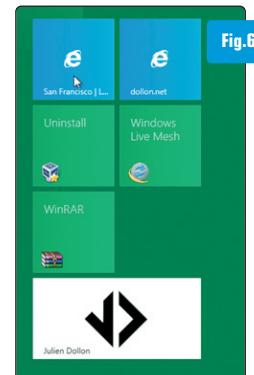


Fig.6

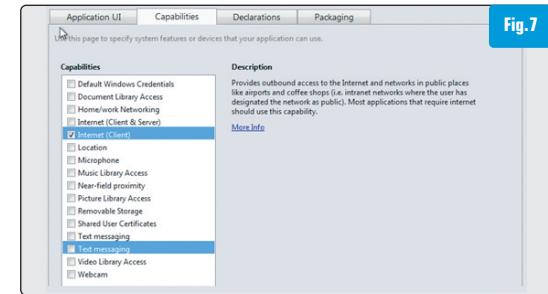


Fig.7

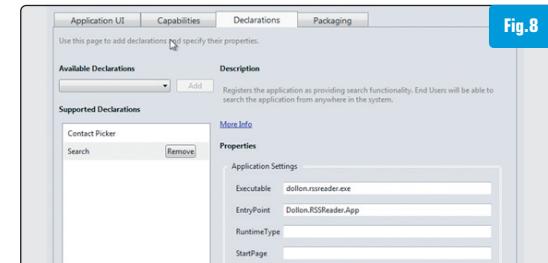


Fig.8

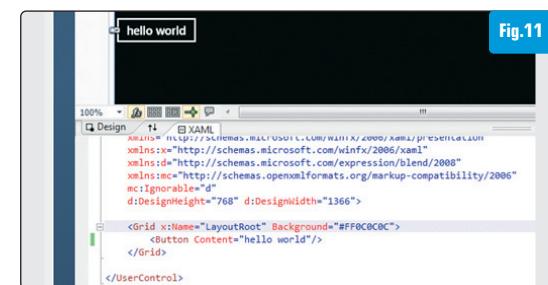


Fig.11

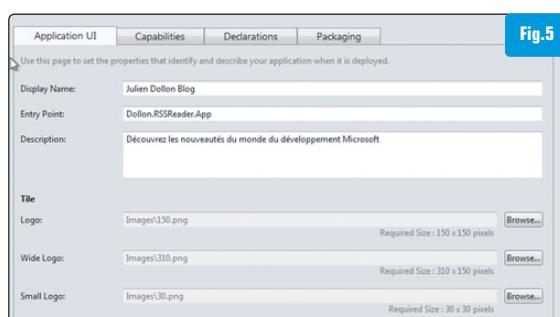


Fig.5

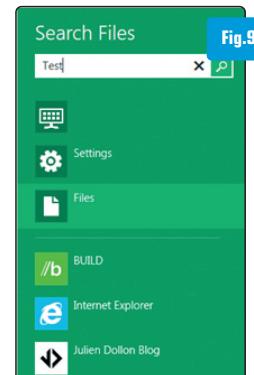


Fig.9

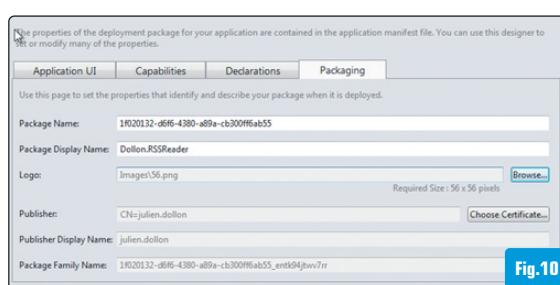


Fig.10

bouton sur la première interface : [Fig.11]. WTF? Là où TOUTES les plateformes de développement XAML auraient affiché un grand bouton centré étiré, là on a un bouton à gauche tout petit... Pas mal de paramètres d'initialisation changent et j'avoue que cela fait un peu bizarre.

• Architecture

Je ne m'attarderai pas sur ce point mais je vous conseille le dernier livre de Jonathan Antoine sur MVVM : [Fig.12].

<http://digitbooks.fr/catalogue/mvvm-antoine-lebrun.html>

Ainsi que MVVMLight pour Windows 8 déjà disponible (merci Laurent!)

<http://geekswithblogs.net/lbugnion/archives/2011/10/03/mvvm-light-toolkit-for-windows-8-preview-mvvmlight-win8.aspx>

• Rendu fluide et rapide

Comme l'indique Microsoft, il nous faut créer une application ergonomique (ce qui inclut la fluidité de celle-ci).

Je ne vais pas faire un cours d'ergonomie ici, mais il ne faut pas oublier que nous sommes dans un contexte "Touch", et il n'y a rien de plus frustrant que d'avoir du mal à atteindre un bouton avec son doigt, ou encore pire, que l'application ne réponde pas suite à un touché (on notera que l'utilisateur est beaucoup moins patient avec du touch qu'avec une souris) [Fig.13].

A ce moment de l'écriture de l'article, j'étais motivé à faire une superbe application mais c'est lors de l'ouverture de Blend 5 Preview

que mon envie s'est vite envolée.

En effet, la dernière mouture de Blend ne supporte pas encore le XAML 😞 alors designer une application à la main dans Visual Studio n'est pas des plus drôle.

Je me contenterai d'une liste à droite et d'un logo à gauche 😊. De plus, certaines choses provoquent des bugs, comme les ObservableCollections ou sur INotifyPropertyChanged. Voilà le rendu : [Fig.14 et 15].

Après quelques manipulations, je trouve le nombre de bugs impressionnant, le manque d'outils flagrant (je n'ai pas pu tester les transitions fluides...). Vivement les prochaines releases ! Le point positif est que les contrôles comme listbox, bouton ou autres sont bien intégrés au style Métro et aux comportements "touch".

Le développement en général, que ce soit en terme de navigation inter page, animations, binding et j'en passe, se font comme en Silverlight/WPF.

• Utilisation de l'AppBar

La barre applicative, ressemblant à celle du Windows Phone, est un atout pour nos applications. Elle permet de rendre invisible à l'écran des options de "second choix". Mais contrairement au téléphone, les icônes ne sont pas centrées mais disposées soit à gauche, soit à droite pour faciliter la prise en main [Fig.16 et 17].

• Gestion des vues (snap, scale...), HD?

Comme nous le savons, Windows 8 n'est pas seulement destiné aux tablettes. Ce qui

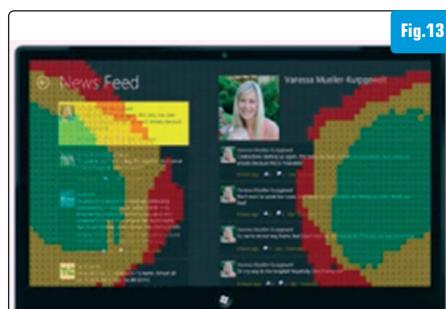
implique de penser à la fois aux écrans type <10" mais aussi aux écrans géants avec de l'Ultra HD. Pour que l'application soit HDReady: 2 solutions. Soit nous travaillons seulement en vectoriel (ce qui peut se faire en XAML mais aussi en HTML5), soit il faut fournir trois tailles d'images par image.

Une image à sa taille normale pour un écran tablette, une image à 140% de sa taille pour les écrans normaux et une image à 180% pour les grands écrans. Une autre dimension à prendre en compte, est l'orientation d'écran et le type d'écran choisi par l'utilisateur. Quand nous développons, l'importance des marges reste la même qu'en Silverlight. La tablette pouvant être utilisée de manière horizontale ou verticale. Blend proposera bientôt un support du XAML et donnera accès au choix de l'orientation lors du design de l'application. Une autre chose à ne pas oublier, est que l'application peut être en position "snap", c'est-à-dire de manière réduite sur le côté d'un écran.

• Implémenter les bons contrats

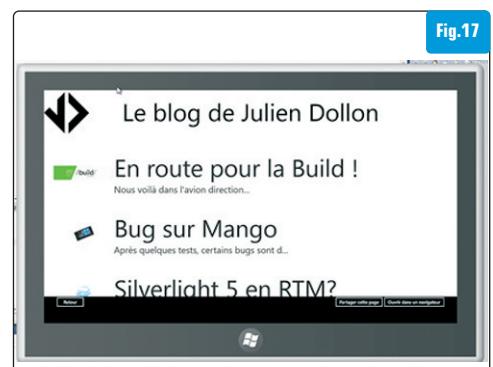
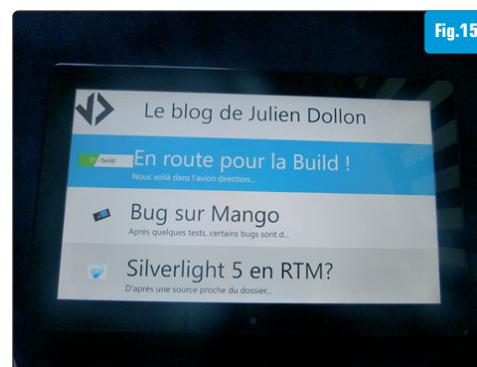
Nous avons vu comment déclarer l'utilisation des contrats. Il est important de s'abonner aux contrats, c'est ce qui poussera l'utilisateur à plus souvent utiliser notre application. Après s'être abonné, notre application apparaît au niveau du système (voir la fonction recherche) : [Fig.18].

Côté code, il nous est possible de choisir une fenêtre en particulier pour le traitement de ce contrat et de récupérer le texte tapé par l'utilisateur : [Fig.19].



```
Fig.12
Fig.13
Fig.16

<ApplicationBar IsPersistent="False" Height="88" VerticalAlignment="Top" Style="{StaticResource AppBarStyle}">
    <TextBlock Text="http://julien.dollon.net" FontSize="40" Margin="10"/>
</ApplicationBar>
<ApplicationBar IsPersistent="True" Height="88" VerticalAlignment="Bottom" Style="{StaticResource AppBarStyle}">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*"/>
            <ColumnDefinition Width="Auto"/>
        </Grid.ColumnDefinitions>
        <StackPanel Grid.Column="0" Content="Retour" HorizontalAlignment="Left" VerticalAlignment="Top"/>
        <StackPanel Grid.Column="1" Orientation="Horizontal" HorizontalAlignment="Right" VerticalAlignment="Top">
            <Button Content="Partager cette page"/>
            <Button Content="Ouvrir dans un navigateur"/>
        </StackPanel>
    </Grid>
</ApplicationBar>
```



• Utiliser la tile pour afficher des informations

La tuile peut être affichée de deux manières: réduite ou normale (seule la réduite est obligatoire). Réduite, cela permet à l'utilisateur de lancer l'application. Cependant, s'il fait le choix de l'avoir en grand, cela va nous permettre d'exploiter cet espace pour afficher des informations, comme une image ou du texte. Autre solution, afficher un compteur dans la tuile (qu'on appelle Badge), de la même manière que d'afficher des images ou du texte, il va falloir formater un flux XML que l'on va envoyer au système : [Fig.20 et 21]. Attention, dans cette version bêta, le compteur met parfois du temps à apparaître.

Exemple de texte pouvant être affiché dans la tuile : [Fig.22 et 23].

Comme à l'image de Windows Phone Mango, nous allons aussi pouvoir générer des tuiles sur la home de l'utilisateur qui réagiront comme des raccourcis vers notre application. Nous ajoutons un bouton à côté de chaque news. Sur le touch de ce bouton, nous déclenchons la création d'une tuile sur la home redirigeant vers SelectedNews.xaml? id=id-de-la-news. Plus d'informations ici (il faut bien creuser, beaucoup d'options sont disponibles comme le fait de mettre une date d'expiration sur une news d'une tuile...) : [http://msdn.microsoft.com/en-us/library/windows/apps/br208661\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/apps/br208661(v=VS.85).aspx)

• Notifier l'utilisateur

Mais où est mon MessageBox.Show?!? 😊

Tout d'abord, les Popup à la "MessageBox" se font désormais comme ceci : [Fig.24 et 25].

La notification d'un utilisateur peut se faire de différentes manières. Tout d'abord de façon passive, pour les choses importantes, c'est-à-dire la mise à jour de la tuile comme vu précédemment. Mais il existe aussi les notifications systèmes (toasts), un peu à la MSN Messenger, à condition que nous ayons laissé le choix à l'utilisateur de désactiver ce type de message dans notre application [Fig.26]. Peu importe le type de notification que nous avons choisi, trois solutions pour les déclencher :

- Grâce à du code exécuter au sein de l'application
- Grâce à une notification programmée (Scheduled)
- Grâce à un système de notification push comme sur téléphone, en utilisant Windows Push Notification qui sera capable de provoquer le réveil de notre application.

Pour plus d'informations sur ces toasts :

<http://www.nickharris.net/2011/09/delivering-notifications-using-windows-azure-and-windows-push-notification-service/>

• Stocker ses préférences dans le cloud

Rien de plus frustrant que de devoir perdre

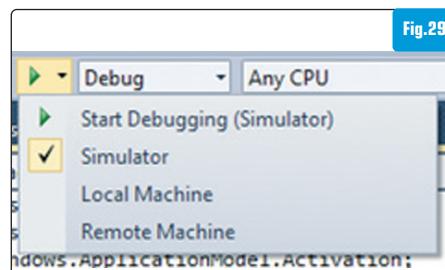


Fig.29

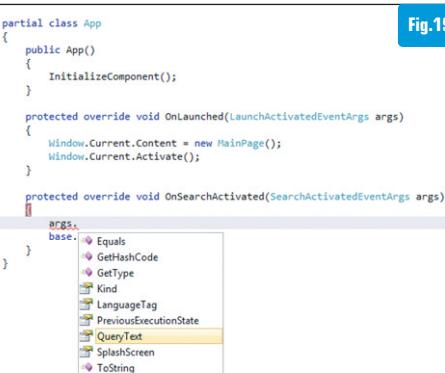


Fig.19

ses données de configurations d'un PC à l'autre. Microsoft résout ce problème en proposant un espace cloud pour stocker les préférences, l'état et une petite partie des données par utilisateur et par application.

Je n'ai pas eu le temps de m'y intéresser et je n'ai pas encore trouvé beaucoup de documentation. Je pense que Microsoft va partir sur l'exploitation du SDK LiveSDK/Skydrive pour stocker ce genre de choses.

Pour plus d'informations, voici un exemple : <http://code.msdn.microsoft.com/windowsapps/Skypad-SkyDrive-Sample-8a187bd8>

• Application active/suspendue

Quand un utilisateur quitte/switch une application Métro pour aller sur une autre, il met l'application en "Suspend".

L'objectif est de garder une bonne expérience utilisateur en gardant une machine réactive. Le problème vient quand l'application a besoin de faire un traitement alors qu'elle est suspendue, ou tout simplement d'enregistrer son état! Pour tester ce comportement

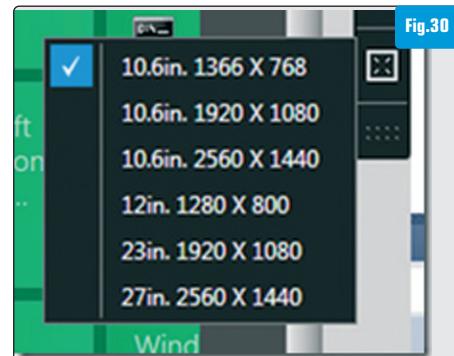


Fig.30

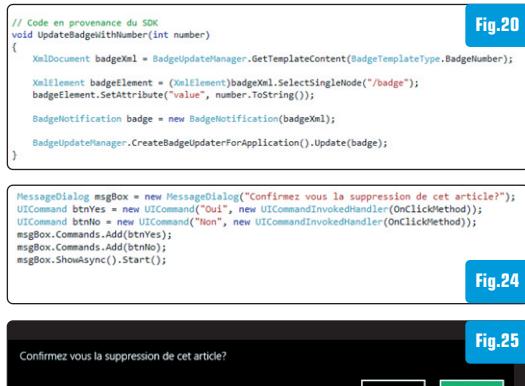


Fig.20



Fig.21

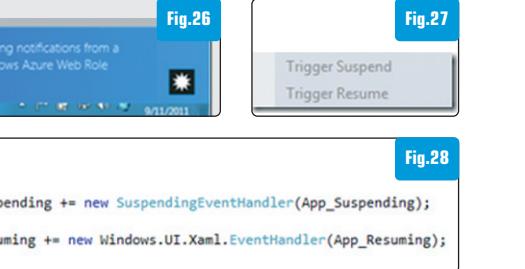


Fig.22



Fig.18



Fig.23

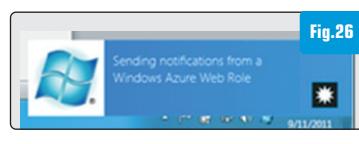


Fig.26



Fig.27



Fig.28



ment, dans le menu debug de Visual Studio existent deux triggers : [Fig.27]. Si un traitement est à faire avant la suspension, nous avons besoin de nous abonner à l'événement Suspending [Fig.28]. Puis on peut utiliser une tâche d'arrière-plan pour finir l'action en cours, que je vais décrire par la suite.

• Tâche en arrière-plan

Nous avons vu que nous pouvions utiliser un service de push pour rafraîchir une application. Il existe également, un Playback Manager pour tout ce qui est jouer du son en tâche de fond. Mais surtout, chose très utile, Microsoft met à disposition le "Background task".

Une tâche en arrière-plan se voit réservée une petite partie de CPU, et ne doit pas nécessiter d'action avec l'utilisateur. Microsoft conseille que la tâche ne dure pas trop longtemps pour économiser la batterie du device. Pour créer une tâche d'arrière-plan, il faut l'identifier par l'application qui en fait la demande et un déclencheur, qui lancera la tâche. Les déclencheurs sont de plusieurs types: Connexion internet disponible ou non, une application est ajoutée au lock screen, un update s'est terminé, la personne s'est connectée/déconnectée...

Si l'application est en cours d'exécution, la tâche d'arrière-plan utilise les mêmes ressources que celle-ci.

Sinon, un thread est créé. Pour des exemples de code, c'est ici que ça se passe :

http://download.microsoft.com/download/B/D/0/BD08229A-9DF6-4B41-AC9B-92F99043EFE6/Introduction_to_Background_Tasks_BUILD.docx

• Debugging

Un outil de performance fera l'objet d'un article supplémentaire : Windows Performance Analysis. Autre astuce, un mode intéressant : c'est la possibilité de lancer l'application avec un simulateur. Au moment du choix d'exécution de l'application, l'option nous est proposée : [Fig.29]. Ce simulateur est des plus pratiques, puisqu'il permet, entre autres, de tester l'application dans plusieurs résolutions, et plusieurs orientations ainsi que d'émuler certaines actions "Touch" qu'on ne peut réaliser sans avoir de tablette [Fig.30 et 31].

> Conclusion

Pour conclure, beaucoup de choses sont à venir. On ne parle plus seulement de .NET mais bien de développement Microsoft. Bon, sinon on est clairement sur une preview de chez preview. C'est léger, voire trop léger pour vous présenter cela maintenant. Je pense que j'attendrai une bêta 2 ou une RC pour m'y repencher sérieusement. En attendant, si vous voulez creuser, je vous conseille ce lien :

<http://code.msdn.microsoft.com/windowsapps>

Julien Dollen

<http://julien.dollen.net>

Développeur .NET à San Francisco et Full Professor SUPINFO

Les rendez-vous de l'open source

07
FEV

Open data : les enjeux de l'ouverture des données publiques

Séminaire Smile

Levallois-Perret - de 14:00 à 17:00

06
MAR

CMS open source : quel outil choisir pour mon projet ?

Séminaire Smile

Nantes - de 09:00 à 12:00

21
MAR

Salon Documation

Le RDV de la gestion de l'information
Paris / CNIT La Défense - de 09:00 à 18:30

27
MAR

Gestion documentaire : panorama des meilleurs outils open source

Séminaire Smile

Paris / Saint-Lazare - de 09:00 à 12:00

15
MAI

Gestion documentaire : panorama des meilleurs outils open source

Séminaire Smile

Nantes - de 09:00 à 12:00

19
JUIN

Salon Solutions Linux

La rencontre des professionnels du libre
Paris / CNIT La Défense - de 09:00 à 18:30



Pour vous inscrire gratuitement,
www.smile.fr ou utilisez ce QR code



Fig.31

Notre sélection de livres

JOSEPH ADLER **R l'essentiel**

Pearson, 48 €

R est le nouveau langage en vogue dans certains domaines informatiques comme la haute performance ou les données distribuées. Open source et puissance, R est l'idéal pour les statistiques et les analyses. Mais à quoi ressemble réellement R, comment le programme-t-on, et avec quels outils ? Ce livre répond à toutes les questions en abordant la syntaxe et la structure du langage, les fonctions, la notion de packages ou encore la manipulation statistique. L'auteur propose un ouvrage clair et très complet pour ceux qui veulent coder en R !

COLLECTIF **DirectAccess**

Eni, 39 €

DirectAccess est une technologie réseau simplifiant les connexions entre terminaux sous Windows 7 et Windows Server 2008 r2. Peu connu du public, DirectAccess est pourtant une fonction intéressante pour éviter les lourdeurs d'un VPN. Mais comme tout réseau, DirectAccess possède son mapping, ses règles, ses contraintes. Les auteurs, tous spécialistes Microsoft, expliquent pas à pas les arcanes du système, les

scénarii d'usages possibles, les outils, les configurations et toutes les astuces indispensables !

Collection « guide de survie »

Pearson, 18-19 €

- Linux : Les principales commandes et lignes de code utilisées au quotidien pour mener à bien les tâches, des plus simples aux plus complexes, sont abordées. L'ouvrage propose plus de 100 fragments de code et commandes personnalisables pour gérer et manier Linux dans toutes les situations.
- Java : Cette édition revue et corrigée présente les enrichissements proposés par la dernière version de Java (Java SE 7). De nombreux exemples de code pour démarrer rapidement ou se sortir d'une impasse !

OLIVER ANDRIEU **réussir son référencement web édition 2012**

Eyrolles, 32 €

Traitant de tous les aspects liés au référencement, ce livre constituera un guide précieux et complet pour tous ceux qui souhaitent renforcer la présence de leur site sur Internet. Il fournit des solutions techniques mais également des conseils pratiques pour mener à bien un tel projet: où

Notre sélection web du mois

➤ **Google Code blog** : ce blog officiel se dédie à l'ensemble des codes, projets, frameworks de Google. Une source d'informations essentielle pour suivre toutes les dernières annonces Google. Il fournit aussi régulièrement des points de vue ainsi que des exemples techniques toujours intéressants. URL : <http://google-code.blogspot.com/>

➤ **Springsource blog** : Springsource blog est le blog de l'éditeur springsource, auteur du framework Java Spring. Il donne les dernières sorties et évolutions de la plateforme. A découvrir ici : <http://blog.springsource.com/>

➤ **R Project** : site incontournable sur le langage R. Le développeur a accès à de nombreuses ressources (documentations, exemples, bug track).

Pour les utilisateurs de R : <http://www.r-project.org/>

➤ **Android France** : un site très complet sur toute l'actualité autour d'Android, aussi bien matériel que logiciel. Le seul défaut est l'absence d'une section développeur ! url : <http://android-france.fr/>

trouver un prestataire et sur quels critères le choisir ? Combien coûte un référencement ? Quelles garanties doit offrir un référencement à ses clients ? Comment gérer le suivi d'un référencement ? Le référencement est un art de tous les jours, et les techniques évoluent à chaque nouvelle version des moteurs de recherche et des nouvelles technologies. Un ouvrage de référence indispensable à tout webmaster !

SILVIA PFEIFFER **HTML 5 video**

Pearson, 36 €

Cet ouvrage présente les bases de la balise HTML5 VIDEO. Vous y verrez comment inclure une vidéo dans une page web, intégrer une compatibilité inter-navigateurs et encoder la vidéo pour une lecture sur tous les players (via des utilitaires open-source). Il traite aussi les propriétés CSS pour la présentation des conteneurs des éléments vidéo ou audio. Il aborde ensuite l'intégration des éléments 'média' avec d'autres technologies web. Vous apprendrez à remplacer les paramètres par défaut des navigateurs (skinning), utiliser l'API Javascript pour les éléments 'média', et l'intégration avec d'autres éléments HTML pour l'audio et la vidéo : SVG, Canvas, and Web Worker Threads.

PAUG : le groupe Android de Paris

Le Paris Android User Group est une association qui anime la communauté des passionnés et professionnels d'Android à Paris.

Le groupe est un facilitateur de rencontres, de connexions et d'opportunités qui :

- organise des conférences, des Apéro Android et est présent dans des salons (il réunit entre 100 et 130 personnes tous les mois),
- intervient dans des tables rondes, et émule plus globalement l'écosystème professionnel Android parisien et français.



Le groupe est géré par des membres actifs bénévoles tous professionnels ou passionnés Android. Le PAUG propose aussi des concours de développements, du contenu et de la documentation en ligne, une bibliothèque d'applications. Le PAUG veut être un acteur de l'écosystème Android sur Paris et sa région.

A noter dans vos tablettes :

Du 17 au 19 février, aura lieu l'Android DevCamp mondial 2012 avec plus de 40 localisations dans le monde. Pour ce premier hackathon international, la communauté du Paris Android User Group accueillera la version parisienne de cet événement gratuit et soutenu par les acteurs de l'innovation et des services mobiles: Proxima Mobile, Initiative Services Mobiles, Geny Mobile, EIT ICT Labs. Site web : <http://www.paug.fr/>

Février

Viens créer ton jeu !

Jusqu'au 6 février, inscris-toi à l'événement jeu du printemps. Epitech organise « Viens créer ton jeu ». En tout, 42 développeurs et passionnés seront sélectionnés pour créer en 48h un jeu vidéo complet. Encadré par le GameDev Lab d'Epitech avec le logiciel Unity 3D, spécialement personnalisé pour l'occasion, aucune notion de développement n'est nécessaire. L'envie et l'imagination sont suffisantes pour créer son jeu vidéo. A voir d'urgence : <http://www.vienscreertonjeu.fr/>



Février

Devcon de RIM

RIM organise sa première conférence développeur Blackberry Devcon Europe les 7 et 8 février à Amsterdam. Occasion pour discuter et voir l'avenir de RIM et des plateformes mobiles du constructeur. Les principaux thèmes des sessions sont le développement web, le développement natif, les technologies Adobe et Java. Bref tout ce qu'il faut savoir pour exploiter au mieux Blackberry et Playbook, en attendant les prochains systèmes prévus pour le 2e semestre de cette année.

Site : <http://www.blackberrydevcon.com/europe/content/sessions/topics>

agenda



► FEVRIER

Du 7 au 9 février, paris 17e Palais des Congrès, les **Microsoft TechDays**, le rendez-vous incontournable des développeurs professionnels <http://www.microsoft.com/france/mstechdays/>

Le 07 février 2012, CNIT Paris La Défense, **NIDays 2012**, l'événement annuel sur la conception, le test et le contrôle <http://www.nidays.fr/>

Le 16 février 2012, Cap 15 1/13 Quai de Grenelle 75015 Paris, **Nokia Developer Day** : une journée autour des technologies Windows Phone http://www.developer.nokia.com/fr/Dev_camp.xhtml

► MARS

Du 20 au 21 mars 2012, La Maison Internationale, 17 boulevard Jourdan 75014 Paris

Congrès Big Data Paris 2012

<http://www.bigdataparis.com>

► ETRANGER

Du 26 au 29 mars, USA, Reston, Virginie, EclipseCon
<http://www.eclipsecon.org/2012/>

Février

NIDays

Le constructeur National Instrument organise sa journée technique le 7 février. National Instrument est spécialisé dans l'instrumentation virtuelle et la conception graphique de système. De nombreuses sessions seront consacrées à la programmation et conception graphique. Ce sont d'ailleurs des briques NI qui animent les outils de programmation Lego Mindstorm NTX. Des ateliers seront proposés sur l'outil phare : LabView. Site : www.nidays.fr

Mars

Big Data à son event en France

Les 20 et 21 mars prochains, le Big Data fait son show à la cité universitaire internationale de Paris. De nombreuses conférences seront données : comprendre le Big data, sa complexité, comment la traiter, origines et formats des données, les solutions Big data du marché, etc. Hadoop sera bien entendu une des vedettes de la conférence. Réservez dès maintenant votre place. Site : <http://www.bigdataparis.com/fr-index.php>



II Appel aux groupes utilisateurs !

Programmez! s'ouvre à toutes les communautés. Notre volonté est de nous rapprocher des communautés et groupes utilisateurs. Chaque mois, Programmez! publiera l'agenda, les événements, les comptes-rendus des groupes utilisateurs. Mais sans vous, nous ne pouvons rien faire ! Envoyez-nous vos agendas, vos événements, vos propositions de comptes-rendus et d'articles ! Programmez! est le media idéal pour les communautés et les groupes utilisateurs ! - contact : ftonic@programmez.com



De la chimie à l'open source

Depuis plus de dix ans, Emmanuel Hugonnet conçoit et met en place des solutions JEE pour de grands comptes de l'industrie et des télécoms.

Cette expérience lui a permis d'avoir une vision globale du développement d'applications professionnelles et de maîtriser les technologies adaptées. Il a participé, durant plusieurs mois, à la réflexion d'Orange Labs sur la mise en œuvre de leur plateforme de développement et d'intégration des nouveaux services du Système d'Information. Depuis, il a rejoint l'équipe Silverpeas (<http://www.silverpeas.org>) afin de moderniser le processus de développement, ouvrir le code et reprendre l'architecture du logiciel. Il anime parallèlement l'AlpesJug (<http://www.silverpeas.org>), le plus enneigé des JUG français. Et pourtant, notre développeur n'était pas forcément prédestiné à l'être...

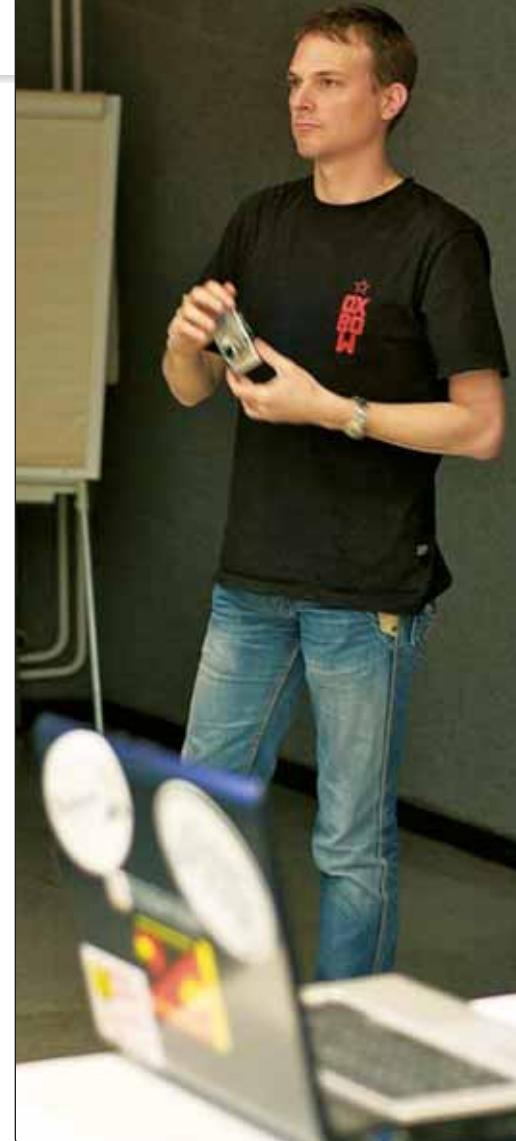
Comment es-tu tombé dans l'informatique et plus spécialement dans le développement ?

J'ai une formation de chimiste mais l'informatique m'a toujours plu : j'ai programmé sur ma vieille CASIO et sur mon Atari 520 ST. En 1999 quand je me suis retrouvé au chôma-

ge j'ai commencé à postuler dans l'informatique car on cherchait des scientifiques pour passer l'an 2000. J'ai eu la chance de rentrer dans une petite SSII parisienne qui se lançait dans Java. C'est comme ça que début 2000 je développais en Java. A l'époque on n'avait pas beaucoup de recul et c'était un vaste monde à défricher. J'ai donc commencé par m'auto-former puis proposer des supports de cours et des formations dans ma SSII aux développeurs internes et externes. En revanche, je ne suis pas un théoricien et j'ai besoin de toucher les choses pour les comprendre, donc je ne pouvais pas écrire ces cours si je ne manipulais pas pour de vrai les techniques. J'étais donc toujours les mains dans le cambouis.

Pour toi, qu'est-ce qui fait que l'on aime toujours et encore le développement, la technique ?

Ma formation initiale, en tant que chimiste, était déjà tournée vers la R&D, et donc découvrir de nouvelles techniques a toujours été moteur pour moi. Développer en infor-



matique c'est comme faire de la R&D en chimie mais avec des retours beaucoup plus rapides, donc c'est encore mieux. L'informatique et le développement sont en perpétuelle évolution, ça bouge tout le temps et on ne peut pas se reposer sur ses acquis. Tous les jours on doit se remettre à l'ouvrage et on apprend de nouveaux trucs. Bref, on n'a pas le temps de s'ennuyer.

Le principal point noir c'est que la technique n'est pas valorisée en France, je me souviens encore d'un entretien d'embauche où on me reprochait de demander le même salaire qu'un chef d'un projet qui aurait eu autant d'expérience que moi. En fait j'ai la chance de bosser dans une société fondée par un ancien développeur qui aime encore la technique et donc on y est bien. Ce n'est pas tout le monde qui peut parler WebDAV, multithreading et ontologie avec son patron pendant le repas de midi.

Tu as gardé un regard très geek : gadget, veille techno, c'est important pour ton job et ta passion ?

C'est devenu ma passion et c'est essentiel





pour mon job, comme je l'ai déjà dit, impossible de rester sur ses acquis, donc il faut toujours regarder tout ce qui se passe. L'avantage c'est qu'avec le temps et l'âge on devient meilleur pour choisir les sujets 'porteurs' et filtrer le bruit.

Etre développeur n'est pas toujours facile : pression, évolution constante, frustration des projets et des "chefs", c'est quoi pour toi d'être développeur aujourd'hui ? Le job a-t-il changé depuis tes débuts ?

La pression ne fait qu'augmenter et il devient de plus en plus difficile de faire bouger les choses. En 2000 c'était le boom de l'internet et on était vraiment des rois. Le monde était assez neuf et vierge et on partait dans tous les sens. Maintenant j'avoue que c'est beaucoup plus professionnel et qu'il y a moins d'enthousiasme. En plus on se retrouve souvent dans un environnement très contraint : choix des outils, de l'OS, pas d'accès à tout ou partie d'internet, avec une organisation de plus en plus pesante. C'est l'une des raisons pour laquelle je suis parti rejoindre une petite structure en 2008.

C'est chez les start-ups et les petits éditeurs que l'on trouve sa vraie place car les décisions techniques ont un vrai impact.

Quand je bossais en SSII j'étais souvent en régie dans de grandes structures où le choix est plus politique que technologique : expliquer à ton chef de projet que tu es beaucoup plus productif avec Netbeans sous Linux qu'avec Eclipse sous Windows c'est un vrai combat de tous les jours.

Surtout quand tu viens de lui casser la mise en page de son document de spécifications Office lorsque tu l'as édité avec LibreOffice.

Et en dehors du boulot, qu'est-ce que tu aimes faire ? Comment trouves-tu l'équilibre entre travail, vie privée, passion, famille ?

En dehors du boulot je m'occupe de l'Alpes-JUG qui organise une conférence mensuelle, sinon je suis en train de rénover ma future maison. C'est vrai qu'il est parfois difficile de décrocher, c'est pourquoi une activité physique permet de se vider la tête. Je profite de la magnifique piscine découverte d'Eybens l'été et de la neige (il y en a beaucoup qui vient de tomber) l'hiver. J'avais tendance à

comprendre une implémentation de pool de connexion qui fuyait et qui nous bloquait pour la release. L'avantage d'être sur un projet libre c'est que le source est toujours disponible et que tu peux travailler de chez toi comme au bureau. Même si toute l'équipe est française et autour de Grenoble, on a profité de la structure nécessaire d'un projet libre pour mettre en place du télétravail. Ça aide aussi pour l'équilibre, car on perd moins de temps dans les transports et on est disponible chez soi pour pas mal de petits trucs.



Le bureau d'Emmanuel

me laisser manger par mon travail et à traîner sur les blogs le soir à la maison, mais il est important de faire une pause sinon on fini par se griller et on n'arrive plus à réfléchir.

Peux-tu nous présenter ton quotidien en quelques mots ?

Tous les jours j'ai la chance de travailler sur un projet libre de portail collaboratif : Silverpeas. C'est un projet commencé en 1999 et que nous essayons de remettre au goût du jour en modernisant complètement l'architecture. On s'arrache parfois les cheveux car il faut se plonger dans les strates de code et essayer de comprendre ce qu'il devait faire. Par exemple je viens de passer deux jours à

Des conseils aux étudiants et dévs qui nous lisent ?

Comme beaucoup j'ai connu le doute après 5 ans d'expérience où l'on hésite à faire le choix de passer chef de projet pour évoluer, obtenir un meilleur salaire et plus de reconnaissance. Je savais que je ne ferai pas un bon chef de projet car la technique me plaisait trop et je me refusais à l'abandonner. J'ai remplacé le terme de développeur sur mon CV par architecte car cela permettait d'obtenir

un meilleur salaire ;-) et aussi un peu plus de respect et puis j'ai cherché une petite structure où on a encore de la liberté et où les développeurs ne sont pas des numéros interchangeables. J'ai aussi contribué à pas mal de projets open source dans ma carrière, ce qui m'a permis de rencontrer virtuellement et en chair et en os des développeurs chevronnés avec qui partager ma passion.

Propos recueillis par François Tonic

Chaque mois, nous faisons parler un développeur. Si tu veux participer, une seule adresse mail à retenir : ftonic@programmez.com

Le mois des cadeaux (offre limitée au 29 février)

Abonnez-vous jusqu'à **-50%**

et pour un **1€** de +,
recevez un "livre numérique" des Éditions ENI

Coupon d'abonnement page 97 et sur www.programmez.com/abonnement.php

Les métiers de l'internet : une profonde évolution des profils et des compétences

Les emplois de l'internet affichent un paradoxe : les demandeurs peinent à trouver un job, tandis que les entreprises peinent à trouver l'employé qui leur convient... parmi les centaines de candidats ! Ils souffrent de leur jeunesse, des métiers comme de celles et ceux qui l'exercent.

Les métiers de l'internet sont comme le média qu'ils alimentent : jeunes. Si l'on se réfère à trois dates - 1992 pour le premier navigateur (Mosaic), 1994 pour les premiers accès payants (opérateurs locaux américains), et 1996 pour les premières utilisations professionnelles – les plus anciens métiers du web ne s'exercent que depuis 15 à 20 ans ! Dans le même temps, les plateformes technologiques, les usages des internautes et les attentes des entreprises n'ont cessé de se créer, d'évoluer et de se transformer.

Le marché de l'emploi internet s'est également transformé, bien au-delà du recul du CV au profit des sites d'emplois. Les métiers du web se sont créés puis démultipliés, sans pour certains disposer de référentiels. Et surtout, ils se sont ouverts à une multitude de profils, avec généralement pour points communs la jeunesse, l'enthousiasme et une absence de qualification certifiée (les technologies évoluent plus rapidement que les processus de certification!), ce qui pour l'entreprise transforme le recrutement en une épreuve. Pour une offre d'emploi qui suit le processus officiel du recrutement, Pôle Emploi et annonces, les candidatures se chiffrent par centaines, mais les profils qui répondent aux attentes manquent à l'appel ! A moins que l'entreprise ne sache définir précisément puis repérer les compétences qu'elle recherche...

Dans pratiquement toutes les branches professionnelles, enseignants et formateurs ont suivi un cursus de formation dans les domaines qu'ils enseignent. De même, le chef d'entreprise possède traditionnellement des compétences dans la profession qu'il exerce et dans les métiers essentiels que sont le commercial et la comptabilité.

> Revoir les bases, fini l'entreprise de papa...

L'internet révolutionne cette vision de « l'entreprise de papa ». Les formateurs sont issus d'autres filières, en particulier l'informatique et la communication, tandis que le chef d'entreprise et ses cadres ne disposent d'aucune compétence sur les métiers de l'internet. Ainsi donc et pour la première fois, l'entreprise recrute à l'aveugle, sur des métiers qu'elle ignore, et pour des fonctions qui généralement ne disposent ni de budget ni de ressources en suffisance pour s'exercer à temps plein. La révolution n'est pas seulement dans l'entreprise, elle est également dans l'exercice des métiers ! A commencer par le recrutement. Le CV n'est plus reconnu. D'ailleurs, qui dans l'entreprise peut valider le profil annoncé sur le papier ? La petite annonce cède également le pas aux sites agrégateurs de profils d'employabilités potentielles. L'entreprise interroge le site d'emploi qui sélectionne et lui propose des profils qui répondent plus ou moins à son attente. Plutôt

moins d'ailleurs sur ces métiers ! Et puis, parmi les profils proposés, le demandeur n'est pas toujours sans emploi... C'est l'empire du réseautage. Bientôt le recrutement s'effectuera sur Facebook, Viadeo ou LinkedIn – les réseaux se voient aujourd'hui plutôt confier des missions de validation du cursus de l'individu et de découverte de sa personnalité -, sur le buzz plus que sur le bouche à oreille.

Les conditions d'exercice n'échappent pas non plus à la révolution. Les métiers de l'internet s'exercent aujourd'hui selon le mode de l'intérim, au rythme des projets, et à temps partiel. Un contrat pour un projet de quelques mois à raison de 2 jours par semaine dans telle entreprise, un simili CDI d'un jour par semaine dans telle autre pour entretenir et alimenter un site ou un blog... A ce propos, la législation sociale française n'est pas adaptée pour ce type d'emploi, qui impose aux salariés qui cumulent des emplois au contrat ou à temps partiel de disposer d'un employeur référent et de calculer le temps de travail cumulé, ou encore qui permet à certains comités d'intervenir sur les conditions d'exercice du poste de travail (comment considérer le bureau au domicile du téléviseur, dans le couloir ou sous l'escalier ?). Mais qui s'en préoccupe ? D'autant que l'employé peut y trouver son intérêt par la diversification de son parcours professionnel et de ses compétences. Sous réserve d'accepter

		Instructions
		Planning training programs
		Service desk
		Operation
		System operation
		Operations management
		Facility management
		Software
		Hardware
		Application software
		Middleware
		Basic software
		Business application package
		Business application system
		Security
		Systems management
		Common application infrastructure
		Database
		Network
		Platform
		Software product development
		Network service
		IT outsourcing
		Systems development
		Infrastructure architecture
		Integration architecture
		Application architecture
Consultant	IT Architect	Business function
		Industry
Sales	Project Management	Sales via media
		Product sales by visiting customers
Marketing	IT Specialist	Consulting sales by visiting customers
		Market communication
		Sales channel strategy
		Marketing management

Classification en cours des métiers de l'internet selon le projet European ICT Professionalism.

un mode de vie qui impose une forte capacité à s'organiser et à résister au stress, à intégrer voire gérer une équipe et à travailler seul. Pas d'antinomies dans cette énumération, juste la description de la pluralité des savoirs.

> Savoirs, savoir-faire, savoir-être

Le savoir-faire ne devrait pas être un problème, les forma-

tions existent, elles sont multiples et souvent de qualité. En revanche, le diplôme n'est pas le gage de trouver un emploi ! Car dans les métiers de l'internet, pour les entreprises, à moins de connaître voire de maîtriser des technologies et méthodologies, le savoir-faire qui figure sur le CV échappe généralement à leur compréhension. Le profil recherché doit alors plutôt répondre à des critères d'employabilité que de compétence. Le candidat saura-t-il comprendre les besoins mal exprimés de l'entreprise ? Saura-t-il intégrer une équipe déjà en place, ou au contraire saura-t-il travailler seul ? Ces questions, et de nombreuses autres, sont d'autant plus prégnantes que le candidat est jeune ! La génération Y n'a-t-elle pas la réputation d'être incontrôlable car elle prétend maîtriser les technologies nouvelles mieux que ses aînés ? Ce qui souvent est vérifié d'ailleurs... Le candidat est donc jugé sur son savoir-faire, mais également sur son savoir-être. Dans un recrutement, aujourd'hui, les deux états se conjuguent, alors qu'avant le seul CV, donc le savoir-faire, suffisait ! Qu'il s'agisse du chef d'entreprise, du responsable du projet ou du service, a fortiori de la DRH (direction des ressources humaines), le recrutement dans les métiers

de l'internet fait sa révolution et se base désormais sur des critères plus subjectifs et certainement moins formels : la capacité d'intégration du candidat, sa capacité de communication tant dans l'entreprise qu'avec ses clients, sa capacité à respecter la hiérarchie et les consignes, sa capacité à adhérer à la stratégie de l'entreprise. Ce que les anglo-saxons appellent en référence aux métiers de l'internet le « e-skills ». Et ici, point de critère de discrimination, tous les sexes et toutes les races sont acceptés. Seuls les handicapés, par la faiblesse de leur bagage scolaire (plus de 80% n'ont pas le Bac et le marché ne propose que trop peu de formations en environnement adapté), éprouvent des difficultés à accéder à ces métiers, et les entreprises du secteur à respecter les quotas.

> Se vendre et assurer son ROI

Dans les métiers de l'internet, le profil du candidat sera donc jugé sur l'évaluation du savoir-faire et du savoir-être. C'est tout du moins ce qui est affirmé, mais généralement le recruteur ne sait qualifier sérieusement ni l'un et/ou l'autre ! Dans ces conditions, il exercera sa sélection moins sur les qualités de l'individu qu'il aura su ou cru percevoir, que sur une échelle de

risque. La grande perdante de cette approche, c'est évidemment la compétence ! Un individu qui correspondra au poste ne sera pas retenu, car le critère subjectif du risque encouru par l'entreprise en le recrutant est dominant. Nombre de jeunes diplômés et qualifiés s'y cassent les dents à l'issue de leur cursus de formation. Et finalement personne ne s'y retrouvera, l'entreprise qui recrute en deçà de ses attentes et peine à alimenter sa valeur immatérielle, son savoir, et le recruté qui ne trouvera pas son équilibre.

Peut-on pallier cette problématique ? L'Observatoire International des Métiers de l'Internet, un projet européen, dont la base est alimentée par des chefs d'entreprise de toute l'Europe, travaille à l'élaboration de profils. Chacun d'entre eux indique les tâches associées aux métiers dans leur entreprise et les compétences principales recherchées lors du recrutement. L'entreprise comme le candidat disposeront ainsi d'un référentiel « métiers de l'internet » sur lequel chacun positionnera soit ses attentes, soit ses compétences.

> Et la formation ?

De nombreuses organisations misent sur la formation afin d'alimenter un vivier de candidats et de compétences dans lequel les entreprises vont de plus en plus piocher. D'autres, à l'image du Syntec Numérique, vont plus loin et visent la sensibilisation aux métiers du numérique dès l'école afin d'orienter les futurs étudiants à choisir un métier du numérique. La condition pour que candidats et recruteurs se retrouvent est que ces formations ne se satisfassent pas des seules technologies pour apporter également des méthodes, une organisation personnelle, du marketing, de la communication, de la vente. Mais il nous apparaît, pour

Référentiel métiers usuels de l'internet et rémunérations : fourchette haute

Source : Cabinet de conseil en RH Aravati - 2010

Fonctions techniques

Directeur Technique (CTO)	100K€
Responsable du développement	70K€
Responsable exploitation	60K€
Chef de projet technique	36K€
Développeur	32K€
Ingénieur Système	32K€
Ingénieur BDD	37K€
Administrateur réseau	40K€
Intégrateur	20K€

E-Commerce

Directeur marketing on line / e-commerce	120K€
Responsable marketing on line / e-commerce	70K€
Responsable acquisition traffic	65K€
Responsable fidélisation on line	70K€
Affiliate manager	48K€
Chargé de partenariats	48K€
Business Développeur Director	75K€
Business Développeur Manager	48K€

En agence

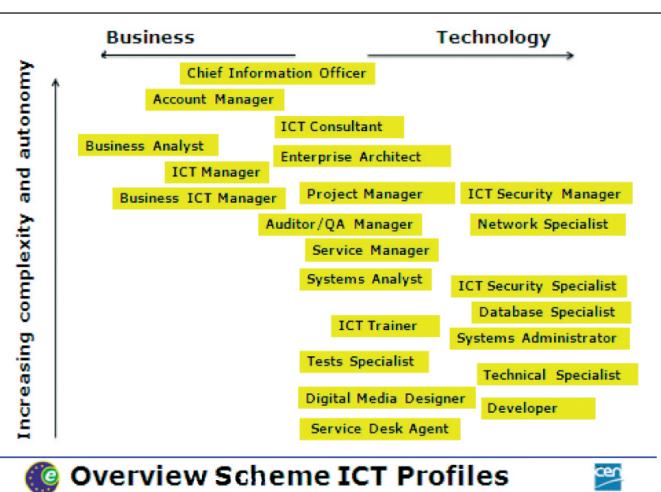
Chef de projet	38K€
Consultant /directeur de clientèle	50K€
Directeur conseil / directeur associé	91K€
Directeur artistique	40K€
Concepteur / rédacteur	48K€
Directeur de création	95K€
Planner stratégique junior	32K€
Planner stratégique confirmé	50K€
Directeur de planning stratégique	87K€
Chef de projet technique	46K€
Directeur de projet technique	63K€
Directeur technique	90K€

En régie

Chef de pub	45K€
Directeur de clientèle	75K€
Chargé d'étude	32K€
Responsable / Directeur Marketing régie	90K€
Traffic Manager	45K€

autant que ces projets dépassent justement l'état de projets, qu'il faudra du temps avant que l'entreprise ne dispose des outils référentiels qui permettent ces retrouvailles. En attendant, les candidats à l'exercice des métiers de l'internet devront démontrer leur capacité à intégrer l'entreprise et sa dynamique, et pourquoi pas démontrer leur capacité à assurer un ROI (retour sur investissement) sur leur poste. En deux mots : se vendre...

Yves Grandmontagne



Source : Direction Générale Entreprises et Industrie de la Commission Européenne.

La diversité des métiers de l'internet selon le projet European ICT Professionalism.

Ces entreprises qui recrutent

Le mois de décembre a été marqué par une hausse inattendue de la demande. Les recrutements n'ont pas marqué la pause annoncée. C'est la bonne surprise de la fin de l'année 2011.

Bonne pêche également pour ce début d'année, Progammmez a repéré pour vous plus de 1500 postes débutants ou confirmés, et des stages, à pourvoir immédiatement et durant l'année 2012. La prudence reste cependant de mise. Certaines SSII ont le vent en poupe et n'hésitent pas à le dire. C'est de bonne guerre, mais cela reste cantonné à des effets d'annonces pour des recrutements qui vont s'étaler sur toute l'année.

Surtout que dans le même temps, la grande majorité des éditeurs et des SSII ne s'affichent pas sur leur stratégie de recrutement. Et l'effet « triple A » sur l'économie ne s'était pas encore fait ressentir au moment où nous écrivions cet article... A suivre donc.

Le baromètre mensuel Hightechpros du marché de la prestation informatique affiche également une hausse de la demande sur la fin de l'année 2011, au point de « débuter 2012 sur une note optimiste ». Parmi les chiffres extraits du baromètre, nous noterons une hausse sensible de la demande sur le Clients-serveur (27%) et les Systèmes, réseaux et sécurité (15,7%). Et à l'inverse un net repli sur l'AS400 (44%) et le Mainframe (13%).

L'évolution en opposition de ces domaines n'est probablement pas fortuite !

> Ingénieurs, développeurs, commerciaux (50 postes)

L'éditeur **LANDesk Software**, spécialiste international de la gestion du cycle de vie des systèmes, de la sécurité du point d'extrémité et de la gestion des services informatiques, recrute des collaborateurs répartis sur l'ensemble de ses sites dans le monde : support technique, ingénieurs commerciaux, channel managers, territory managers, développeurs, responsables marketing produits ; etc.

www.landesk.fr

> Développeurs (5 postes)

Isagri, vénérable (créé en 1983) éditeur de solutions informatiques pour l'agriculture, la viticulture et les comptables, renforce ses équipes avec le recrutement de développeurs sur les technologies WPF, Silverlight, Visual Studio 2008 (.Net 3.5, C#), et SQL Server. www.isagri.fr

> Consultants et formateurs (20 postes)

La SSII **OXiane** recrute des ingénieurs d'affaires, des consultants formateurs, des ingénieurs et des architectes JAVA EE pour son siège situé à Boulogne Billancourt et ses filiales en Aquitaine, en Bretagne, à Montpellier et au Luxembourg. www.oxiane.com

> Managers et consultants (350 postes)

Présente en Europe et en Asie, la SSII **Micropole** recherche des profils de managers de projets, consultants, ingénieurs d'affaires et experts techniques pour le e-Business (MDM, e-commerce, dématérialisation, Réseaux Sociaux d'Entreprises...), la Business Intelligence et l'ERP (avec un focus sur les technologies SAP). Les postes sont à pourvoir en Europe, dont 300 en France. www.micropole.com

> Consultants experts (145 postes)

Damilo Consulting, société de conseil et de services en ingénierie informatique, recrute des consultants AMOA Finance, consultants ERP (SIRH), consultants Banque, consultants MOA/MOE, ingénieurs d'Etudes NTIC et ingénieurs d'Etudes. www.acial.fr

> Consultants (300 postes)

Solucom, cabinet de conseil en management et système d'information indépendant, s'est fixé comme objectif de devenir le premier cabinet de conseil indépendant en France. La première étape de son plan stratégique « Solucom 2015 » sera marquée par 200 à 300 embauches et 50 à 80 stagiaires. www.solucom.fr

> Experts infogérance et ingénieurs commerciaux (60 postes)

D.FI, intégrateur et prescripteur de technologies, annonce le recrutement en France (Nord, Pays de la Loire/Bretagne, Rhône Alpes, Aquitaine, Midi Pyrénées) de cadres, experts infogérance et commerciaux expérimentés, dont 20% de jeunes diplômés. dfinet.d-fi.fr

> Ingénieurs et grandes écoles (300 postes)

Umanis, SSII spécialisée en Business Intelligence, Web/IT, Infrastructures Informatiques, Gestion de la Relation Client et ERP, recrute sur Paris et Toulouse des profils débutants issus de grandes écoles de commerce et d'ingénieurs, et profils expérimentés venant de SSII ou d'éditeurs, pour des postes techniques (infrastructure, production), fonctionnels (AMOA, gestion du changement, ITIL), management (team leader, chef de projet, responsable opérationnel dédié) et commerciaux. www.umanis.com

> Ingénieurs (300 postes)

ITS Group, groupe de conseil et d'expertise pour le développement des infrastructures IT, recrute en Ile-de-France, pour ses six agences régionales et sa filiale au Benelux, principalement dans les métiers de l'ingénierie système, la production informatique et la gestion de projets. [# Y.G.](http://www.itsgroup.com)

Exclusif :
2530 postes à pourvoir.

Le cumul des recrutements de cette page et de l'interview ATOS (p36) atteint ce chiffre.

OFFRE D'EMPLOI : INTERVIEWS DE RECRUTEURS www.programmez/emploi.php

Formation

pour dirigeants et équipes IT



- ✓ Gérer le **TEMPS**
- ✓ **MANAGER** les projets
- ✓ COMMUNIQUER

www.know-formation.com

CONTACT :

Stéphanie Khalif-Vennat : stephanie@know-formation.com - Tél. 01 74 70 48 91 - Fax 01 41 39 00 22



Annick Riou, Directeur du Développement des RH.

ATOS : 1000 recrutements en 2012

Atos prévoit de recruter environ 1000 personnes en 2012 en France, sur ses 20 sites et proposera environ 600 stages. Annick Riou, Directrice du Développement des Ressources Humaines chez Atos, explique la stratégie de l'entreprise.

Vous annoncez 1000 recrutements pour 2012 ?

Nous sommes toujours dans une perspective de recrutement important avec une révision de ces prévisions début 2012 en fonction de l'évolution de notre environnement économique.

Portez-vous une part importante de votre recrutement sur les jeunes diplômés ?

Oui, plus de la moitié de nos recrutements concerneront les jeunes diplômés ; ils seront équitablement répartis entre l'Ile de France et les régions. Ce recrutement de jeunes diplômés permet de faire évoluer les juniors qui nous ont rejoints ces deux dernières années et qui vont pouvoir évoluer professionnellement.

Assurez-vous au sein de votre entreprise une transmission des compétences ?

Bien sûr, la transmission des compétences et des connaissances est un point fondamental, car c'est un élément clé dans la réussite et le pilotage des projets et des missions que nous menons pour nos clients. Dans le cadre d'une mission par exemple, le passage de relais au sein des équipes Atos doit se faire en toute transparence et sans aucune incidence sur le déroulement du projet. Par ailleurs, cette transmission des compétences est souvent accompagnée de formation : nous avions créé depuis plusieurs années des cursus métiers spécifiques à Atos.

Proposez-vous des formations ?

Nous disposons d'un catalogue de formation en ligne accessible à l'ensemble des salariés d'Atos . Ils peuvent faire leur demande en ligne et un processus de validation par le management est géré dans l'outil. De nombreuses formations sont proposées dans ce catalogue : des formations techniques et méthodologiques variées, des cursus

métiers (projet, management, commerce, ...), des formations linguistiques ou de développement personnel, etc.

Atos dispose également d'un centre de formation qui propose plus de 350 stages s'adressant à nos clients ainsi qu'aux salariés du Groupe. Les domaines couverts par Atos Formation sont : l'Informatique, le Management, la Gestion de projet, le Développement personnel, la Bureautique. Enfin, notre université, Atos University, qui est accessible à l'ensemble des 42 pays du groupe propose des cursus de management et commercial à nos hauts potentiels.

nités de mobilité géographique (nous sommes présents dans 42 pays et compsons une vingtaine de sites en France) ; mais aussi de mobilité fonctionnelle grâce à notre capacité à accompagner le développement des carrières et des talents (tutorat, formations, etc.).

Notre programme « WellBeing@Work/Bien être au travail » donne la priorité aux conditions de travail : télétravail, développement des outils collaboratifs, nouveaux modes de management, reconnaissance des talents et programmes de formation

Quelles sont les valeurs de l'entreprise ?

Nous sommes convaincus que les valeurs sont déterminantes pour la qualité de nos prestations. Les traduire au quotidien, c'est agir d'une même voix dans l'intérêt de nos collègues, de nos clients et partenaires. Nous invitons donc l'ensemble de nos salariés et ceux qui souhaitent nous rejoindre à réfléchir sur la mise en application de ces valeurs au quotidien, dans le cadre de leur travail : « Responsabilité, Confiance, Compétitivité opérationnelle, Sens du service, Innovation, Bien-être social et Excellence ».

Propos recueillis par Thierry Lellouche
Octobre 2011

“Le centre de formation propose plus de 350 stages”

Quel message adressez-vous à un candidat qui voudrait postuler chez vous ?

S'il a le sens de la relation client, le goût de l'innovation et du travail en équipe, il pourra trouver chez Atos un grand nombre d'opportunités de missions, de projets, dans des secteurs de marché variés ; cette diversité lui permettra d'acquérir une expérience riche et variée.

Nos équipes pilotent des projets passionnantes pour des clients de tous les secteurs d'activités (EDF, Thalès, EADS, Michelin, Renault, les principales banques françaises ainsi que le secteur public. Atos offre une grande diversité de postes et de missions qui permettent d'acquérir des compétences pointues et variées.

Nous offrons également de réelles opportu-

Plus de 2 000 ingénieurs développement chez Atos

- Plus de 2 000 ingénieurs développement en France, sans tenir compte de profils proches (intégrateurs, spécialistes des tests, ...), soit un peu moins de 15% des effectifs.
- Environ 1 500 Chefs de Projets, environ 10% des effectifs

Introduction à l'architecture CUDA

3^e partie

Dans la première partie de cette série consacrée à l'introduction à la programmation GPU (Graphics Processing Unit), nous avons présenté les différentes technologies et l'algorithme qui sera utilisé pour chaque technologie. Nous avons constaté les performances de cet algorithme à la fois en mode séquentiel et en mode parallèle sur un CPU (Central Processing Unit) de type quad-cœurs. Dans la seconde partie, nous avons présenté comment installer l'environnement de développement CUDA C sous Windows avec Visual Studio 2010. Nous avons aussi montré comment interroger une carte graphique compatible CUDA afin de récupérer ses caractéristiques techniques.

Dans cette troisième partie, nous allons vous présenter les fondamentaux de l'architecture CUDA afin de vous sensibiliser à la programmation GPU qui réclame une attention importante sur l'architecture matérielle. Nous allons pour cela implémenter avec CUDA l'algorithme présenté dans la première partie.

ARCHITECTURE CUDA

Si vous êtes néophyte en programmation CUDA, comprendre l'organisation des threads GPU d'une architecture CUDA est essentiel pour commencer à programmer vos propres algorithmes. Vous devez considérer les threads GPU comme à la fois plus nombreux, plus légers et plus rapides à lancer que leurs homologues CPU. Par exemple, la contention mémoire que peut provoquer un nombre très important de threads CPU a beaucoup moins de sens avec des threads GPU. Les threads GPU sont organisés d'une manière totalement différente de ceux d'un CPU. Les threads GPU sont répartis dans des structures de type **blocs**, qui sont elles-mêmes étagées dans une structure de type grille, [Fig.1]. De la même manière, plusieurs **grilles** peuvent co-exister si vous possédez plusieurs cartes graphiques. Les blocs de threads représentés ici par les lignes horizontales contiennent sept threads chacun. Il y a six lignes de **blocs de threads** sur la figure 1. Sur le plan de la mémoire, chaque **bloc de threads** possède un **contrôleur de registres** (rectangle orange) et une zone de **mémoire partagée** (rectangle rouge) accessible par tous les **threads du bloc**. Cette mémoire partagée est extrêmement rapide et elle peut être utilisée pour accélérer les performances de

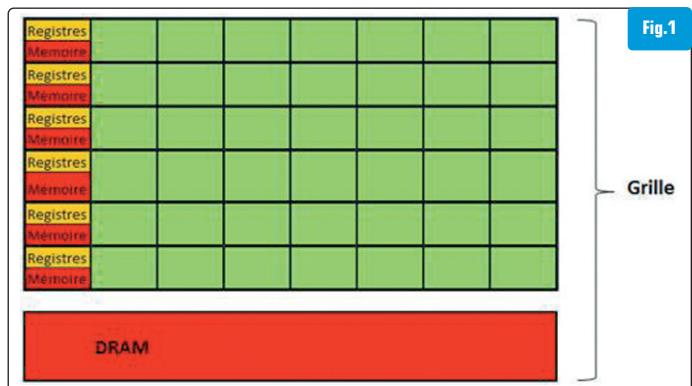


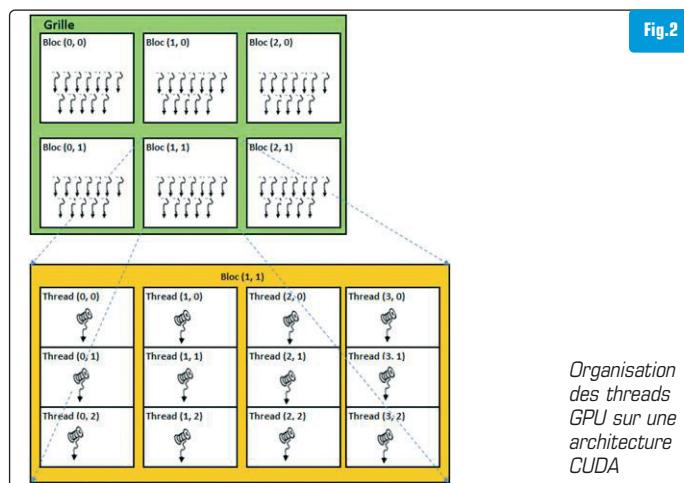
Schéma simplifié de l'architecture physique GPU

vos algorithmes. Une grille contient aussi de la **mémoire globale** (rectangle rouge DRAM) plus lente que la mémoire partagée, très simple à utiliser, mais surtout bien plus volumineuse. C'est à travers cette mémoire que les échanges de données se font entre votre host (thread CPU responsable du lancement du traitement GPU) et votre carte graphique. Le host alloue la mémoire nécessaire sur le GPU, puis transfère les octets recopier dans la mémoire allouée précédemment, mémoire host vers mémoire GPU et inversement dans le cas d'un retour du code GPU. Nous reviendrons plus en détail sur l'organisation mémoire dans un prochain article, mais pour l'instant passons à l'organisation des threads GPU.

ORGANISATION DES THREADS

Généralement, l'exécution d'un programme CUDA, se traduit par le lancement d'une ou plusieurs fonctions dites « noyau ». Ce nom peut vous sembler très technique, mais il signifie simplement que le code respectif de ces fonctions va s'exécuter sur la carte graphique et non sur les cœurs CPU de votre machine. Ces fonctions sont en charge de définir le nombre de threads souhaités pour l'exécution. Dès qu'une fonction noyau est lancée, une organisation de threads se met en place pour satisfaire la demande de calcul. Cette organisation se traduit par une grille contenant souvent des milliers voire des millions de threads.

Cette organisation physique peut vous sembler étrange, mais vous devez vous familiariser avec l'organisation atypique des threads GPU que l'on retrouvera sous une forme quasi identique en OpenCL ou en C++ AMP. En effet, pour programmer votre calcul GPU, vous allez devoir vous familiariser avec ce découpage structuré qui vous permettra progressivement de vous repérer facilement. Vous êtes en charge d'exprimer combien de **blocs et threads** sont nécessaires pour exécuter votre algorithme. Sur la [Fig.2], la grille verte contient six blocs indiqués via un numéro de ligne et un numéro de colonne.



Organisation des threads GPU sur une architecture CUDA

C'est ainsi que nous repèrerons les blocs dans nos fonctions noyaux. Chaque bloc contient 12 threads, les indices des threads permettent à votre code de savoir où il se trouve vis-à-vis de cette topologie de 72 threads. Enfin, notons que nous aurions pu représenter les blocs sous la forme d'un cube, car le système de coordonnées de blocs repose en réalité sur **3 dimensions**.

Si vous souhaitez paralléliser vos codes via **CUDA C**, vous devez tenir compte de cette organisation. En effet, calculer le nombre de threads par bloc et le nombre de blocs est nécessaire pour paralléliser votre traitement. Généralement votre traitement réclamera un nombre d'itérations supérieur à un bloc de threads. Dans ce cas, vous aurez besoin d'utiliser plusieurs blocs pour satisfaire l'exécution de votre algorithme. Dans notre exemple, le produit matriciel de référence engage des matrices de taille 1024 * 1024, ce qui réclame **1 048 576 threads** !

L'exécution de l'outil **deviceQuery.exe**, [Fig.3], présenté dans l'article précédent nous permet d'obtenir de nombreuses informations sur l'architecture CUDA courante.

Mais dans le cadre de cette introduction, nous allons nous attarder sur les éléments cercleés en rouge sur la *figure 3*. En effet, lorsque nous lancerons des calculs sur la carte graphique nous serons heureux de connaître le nombre maximum de threads par blocs, la taille maximum d'un bloc et la taille maximum d'une grille. Tout développeur **CUDA** doit tenir compte de ces informations pour paralléliser ses traitements.

COMMUNICATION ENTRE LE CODE CPU ET LE CODE GPU

Lorsque vous lancez un programme qui réalise un calcul sur GPU, vous utilisez malgré vous un thread système par défaut pour démarer votre application. Un thread CPU est à la fois nécessaire pour exécuter votre programme, mais aussi pour héberger le code noyau CUDA, avant son transfert sur la carte graphique. Ce code noyau sera exécuté par de nombreux threads GPU, qui participeront massivement à la réduction temps de calcul. Lorsque nous parlons de code GPU, nous parlons d'un code qui s'exécute sur l'architecture **CUDA**. En effet la programmation GPU distingue deux modes d'exécution : **host** et **kernel**. Le mode host, est le mode d'exécution par défaut de tout programme Win32, mais la programmation GPU permet d'exécuter votre algorithme sur votre carte graphique. Pour clarifier l'exécution d'un programme **CUDA**, je vous propose le schéma de la [Fig.4]. C'est la méthode *MatrixMultiplyCudaSquare* qui illustre l'essentiel des étapes décrites sur la *figure 4*. Cette fonction appartient au fichier « *MultMatrix.cu* » à télécharger avec cet article.

Fig.3

```

C:\ProgramData\NVIDIA Corporation\NVIDIA GPU Computing SDK 4.0\Bin\win64\Release\deviceQuery.exe Starting...
GPU Device Query (Runtime API) version <CUDART static linking>
Found 1 CUDA Capable device(s)
Device 0: "Quadro 2000M"
  CUDA Driver Version / Runtime Version          : 4.0 / 4.0
  Total Global Memory:                            1933 Mbytes (207998528 bytes)
  Total amount of constant memory:               65536 bytes
  Max dimension of a block:                      1024 x 1024 x 64
  Maximum size of a grid:                         2147483647 bytes
  Maximum memory pitch:                          512 bytes
  Maximum number of threads per block:           1024
  Maximum sizes of each dimension of a grid:    1024 x 1024 x 64
  Maximum memory pitch:                          512 bytes
  Texture alignment:                            256 bytes
  Concurrent copy and execution:                No
  Run time limit on kernels:                     No
  Support host page-locked memory mapping:      Yes
  Support host pinned memory:                   Yes
  Support concurrent kernel execution:          Yes
  Alignment requirement for Surfaces:           No
  Device has ECC support enabled:               No
  Device supports Unified Addressing (UVA):     No
  Device supports Unified Memory:                No
  Device supports Compute Model:                 1 > 0
  Default Compute Model:                        < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >
deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 4.0, CUDA Runtime Version = 4.0, NumDevs = 1, Device = Quadro 2
deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 4.0, CUDA Runtime Version = 4.0, NumDevs = 1, Device = Quadro 2
deviceQuery.exe test results...
Press ENTER to exit...

```

Exécution de l'outil *DeviceQuery.exe*

```

void MatrixMultiplyCudaSquare(float* A, float* B, float* C, int size)
{
    float* deviceA;
    float* deviceB;
    float* deviceC;
    size_t sizeAlloc = size*size*sizeof(float);

    checkCUDA(cudaMalloc((void**)&deviceA, sizeAlloc), «cudaMalloc»);
    checkCUDA(cudaMalloc((void**)&deviceB, sizeAlloc), «cudaMalloc»);
    checkCUDA(cudaMalloc((void**)&deviceC, sizeAlloc), «cudaMalloc»);

    checkCUDA(cudaMemcpy(deviceA, A, sizeAlloc, cudaMemcpyHostToDevice), «cudaMemcpy»);
    checkCUDA(cudaMemcpy(deviceB, B, sizeAlloc, cudaMemcpyHostToDevice), «cudaMemcpy»);
    checkCUDA(cudaMemcpy(deviceC, C, sizeAlloc, cudaMemcpyHostToDevice), «cudaMemcpy»);

    dim3 threadsPerBlock(BLOCK_SIZE, BLOCK_SIZE);
    dim3 numBlocks(adjust(size) / threadsPerBlock.x, adjust(size) / threadsPerBlock.y);

    MatrixMultiplySquare<<<numBlocks, threadsPerBlock>>>(deviceA, deviceB, deviceC, size);
    checkCUDA(«MatrixMultiply»);

    checkCUDA(cudaMemcpy(C, deviceC, sizeAlloc, cudaMemcpyDeviceToHost), «cudaMemcpy»);

    checkCUDA(cudaFree(deviceA), «cudaFree»);
    checkCUDA(cudaFree(deviceB), «cudaFree»);
    checkCUDA(cudaFree(deviceC), «cudaFree»);
}

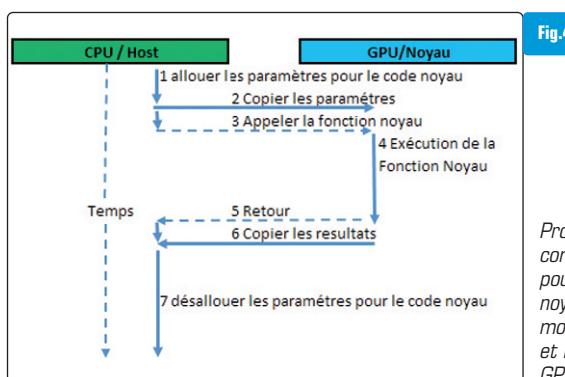
```

La fonction MatrixMultiplyCudaSquare en charge de lancer l'algorithme noyau

Reprenez chacune des étapes avec une illustration du fragment du code C associé.

1 - Définir et allouer les paramètres qui devront être copiés côté GPU pour les fournir à la fonction noyau

La transmission des paramètres à notre fonction noyau, doit passer par des variables dont l'allocation mémoire est issue de la carte graphique. Dans notre cas, nous devons recopier les 3 vecteurs représentant nos matrices A, B et C dans des variables préfixées par le mot **device**, deviceA, deviceB et deviceC. Pour allouer de la mémoire dans la mémoire globale de la carte graphique, nous utilisons la fonction **cudaMalloc()**, qui reprend le principe de la célèbre



Team Foundation Server : Tout pour la qualité logicielle



- Cycle de vie des applications : pourquoi y aller ?
- Les différents tests
- Comment automatiser les tests sous Visual Studio ?
- TFS Lab Management, l'art de maîtriser les tests
- Découvrez la prochaine version

édito

Qualité logicielle, vous ferez !

Une fois n'est pas coutume, nous parlerons de tests, de cycle de vie de l'application, de bonnes pratiques, bref tout ce qui fait la qualité logicielle. Impensable de déployer en entreprise ou auprès du public, une application, quelle qu'elle soit, sans avoir respecté un minimum de règles de développement, de suivi de projet, ou de bonnes pratiques sur le code, l'architecture et les tests.

La qualité logicielle ne s'improvise pas. Il faut utiliser les bons outils, les bonnes méthodes et règles au bon moment. Mais l'outillage n'est pas une finalité. Car durant des années, ces outils restaient sagement dans les armoires sans être utilisés. Il faut que les développeurs, chefs de projet, testeurs, utilisateurs, architectes, soient sensibilisés, formés à mettre en œuvre des suites logicielles telles que Team Foundation Server (TFS) de Microsoft.

Dites-vous que la qualité logicielle n'est pas négociable dans un monde informatique hétérogène, massivement distribué et connecté, où la moindre erreur de fonctionnement peut figer l'activité d'une entreprise toute entière. Le cloud computing impose une nouvelle rigueur ainsi que les applications smartphone, dans un monde de plus en plus souvent connecté aux serveurs. Mais au-delà de la technique pure, la communication entre chaque membre de l'équipe est vitale. La bonne information au bon moment prend tout son sens.

Dans ce cahier spécial, nous allons nous plonger dans TFS. Nous aborderons les bonnes pratiques (notamment Scrum), comment définir et mettre en place un plan de tests, comment automatiser les tests, utiliser toute la puissance de Lab Manager. Enfin, nous ferons un point sur la prochaine version.

François Tonic

sommaire

GAMME

A vous de choisir ! 3

UN ALM SUR MESURE

4

GESTION

Cycle de vie des applications : pourquoi y aller ? 6



D.R.

AGILITE

Visual Studio et l'agilité 8

QUALITE

Collaboration, partage, communication : les clés d'une équipe agile et de qualité 11

Tests unitaires, d'interface utilisateurs, de charge : une palette d'outils pour maîtriser votre qualité logicielle 12

L'automatisation de tests sous Visual Studio 15

Planifier et organiser les tests avec Microsoft 18

Améliorer la qualité logicielle avec Team Foundation Server 2010 19

Intégration continue et tests avec TFS Lab Management 20

FUTUR

Aperçu de la prochaine version 23

Les principales nouveautés VS 11 sur le cycle de vie 24

A vous de choisir !

Avec Visual Studio (VS) 2010, Microsoft a rationalisé l'ensemble de la gamme afin de la rendre plus compréhensible. D'autre part, pour répandre l'usage de l'ALM, TFS est disponible par défaut avec plusieurs éditions de Visual Studio 2010 :

Editions de Visual Studio 2010		TFS
Professional avec MSDN		Oui
Premium avec MSDN		Oui
Ultimate avec MSDN		Oui
Test Professional avec MSDN		Oui

TFS peut être acheté séparément si vous n'utilisez pas VS 2010 (version antérieure ou outils non .Net). Précision importante : TFS 2010 est disponible uniquement version 64-bit. D'autre part, selon la version de VS 2010, le niveau de fonctionnalités dans les tests, ALM... diffère beaucoup. Voici un petit tableau explicatif :

	VS 2010 Professional avec MSDN	VS 2010 Premium avec MSDN	VS 2010 Ultimate avec MSDN	VS 2010 Test Professional avec MSDN
Debugging et diagnostics	●	●	●●●	
Outils de tests	●	●●	●●●●	●●●
IDE	●●●●	●●●●	●●●●	●
Développement pour bases de données	●	●●●●	●●●●	
Plateforme de développement	●●●●	●●●●	●●●●	
Architecture et modélisation		●	●●●●	
Lab Management			●●●●	●●●●●
TFS	●●●●	●●●●	●●●●	●●●●
Contrôle de source	Oui	Oui	Oui	Oui
Gestion des workitems	Oui	Oui	Oui	Oui
Build	Oui	Oui	Oui	Oui
Reporting et BI	Oui	Oui	Oui	Oui
Guide de planning agile	Oui	Oui	Oui	Oui
Team Explorer	Oui	Oui	Oui	Oui
Gestion des cas de test			Oui	Oui

● faible support, ●● support moyen, ●●● bon support, ●●●● support total

Un déploiement à multiples couches

Le déploiement d'une plateforme TFS implique le déploiement des éléments suivants :

- application logique, données, et les clients tiers pour Team Foundation
- localisation physique des serveurs
- Team Foundation Build et la fabrication des build
- Team Foundation Server Proxy

TFS implique un déploiement d'un environnement de données complet pour gérer les données projet, les sources, le reporting. Il se décompose en 4 couches :

- TFS Configuration : base stockant les catalogues de ressources et les informations de configuration de TFS.
- TFS Warehouse : base de données stockant les données des rapports
- TFS Analysis : base multidimensionnelle stockant les données agrégées provenant des équipes projets.
- Databases for team project collections : chaque collection d'une équipe projet possède sa propre base dans laquelle sont stockées les données.

La mise en place et la gestion des bases de données TFS (déploiement, maintenance, mise à jour...) est une des tâches sensibles dans l'administration de son TFS. La configuration après l'installation de TFS se fait dans le Configuration Center. L'installation et la configuration ont été grandement améliorées par rapport à la version 2008.

Si vous disposez d'un annuaire d'entreprise, cela facilitera grandement la mise en place des droits d'accès et la déclaration des utilisateurs. Ce centre de configuration permettra de créer la collection TFS dans la base de données. La collection permettra d'y déposer vos projets. Pour toute la partie collaborative, accès web, partages, Sharepoint est nécessaire. Le déploiement d'un TFS prendra plusieurs jours. En cas de doute sur l'architecture à adopter, consulter un expert TFS, un intégrateur certifié.

Définir son infrastructure matérielle et logicielle

Comme nous l'avons vu plus haut, TFS se compose de différents outils et couches. Il faut disposer de TFS, Visual Studio, SQL Server, Windows sur le poste client, Windows Server sur

les serveurs (avec IIS pour lancer les services web nécessaires).

Idéalement, dans une infrastructure à montée en charge et pour assurer une fiabilité de l'ensemble, 1 serveur applicatif = 1 serveur physique. Ainsi, la base de données, le build, TFS en lui-même seront à déployer sur des serveurs différents. Il faut au moins disposer de deux serveurs physiques :

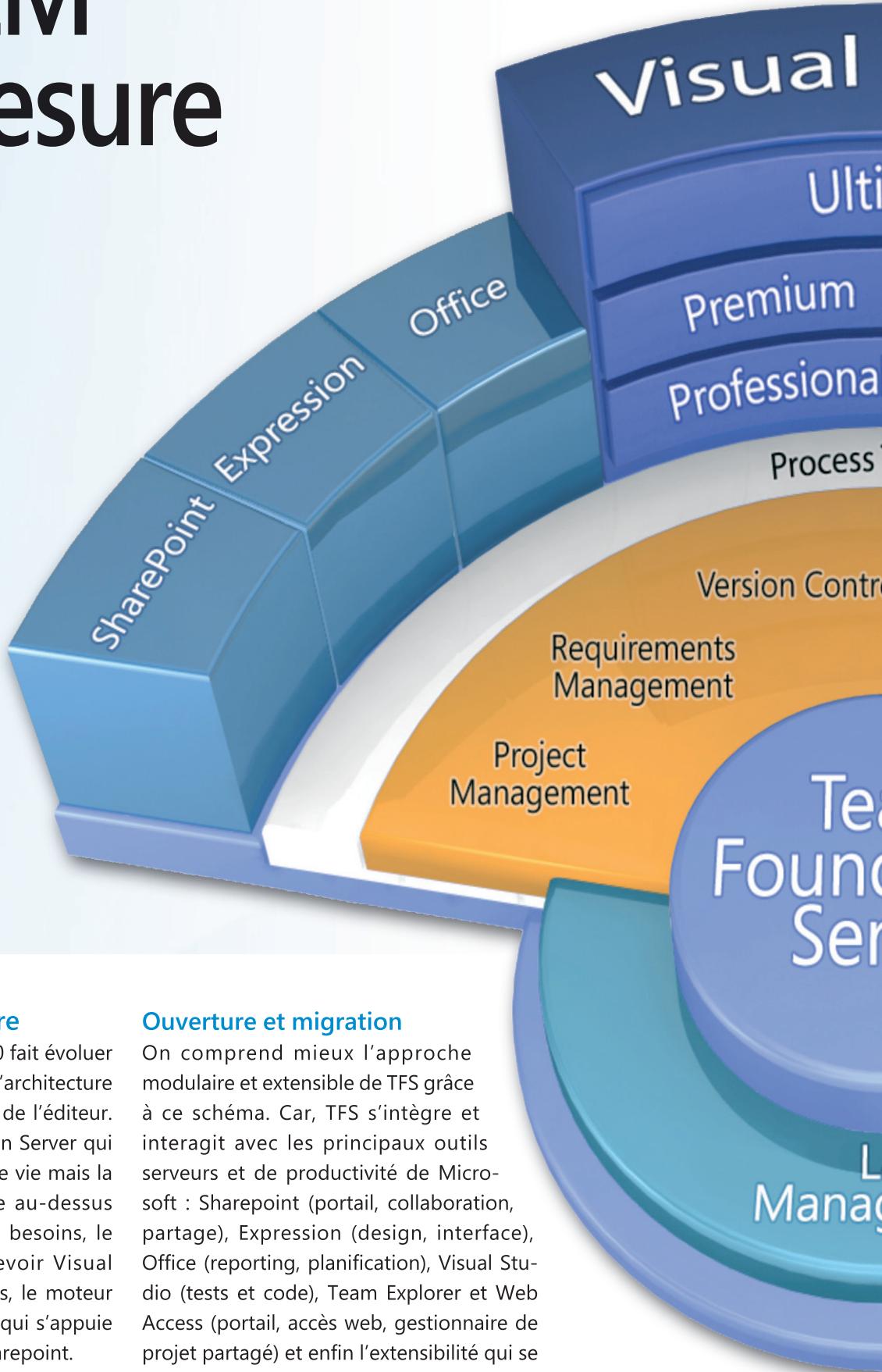
- 1 pour Windows Server, la base de données, TFS...
- 1 pour le build.

Ensuite il faudra tailler votre réseau et le nombre de serveurs selon le nombre d'utilisateurs et la charge attendue. En effet, vous ne dimensionnez pas de la même façon une infrastructure TFS pour 15-20 utilisateurs et une infrastructure pour 200 ou 2000 utilisateurs !

Au-delà du matériel, il ne faut pas négliger les ressources : mémoire vive, stockage, nombre de processeurs. Pareillement, sur les postes de travail des développeurs sur lesquels les outils de développement seront déployés (que ce soit Visual Studio ou Eclipse). Pour les autres utilisateurs (architecte, direction, testeurs), les exigences matérielles seront moindres.

Un ALM sur mesure

Avec sa gamme 2010, Microsoft a fait le ménage à la fois dans Visual Studio et ses outils ALM, de tests et de versionning. La gamme Visual Studio par exemple a été réduite de moitié afin de la rendre plus lisible.

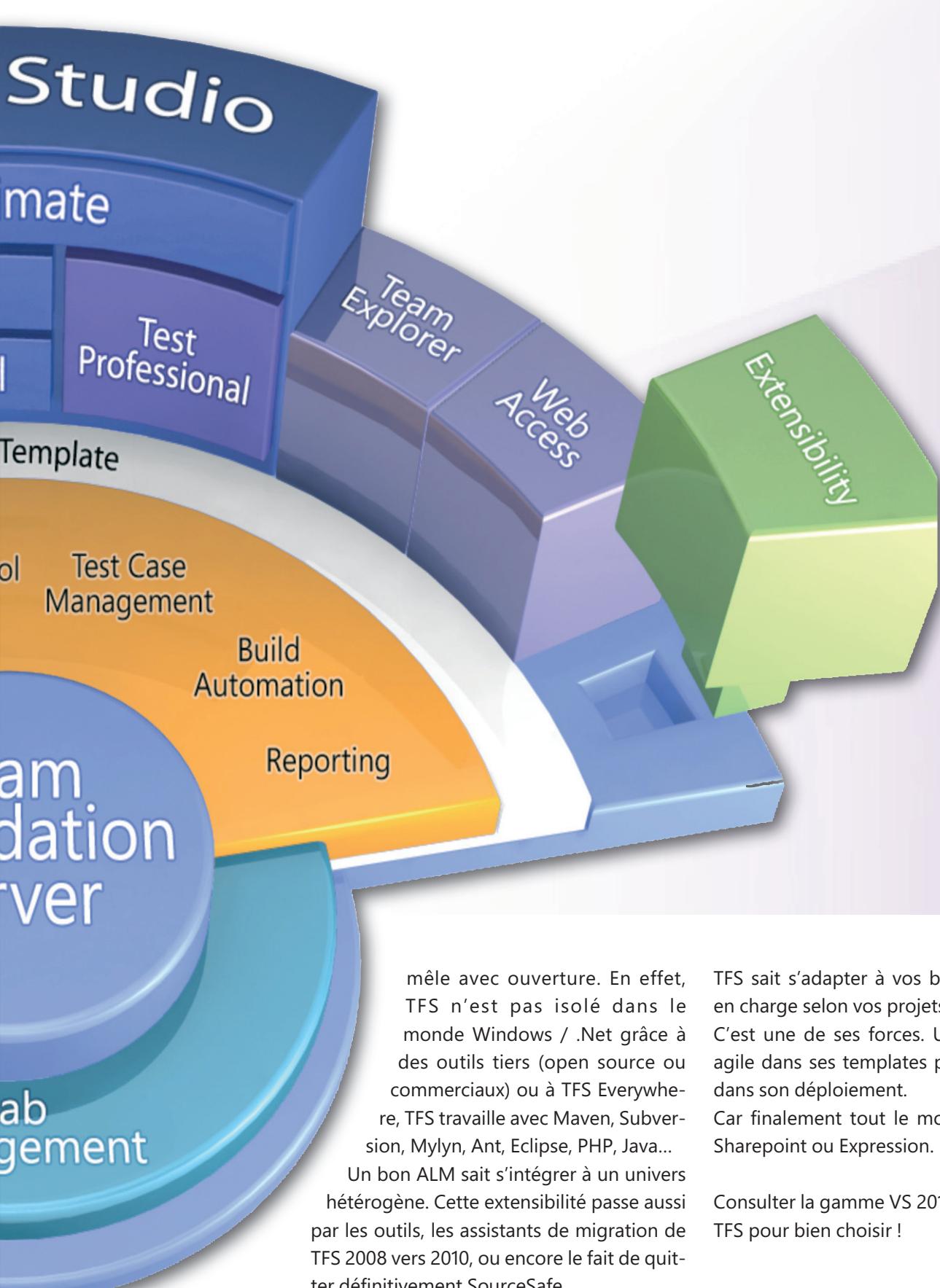


Une approche modulaire

Concernant TFS, la version 2010 fait évoluer l'approche fonctionnement et l'architecture globale de la plateforme ALM de l'éditeur. Le cœur reste Team Foundation Server qui constitue le moteur de cycle de vie mais la réelle valeur ajoutée se situe au-dessus avec la gestion projet et des besoins, le contrôleur de version (au revoir Visual SourceSage), les outils de tests, le moteur de build et la partie reporting qui s'appuie sur les derniers outils Excel, Sharepoint.

Ouverture et migration

On comprend mieux l'approche modulaire et extensible de TFS grâce à ce schéma. Car, TFS s'intègre et interagit avec les principaux outils serveurs et de productivité de Microsoft : Sharepoint (portail, collaboration, partage), Expression (design, interface), Office (reporting, planification), Visual Studio (tests et code), Team Explorer et Web Access (portail, accès web, gestionnaire de projet partagé) et enfin l'extensibilité qui se



mèle avec ouverture. En effet, TFS n'est pas isolé dans le monde Windows / .Net grâce à des outils tiers (open source ou commerciaux) ou à TFS Everywhere, TFS travaille avec Maven, Subversion, Mylyn, Ant, Eclipse, PHP, Java...

Un bon ALM sait s'intégrer à un univers hétérogène. Cette extensibilité passe aussi par les outils, les assistants de migration de TFS 2008 vers 2010, ou encore le fait de quitter définitivement SourceSafe.

TFS sait s'adapter à vos besoins et monter en charge selon vos projets.

C'est une de ses forces. Un ALM doit être agile dans ses templates projets mais aussi dans son déploiement.

Car finalement tout le monde n'utilise pas Sharepoint ou Expression.

Consulter la gamme VS 2010 et les fonctions TFS pour bien choisir !

Cycle de vie des applications : pourquoi y aller ?

Le déroulement incertain des projets informatiques, que ce soit sur les aspects délai, budget, périmètre, ou encore qualité serait-il un fait avéré, une tautologie ? Nous allons tenter de répondre à cette question dans le cadre de cet article en analysant les tenants et les aboutissants d'un projet informatique et de démontrer comment l'ALM apporte une réponse pertinente au travers d'un ensemble riche de méthodes et outils.



LES PRINCIPALES CAUSES D'ÉCHECS PROJETS

Les raisons majeures qui sont à l'origine des échecs projets sont de deux ordres : Génériques et opérationnelles. Les causes génériques sont : **Le Facteur « Time to Market »** toujours plus agressif entraînant des rythmes projets difficilement soutenables et des dérapages dans les délais qui statistiquement constituent une des premières causes de conflit sur projet

Des besoins métiers toujours plus complexes côté client pour assurer la compétitivité, induisant de la complexité côté implémentation technique

Une évolution incessante et très rapide des technologies et des méthodes de développement apportant une surcouche de complexité supplémentaire dont tout le monde se serait bien passé.

Ceci étant dit, l'exercice qu'il est intéressant de mener est ce que l'on appelle une « Root Cause Analysis ». L'idée ici étant de déterminer quels sont les facteurs qui vont contribuer le plus fortement au fait que les projets, au plan opérationnel, soient en difficulté.

Tout le monde connaît ces principales causes d'échecs, simplement, la difficulté va consister à les prioriser, nous allons donc proposer cette liste, qui bien évidemment n'est pas exhaustive :

- Gestion de Projet Lacunaire
- Manque de Réactivité Face au Changement
- Manque de communication / collaboration, information disséminée
- Spécification incomplètes ou surréalistes
- Mauvaise gestion des risques

Gestion de projet lacunaire

La gestion de projet reste un élément clé dans la réussite d'un projet. Cette activité va permettre

de s'assurer de la satisfaction du client qui est un des objectifs majeurs de tout projet par la capacité à communiquer au travers d'un certain nombre de rapports sur l'état d'avancement, la qualité, le budget et le planning en vue de détecter tout écart et enclencher si nécessaire des actions correctives au plus vite afin de réaliser le projet avec les attentes du client et les objectifs stratégiques de l'entreprise.

Manque de réactivité face au changement

Entre le moment où un projet démarre et sa livraison, beaucoup de changements vont intervenir, c'est inévitable. Cet état de fait est particulièrement impactant dans des contrats de type forfait dans lesquels les intégrateurs s'engagent à délivrer un périmètre en s'appuyant sur le fameux cycle en V, voire Waterfall (Casque), rendant encore plus compliqué tout changement, du fait de la lourdeur des processus. Tout changement doit au contraire être accueilli favorablement comme avantage compétitif pour le client.

Manque de communication, information disséminée

Pour faire simple, la communication est la clé de voûte d'un projet informatique, pour comprendre comment ce point est fondamental en projet, on parle souvent du syndrome de la Tour de Babel. La tour de Babel a été le premier grand projet de l'humanité ayant subi un échec, il y a près de 5000 ans.

Ce projet, surréaliste en termes d'exigences au demeurant, a échoué à partir du moment où les bâtisseurs se mirent à parler des langues différentes et donc ne plus communiquer efficacement, la leçon de cette histoire nous a démontré la nécessité qu'ont les parties pre-

nantes de se parler, de se comprendre pour réaliser de grands projets avec succès.

Spécifications incomplètes ou surréalistes

La gestion des spécifications constitue indéniablement une des principales sources de problèmes sur les projets lorsque l'on interroge des parties prenantes. Tout le monde le sait et pourtant la situation continue. Démarrer un projet avec un périmètre insuffisamment défini ou surréaliste est une ineptie qui va précipiter l'équipe en charge du projet dans des difficultés de tout instant.

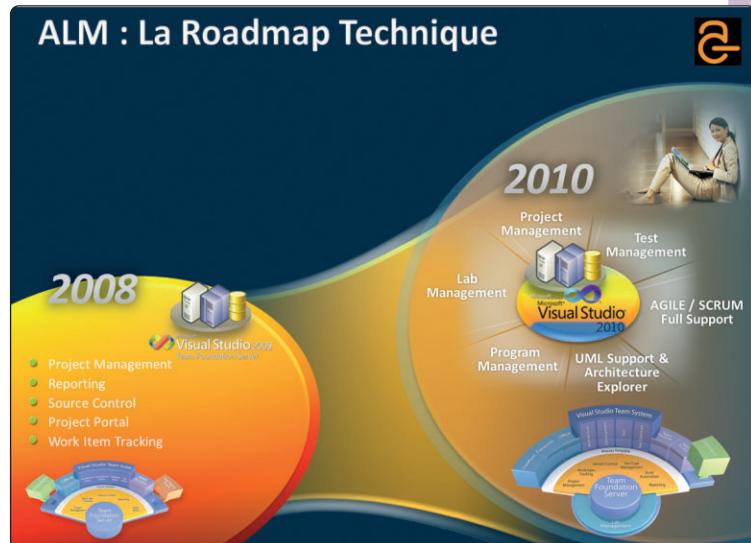
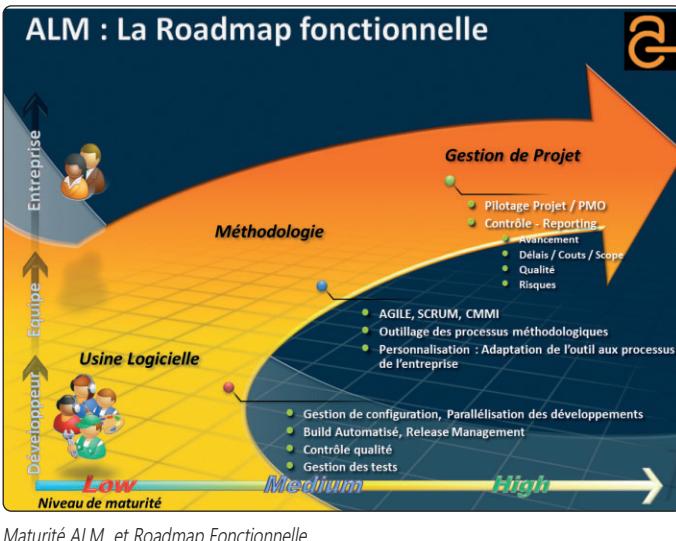
Mauvaise Gestion des risques

Qui gère ses risques sur un projet ? A cette question suit inéluctablement une réponse évasive. Ceci est pourtant une erreur stratégique. Chaque membre de l'équipe projet est conscient des risques potentiels et pourtant, ils ne sont pas gérés et s'imposent comme une fatalité.

LA SOLUTION ALM DE MICROSOFT : VISUAL STUDIO 2010 / TEAM FOUNDATION SERVER 2010

La plateforme ALM Microsoft TFS 2010 fournit un ensemble d'outils très riche fonctionnellement, permettant d'adresser l'ensemble des activités liées au cycle de vie du développement d'une application de la définition des exigences jusqu'aux phases de Test et Déploiement avec un support riche pour la méthodologie et la gestion de projet.

Ci-contre la Roadmap fonctionnelle de l'ALM TFS 2010 apporte un grand nombre de fonctionnalités à haute valeur pour la réalisation de projets dont :



Un support hiérarchique pour les Work Items :

Cette fonctionnalité permet une véritable activité de Gestion de projet avec le support d'un WBS Projet (Work Breakdown Structure qui représente l'arborescence hiérarchique des livrables d'un projet), ceci permet par exemple de gérer une arborescence de type Projet / Phase / Module / Exigence / Tâche en mode CMMI ou bien User Story / Task pour un Carnet de produit en mode Agile ou SCRUM, et de se synchroniser avec un fichier MS Project ou encore avec Project Server. TFS 2010 intègre de plus un mécanisme de Roll up permettant la consolidation hiérarchique de données horaires ou financières.

Une Plateforme de Test : Avec Test & Lab Center, Microsoft propose une offre complète pour des équipes de test professionnelles, la plateforme est entièrement intégrée à TFS 2010 et propose un client dédié aux testeurs fonctionnels avec des outils de planification de campagnes de test, un outil d'exécution des tests incluant des possibilités d'automatisation et de gestion de paramètres de tests pour améliorer la productivité et la fiabilité lors de la phase d'exécution. Avec cette plateforme, Microsoft adresse le syndrome bien connu de la non-reproductibilité des bugs en fournissant un mécanisme de collecte d'artefacts contextuels au test qui sont téléchargés automatiquement dans les données lors de la création d'un bug dont : les étapes détaillées du test avec les résultats, le Recording Video de la session de test, les données IntelliTrace (débogueur intelligent Microsoft) permettant au développeur de connaître le code fautif avec les données source de l'exception, la pile d'appel (Mini dump mémoire) et également des données de type Event Log et Informations Système. De plus avec Lab Center, Microsoft fournit une plateforme de Provisioning d'environnement pour les tests avec des capacités de

déploiement des binaires applicatifs et d'exécution de tests de manière automatisée unitaire ou fonctionnelle. Cette dernière fonctionnalité améliore considérablement la productivité et la qualité sur les projets car il est bien connu que le Provisioning et la maintenance d'environnements d'exécution pour les tests est coûteuse en temps et répétitive, tout ce que l'on veut éviter sur un projet.

Le Support de l'UML : Des cas d'utilisation ou tout type de diagramme UML peuvent être créés avec ce module et surtout on notera la possibilité de les relier à n'importe quel autre artefact projet.

Une Plateforme de Build : Une nouvelle version de Team Build basée sur le moteur dernière génération Microsoft Workflow Foundation 4.0 s'impose comme un outil exceptionnel pour l'ensemble des concepts d'intégration continue pour une usine logicielle. Cette nouvelle version permet en plus des options standard d'Analyse de qualité du code, d'exécution de listes de tests unitaires avec le calcul de la couverture de code, de personnaliser simplement un processus de Build avec par exemple le rajout d'activités de déploiement en fin de Build ou encore la génération de release note automatique.

Program and Portfolio Management : Le produit vient en standard avec une intégration avec l'EPM Project Server 2010 ou 2007 offrant des fonctionnalités de consolidation niveau entreprise pour la gestion des ressources, de Programmes et ou Portfolio au-dessus de TFS 2010 qui représente le suivi opérationnel projet par projet. Un des facteurs clé de TFS est un ensemble très complet de rapports fournis en standard qui correspondent à plus de 80% des besoins de Reporting.

Par exemple, le rapport ci-dessous nommé « **Requirement Overview** » qui donne un vue complète du statut d'un projet avec les exi-

gences du projet à gauche, et en regard, l'avancement, le reste à faire, le nombre de tests écrits par exigence, l'état et le résultat de l'exécution des tests et les bugs ouverts et résolus. On notera la notion de roll up des données du niveau exigence jusqu'au niveau projet.

Un autre exemple de rapport est le Build Success Over Time montrant la stabilité du processus de Build avec une légende pour le niveau de qualité de chaque Build en fonction du résultat de la compilation mais également des tests et de la couverture de code jour après jour.

Pour finir

En conclusion, la mise en œuvre d'une plate-forme ALM adresse réellement l'ensemble des besoins du cycle de vie projet avec un support fort d'activités en mode Collaboratif, un socle pour la méthodologie de type Agile, Scrum ou CMMI, des fonctions de Contrôle de version, la Gestion des Builds, des rapports et tableaux de bord, un centre de qualité logicielle et un ensemble d'outils connectés adapté aux différents rôles présents sur le projet, leur permettant de mieux collaborer et communiquer en partageant la même information avec leurs outils du quotidien. Des fonctions de gouvernance niveau entreprise peuvent également appuyer une gestion de portefeuilles projets de l'identification d'une opportunité jusqu'au déploiement en production. Enfin, au plan technique, la plateforme ALM de Microsoft est interopérable avec différents types de langages et technologies permettant d'élargir son périmètre d'adoption et surtout de répondre aux exigences d'entreprises exploitant un large panel de technologies Microsoft et non Microsoft.

Philippe Puschmann

Architecte senior – capability group application development, Avanade

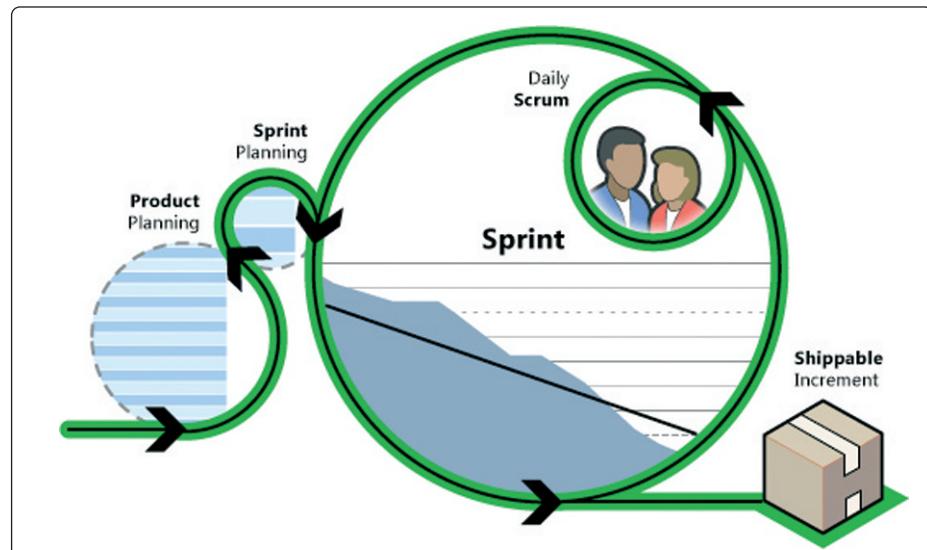
Visual Studio et l'agilité

Visual Studio est décliné sous différentes versions qui permettent chacune de couvrir différents niveaux de ce que l'on appelle la « Gestion du cycle de vie d'une application » (soit Application Lifecycle Management ou ALM pour les intimes). La version la plus complète, Visual Studio Ultimate avec MSDN Ultimate, couvre la totalité du cycle de vie du développement d'une application, démarrant de la spécification des différentes fonctionnalités à l'aide de User Stories ou de Product Backlog, jusqu'aux tests fonctionnels des utilisateurs, en passant par des rapports sur le taux d'avancement de l'itération en cours, etc. L'offre Visual Studio 2010 est une gamme de produits qui vont permettre à chaque acteur d'un projet de développement d'interagir avec un outil adapté...

Mais avant de poursuivre, il est nécessaire de définir et clarifier un terme régulièrement entendu/utilisé concernant la gestion de projets en environnement Microsoft : Le produit Team Foundation Server (abrégé TFS) est un service qui va permettre de centraliser toutes les informations manipulées au travers des différents outils qui composent la suite Visual Studio. Il est très commun de faire l'amalgame avec TFS et l'ALM bien que le premier soit un outil qui permet de mettre en pratique le second.

S'il est possible d'utiliser ces différents outils pour une gestion de projet classique, ceux-ci prennent tout leur sens lors qu'ils sont utilisés pour suivre une méthodologie de type Agile. En effet de nombreuses étapes spécifiques à cette dernière (détaillées tout au long cet article) y sont simplifiées et optimisées.

La matière première de cette méthodologie, ce sont les différents éléments de travail (appelés Work Items) qui seront manipulés tout au long du projet. Le découpage du projet se fera très souvent par fonctionnalités définies dans des User Stories (ou Product Backlog en Scrum).



Les grands principes d'organisation de Scrum

Celles-ci seront réparties dans des itérations, chaque itération correspondant à une livraison d'un lot de fonctionnalités. À ces fonctionnalités seront associées des tâches (Task) à réaliser, des cas de tests (Test cases) qui permettront de s'assurer de leur fonctionnement, ou encore différents bugs qui seront trouvés tout au long du cycle de vie de l'application.

Le suivi du cycle de vie de ces différentes tâches possède également un rôle important dans la prise de décision et la priorisation des différentes actions à effectuer. Pour cela un tableau de bord, disponible au travers d'un portail SharePoint, présente de nombreux rapports de suivis simples, auxquels s'ajoutent des rapports plus complexes générés à partir d'un cube OLAP. Parmi ces derniers, on pourra trouver des rapports mettant en forme des courbes, présentant l'état d'avancement des Work Items au cours des différentes itérations ou Sprints. L'un des plus notables et des plus connus est le rapport

de Burndown qui permet de visualiser le comparatif de la courbe idéale et réelle de l'avancement des tâches d'une itération.

Autre sujet très important en développement Agile : S'assurer fréquemment de l'intégrité des livrables produits, soit faire ce que l'on appelle de l'intégration continue. Pour cela, la partie Team Build de TFS se chargera de compiler et de tester régulièrement le code source du projet (à des dates précises, voire à chaque modification du code source). Cela permettra également de livrer à intervalles réguliers une version des exécutables qui certes ne possèdera pas toutes les fonctionnalités de l'application finale, mais sur laquelle les clients finaux pourront émettre des critiques et vérifier les fonctionnalités.

Intimement liée à l'intégration continue, la partie Test de TFS joue un rôle important dans la vérification de la qualité logicielle et de la conformité du projet d'équipe avec les attentes des clients. De nombreux outils pour faire de la modélisation, de l'analyse de code ou encore des tests unitaires permettent de vérifier le projet d'équipe très régulièrement (Ex. à chaque compilation locale ou serveur).

L'importance des retours clients a également

été prise en compte en mettant à disposition l'outil Test Manager qui permet la mise en place de tests fonctionnels et facilite leur automatisation.

Élément non spécifique à la méthode Agile, le versioning du code source est également géré dans cette offre.

Le Process Template (Modèle de processus)

Si tous ces composants de Team Foundation Server aident à la mise en place de certains processus de la méthodologie Agile, ils peuvent posséder un comportement similaire ou différent selon le projet d'équipe. Tout cela est défini dans un regroupement de fichiers XML appelé Process Template. Ce dernier, qui sera différent suivant la méthodologie choisie, définira le comportement des différentes parties de TFS et pourra être personnalisé. On pourra ainsi y trouver les itérations du projet, les workflow de Build qui seront disponibles pour l'intégration continue ou encore quelques rapports de base. C'est également au travers du Process Template que seront définis les différents Work Items du projet. Ces derniers étant la pierre angulaire de la gestion de projet avec TFS, il est très important d'utiliser des Work Items qui sont pertinents selon la méthodologie choisie. Il sera alors possible d'en ajouter de nouveaux, d'en supprimer ou même de personnaliser ceux déjà présents dans le Process Template de base.

Les caractéristiques d'un Work Item se trouvent dans une « Work Item Type Definition » dans laquelle sont définis :

- Le nom, le type des différentes données stockées, si celles-ci doivent être reportées dans

les rapports du cube OLAP (exemple : Title, AssignedTo, RemainingWork, Status)

- L'affichage graphique de la fiche du Work Item (Affichage sous Visual Studio et web)
- Le workflow du Work Item : Les différentes manières de le passer d'un état à un autre (exemple : Fermer une tâche, rouvrir un Bug, mettre à jour un cas de Test)

Un Process Template Agile est fourni par défaut lors de l'installation de Team Foundation Server 2010 (accompagné d'un Template CMMI) et un nouveau Process Template Scrum sera disponible de base dans la nouvelle version de TFS 11 (également disponible dans la galerie des extensions de Visual Studio) (**Fig. 1**).

Ces deux Process Template se différencient principalement par leurs différents types de Work Items.

Le Process Template Agile est inspiré de la méthode Scrum mais reste plus large que ce dernier. Il sera ainsi possible d'utiliser quasi n'importe quelle méthode de gestion de projet avec celui-ci. On parlera ainsi de User Story (Récits utilisateurs) exprimant des besoins qui vont être regroupés temporellement dans des itérations. Elles contiendront des tâches sur lesquelles il faudra saisir le temps passé, l'estimation d'origine, etc.

Le Process Template Scrum quant à lui est plus spécifique à la gestion de projet utilisant la méthodologie Scrum. De nouveaux types de Work Items y sont ainsi définis et de nouveaux rapports y font leur apparition. On parlera alors cette fois-ci de Product Backlog qui seront regroupés temporellement dans des Sprints. Pour implémenter ces premiers, des éléments

tâches seront toujours utilisés, mais cette fois-ci, seul le temps restant sera nécessaire.

Un outil pour les gouverner tous, et dans TFS les lier

Tous ces éléments mis à disposition par TFS permettent d'implémenter les bonnes pratiques de la méthodologie Agile. Mais cette méthodologie, aussi efficace soit elle, ne peut être mise en place de manière convenable si elle n'est pas bien outillée, et demande ou perturbe trop les différents intervenants du projet. C'est pourquoi Visual Studio ALM fournit différents outils, adaptés aux rôles de chacun afin de leurs permettre d'accéder à ces informations centralisées depuis leur environnement de prédilection.

La nouvelle version de Visual Studio actuellement disponible en « Developer Preview », a vu son site web Team Web Access enrichi de nombreuses fonctionnalités spécifiques à la gestion en mode Agile. Souvent délaissé dans les versions précédentes au profit du Portail d'équipe SharePoint, celui-ci revient au centre de la gestion de projet d'équipe et s'adresse à tous ses participants, incluant de nouvelles interfaces de saisies:

- Les Sprints (ou Itérations) y sont maintenant planifiés directement à coup de Drag and Drop et une notion de début et de fin a été ajoutée à ceux-ci pour contextualiser les différents rapports générés.
- Il est possible de spécifier le temps de travail disponible pour chacun des développeurs lors du Sprint. On pourra alors y préciser le nombre de jours par semaine que chaque participant du projet y consacrera afin de pouvoir répartir de manière la plus pertinente possible les différentes tâches qui constitueront le Sprint.
- Il existe également une interface de gestion des tâches en mode Post-it, bien connue des habitués de la méthodologie Agile. On pourra tout simplement y Drag and Droper les différents Post-it pour leur faire changer d'état (**Fig. 2**).

• Le rapport de Burndown est dorénavant disponible directement depuis le tableau de Post-It du Team Web Access. De plus, il est maintenant directement mis à jour lors des modifications effectuées sur les Work Items et ne nécessite plus d'attendre la reconstruction quotidienne du cube OLAP.

Bien entendu, le client Team Explorer pour Visual Studio, principalement destiné aux développeurs, est toujours de la partie. Celui-ci se voit ajouter de nouvelles interfaces d'interaction avec TFS, en plus de bénéficier d'un nouveau design beaucoup plus clair et intuitif.

STORIES	TEAM MEMBERS	
NEW 33 h	ACTIVE 22 h	CLOSED 10 h
Etienne 26 h	Implémenter le drag n'drop 12 Etienne	Gérer le changement d'état dans TFS 4 Etienne
Florent 23 h	Créer l'accès aux données via l'API 5 Florent	Créer le formulaire de visualisation 10 Florent
Léonard 16 h	Créer le formulaire d'édition 10 Léonard	Lier l'accès aux données avec l'API TFS 8 Florent

QUALITÉ LOGICIELLE

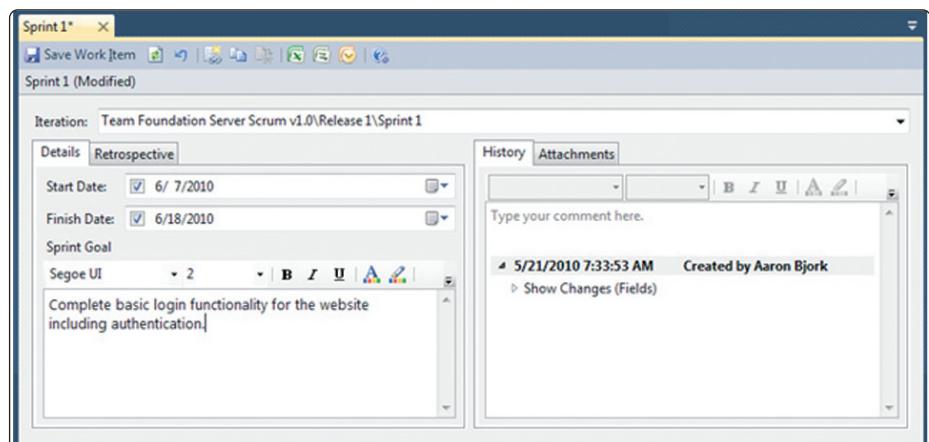
Il reste le moyen le plus complet de manipuler les différents éléments de TFS : Gestion du code source et des Work Items, création et visualisation des Builds serveurs, liens vers le portail d'équipe et le site de Reporting, accès aux documents du site SharePoint, tout y est.

Une nouvelle section nommée My Work permettra de visualiser et modifier en un clic l'état des différents Work Items (ex. Tâches ou bugs) qui nous sont assignés. Il sera alors plus intuitif pour le développeur de sélectionner directement depuis Visual Studio les tâches sur lesquelles il décide de travailler, avant même de commencer à modifier le code source.

Un nouveau Work Item verra également le jour, nommé Code Review. Ce dernier facilitera l'ajout d'un processus de revue de code lors du développement du projet d'équipe. Les développeurs pourront alors, en lieu et place de réaliser un archivage de leurs modifications de code source, décider de créer une nouvelle demande de revue de code et d'y spécifier un autre développeur. Celui-ci sera alors chargé de regarder, commenter puis valider le code source qui lui sera envoyé. Bien évidemment, les demandes de revue de code apparaîtront automatiquement dans la section My Work des développeurs.

Pour les chefs de projet, MOA, etc. le couple Microsoft Excel et Microsoft Project est toujours disponible et permet de manipuler les différents Work Items, Product Backlog, Tâches et autres Bugs. Il est ainsi possible de sauvegarder ses feuilles personnalisées et de les synchroniser à volonté avec le serveur TFS.

Microsoft Project rajoute une couche supplé-



Un work-item dans VS Scrum 1.0

mentaire à Excel : Il permet de visualiser plus facilement la planification des différents Work Items et introduit une notion de dépendance entre ceux-ci.

Un petit nouveau fait également son apparition dans la nouvelle version de TFS 11 : la possibilité de faire du Storyboarding à partir de Powerpoint. En effet, ce dernier étant initialement très pratique pour créer rapidement une interface, il aura suffi de rajouter quelques options à celui-ci pour faire de cet outil connu un outil utile pour montrer l'enchaînement de différents écrans. Il sera alors possible de créer au travers de l'onglet Storyboarding ce que l'on appelle des Shapes (Formes), de les enregistrer puis de les réutiliser à travers divers écrans de l'application. Enfin, Microsoft Test manager s'occupera, comme son nom l'indique, de gérer les différents aspects des tests de votre projet d'équipe. Il sera toujours aussi pratique d'y définir et vérifier

fier des tests concernant les différentes fonctionnalités de votre application, puis de les automatiser au travers d'une Build serveur pour éviter les régressions.

Sa grande force réside dans sa capacité à récolter de précieuses informations de la part de n'importe quel utilisateur (Enregistrements, commentaires, Work Items, etc.)

S'il était assez simple de définir manuellement les scénarios des différents cas de tests fonctionnels de votre projet d'équipe, Team Foundation Server 11 apporte une nouvelle fonctionnalité appelée « Tests exploratoires ». Ceux-ci vont permettre de tester l'application sans scénario prédéfini, puis de créer automatiquement, par exemple lors de la rencontre d'une erreur, un bug ainsi qu'un cas de test associé. Ces derniers possèderont comme scénario, afin de reproduire ce bug ou vérifier ce test, les différentes actions enregistrées durant le test exploratoire.

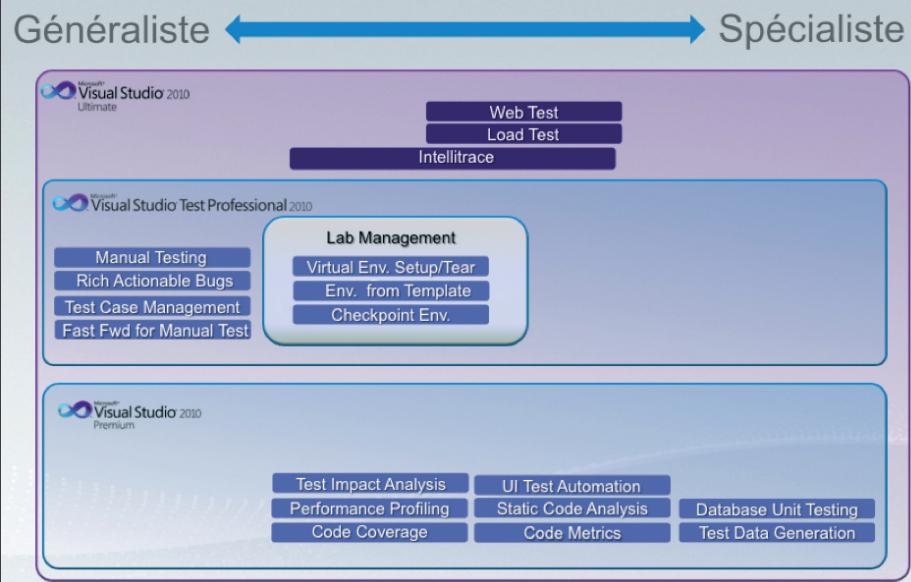
Une nouvelle fonctionnalité de Feedback a été mise en place. Il sera alors possible de lancer une campagne de Feedback à partir de TFS et d'inviter chacun des participants à tester les différentes fonctionnalités implémentées durant un Sprint défini puis de fournir son Feedback. Celui-ci pourra comporter un enregistrement vidéo voire audio, des notes écrites et des Screenshots ainsi qu'un système de notation de la fonctionnalité testée.

Actuellement en Developer Preview, la nouvelle version de TFS met encore plus l'accent sur l'agilité en mettant en avant le Process Template Scrum et en fournissant des outils de plus en plus perfectionnés pour chacun des membres du projet d'équipe, clients et testeurs compris.

Vivien Fabing

Consultant ALM chez Infinite Square
<http://blogs.developpeur.org/vivien>

Visual Studio 2010 v.s. Activités du testeur



Collaboration, partage, communication : les clés d'une équipe agile et de qualité

Un projet « moderne » doit s'appuyer sur une suite logicielle capable de fournir des services à la fois verticaux et transverses. Transverses, dans le sens où cette suite sert de fondation technique, de référentiels à l'ensemble des outils utilisés par la ou les équipes. Verticaux, car chaque catégorie d'intervenant (technique, architecture, métier, utilisateur, direction) a accès aux fonctions et outils nécessaires pour mener à bien sa mission, son travail.

Décloisonner pour mieux travailler ensemble

Une entreprise, qu'elle soit grande ou petite, possède différents départements (finance, direction, commercial / marketing, informatique...) avec différentes équipes. Encore aujourd'hui, ces départements travaillent en silo et en communication limitée avec l'environnement. Une plate-forme ALM va contribuer à décloisonner les équipes, les personnes pour assurer le succès des projets. Sans partage, sans collaboration, le projet ira à l'échec ainsi que l'ALM. Pour cela, il faut faire travailler ensemble trois grandes entités : la demande (utilisateur, client, direction, le métier), l'informatique (équipe technique) et la partie déploiement (administrateurs réseaux, responsable projet...).

Cependant, n'oublions pas que l'outillage ne fait pas tout. Il faut que les équipes et l'ensemble des intervenants soient sensibilisés à la démarche ALM et qualité logicielle. Or, le changement des habitudes et le cloisonnement constituent des remparts à la démarche ALM.

Préparer soigneusement son ALM

Avant tout déploiement d'une suite de cycle de vie / qualité, il faut impérativement préparer la gestion du changement, faire participer les

équipes, informer, former, expliquer. Cette phase amont est cruciale car elle conditionne l'utilisation de l'ALM au quotidien.

Un projet ALM ne s'improvise pas, cette approche bouleverse l'organisation des équipes et de l'entreprise.

Il est tout d'abord important d'auditer et de définir vos besoins, les outils déjà utilisés, le patrimoine applicatif à maintenir (et donc son code) induisant une future migration et de définir un organigramme précis des équipes. A partir de cet audit, vous pouvez définir les besoins réels et moduler l'ALM en conséquence. Il n'est pas utile de déployer un ALM complet si les besoins se limitent au contrôle de version, à la génération et l'intégration continue.

Il ne faut jamais oublier qu'un ALM doit être lui-même agile, c'est-à-dire modulaire et capable de s'adapter à vos besoins, aux exigences de chaque projet. D'autre part, il ne se déploie pas d'un coup de baguette magique, mais progressivement. D'abord sur des équipes réduites pour des projets non critiques. La montée en charge doit s'étaler sur plusieurs mois afin que les administrateurs ALM et les équipes adoptent les outils et acquièrent les nouveaux réflexes. Avec un modèle trop rigide de sa qualité logicielle on prendra un risque d'échec de sa plate-

forme ALM. Bref, une solution ALM n'est pas la solution à tous vos problèmes, ni un objectif ultime. Un ALM ne servira à rien si l'organisation, les méthodes de travail ne changent pas. Et il ne faut pas que votre ALM soit isolé du reste du système d'information, sa réussite dépend de son intégration et de la bonne volonté de tout le monde !

Développeurs, testeurs, utilisateurs, chef de projet, direction... même combat !

Nous avons l'habitude de dire : avoir la bonne information au bon format et au bon moment. Un ALM est comme un gros référentiel traitant et stockant diverses données. La plateforme traite et cible les informations selon le profil du collaborateur. Ainsi, un développeur aura des besoins techniques (code, description des bugs, planning des itérations, information sur les build, etc.), alors qu'un testeur demandera surtout le bugtrack, la traçabilité des corrections, l'accès aux modules de tests, une vue de planning plus haute que le développeur, etc.).

La collaboration entre tous les membres des équipes est donc primordiale, même si chaque profil a son proche planning, ses propres démarches. L'ALM gérera en interne tout cela.

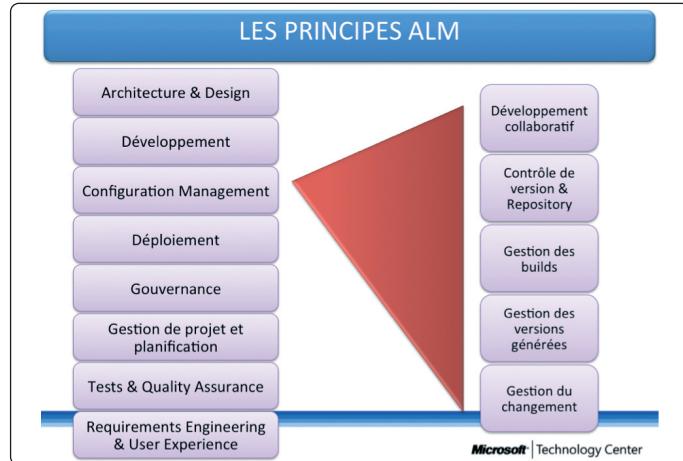
La plateforme met à la disposition des intervenants une communication élargie : messagerie, messagerie instantanée, VoIP, gestion de la présence, partage documentaire, suivi des documents, gestion des itérations, etc.

Vous devez tendre vers le cercle vertueux.

A RETENIR

Une solution de gestion de cycle de vie de l'application doit :

- fluidifier les processus, les flux de données, d'information entre les équipes et chaque personne
- aligner le métier / la logique business et la technique
- fournir la bonne information à la bonne personne, au bon moment
- établir un référentiel unique et commun à tout le monde
- créer un cercle vertueux dans la vie du projet de l'expression des demandes à la maintenance



Tests unitaires, d'interface utilisateurs, de charge : une palette d'outils pour maîtriser votre qualité logicielle

La qualité d'un logiciel passe obligatoirement par sa conformité par rapport aux spécifications. Qu'il s'agisse de spécifications techniques ou fonctionnelles, les créateurs de logiciels doivent garantir cette conformité, afin de démontrer la qualité de leur production.

Heureusement, les outils dont disposent les équipes projets sont de plus en plus performants, et accessibles. Nous allons passer en revue dans cet article les très nombreuses possibilités qu'offre la plateforme d'industrialisation logicielle de Microsoft : Visual Studio 2010 et Team Foundation Server 2010.

Tester d'accord, mais tester quoi ?

Le logiciel doit être conforme à différents niveaux :

- Conformité des traitements : les algorithmes et les traitements sont-ils correctement implémentés, et les fonctions renvoient-elles le bon résultat ?
- Conformité des scénarios d'utilisation : l'enchaînement des écrans, le libellé des messages, des boutons et des liens, la présence et les propriétés des contrôles sont-elles conformes aux spécifications ?
- Conformité de l'apparence de l'interface utilisateur : les couleurs et les images sont-elles correctement définies, les contrôles sont-ils placés au bon endroit ?
- Conformité des temps de réponses : les pages apparaissent-elles après un délai acceptable ?

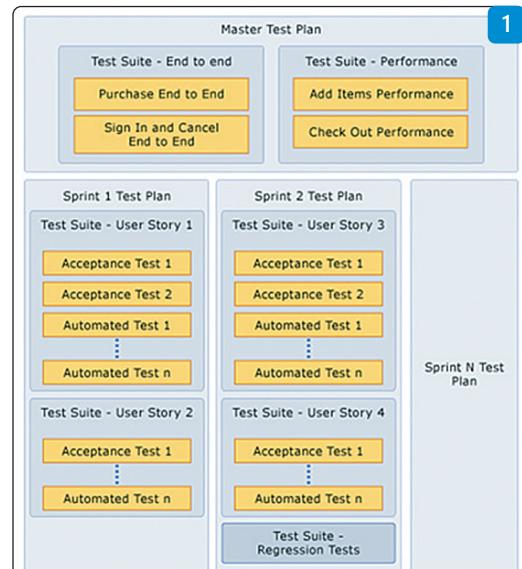
- Aptitudes à la montée en charge : les performances sont-elles toujours conformes lorsque de nombreux utilisateurs sont connectés et utilisent l'application ?

L'enjeu d'une plateforme d'industrialisation logicielle est de fournir aux équipes projets une solution adaptée pour tester chacun de ces niveaux, soit de manière automatique, soit manuellement.

Mais la réponse à la qualité ne se situe pas uniquement dans l'outil ! Encore faut-il aborder le test avec méthodologie, en sachant placer l'effort de test sur les zones réellement à risque, avec une approche pragmatique permettant de soutenir l'effort de test dans la durée. Une application conforme à la première version aurait rapidement une image détestable si les versions suivantes fourraient de bugs... La stratégie de test d'une application va donc permettre de définir la manière dont l'application va être testée, en combinant les différents types de tests (**Fig. 1**).

Au début était le test unitaire...

Le premier élément à garantir dans une application est le respect des règles de traitements. Ceux-ci sont généralement implémentés dans



Exemple de plan de test

la couche Business de l'application, sous forme de méthodes de service. Chaque méthode de service peut ainsi être validée par un test, qui consistera à injecter un jeu de données d'entrée à cette méthode, et de comparer la réponse de la méthode au résultat attendu. Il s'agit du principe du test unitaire, car il s'intéresse à la seule fonction, sans s'occuper de son environnement (**Fig. 2**).

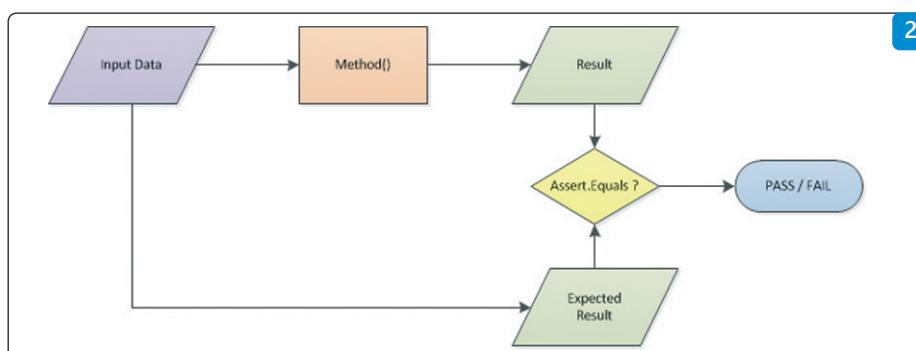
Avec Visual Studio 2010, la mise en place de test unitaire est extrêmement simple. Il suffit de se placer sur la fonction que l'on souhaite tester, d'effectuer un clic droit, et de sélectionner

Créer des tests unitaires (**Fig. 3**).

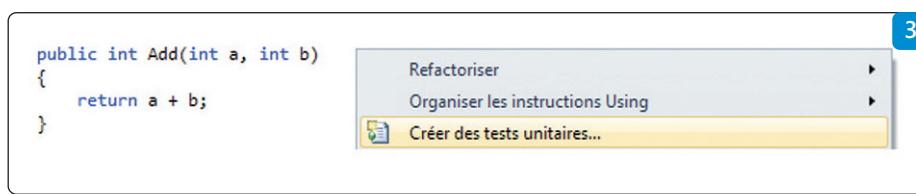
Quelques validations plus tard, un nouveau projet est ajouté à notre solution : il s'agit d'un projet de test, qui permettra de gérer le code source de test de notre application.

Une méthode de test a été créée, et nous n'avons plus qu'à renseigner les paramètres d'entrée, et le résultat attendu (**Fig. 4**).

Le test apparaît alors dans la fenêtre d'Affichage des tests, d'où il est possible de l'exécuter : le résultat est conforme (**Fig. 5**).



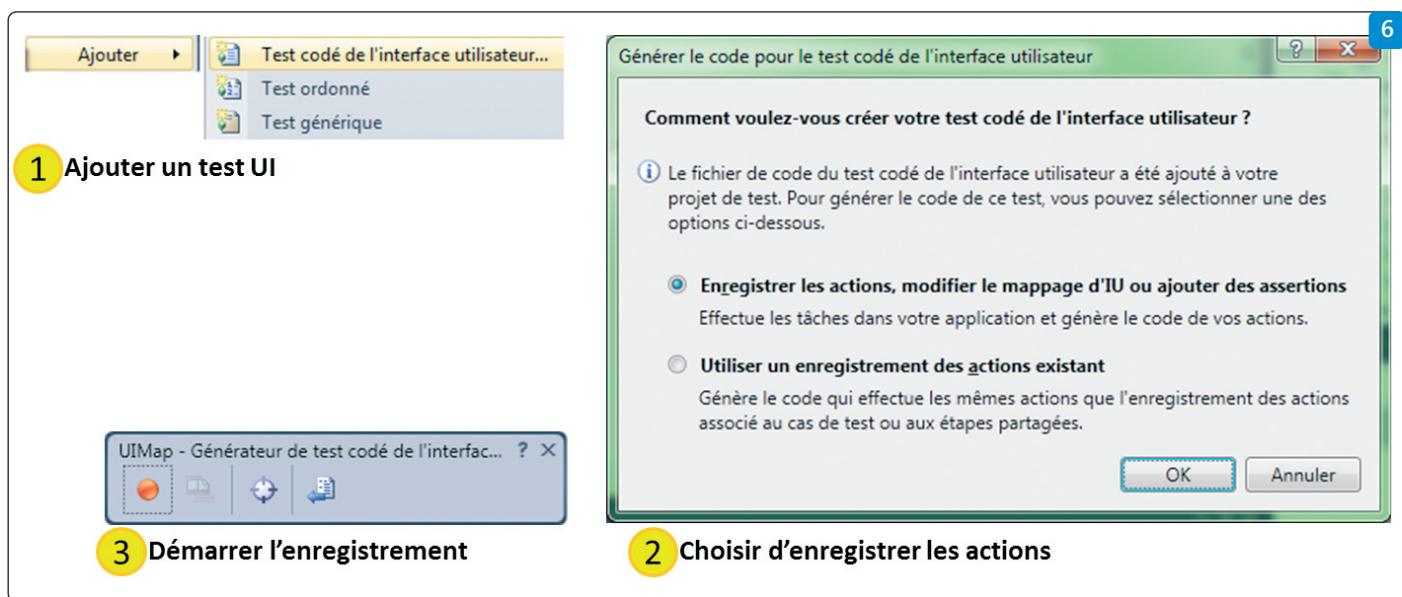
Les tests unitaires



Créer des tests unitaires

```
[TestMethod()]
public void AddTest()
{
    Calculatrice target = new Calculatrice();
    int a = 3;
    int b = 2;
    int expected = 5;
    int actual;
    actual = target.Add(a, b);
    Assert.AreEqual(expected, actual);
}
```

Code de test unitaire



Créer un test codé d'Interface Utilisateur

...puis vient le test codé d'interface utilisateur

Le test des méthodes de services représente forcément une part importante de la qualité d'une application. Cependant, il ne permet qu'une validation technique, très proche du code.

Mise en place des tests codés d'interface utilisateur

Une nouveauté très intéressante, apparue avec Visual Studio 2010, consiste à tester notre application à partir de son interface utilisateur. Ce test se rapproche des tests fonctionnels, en simulant un utilisateur réel. Ces tests d'UI (User Interface) vont nous permettre d'exécuter automatiquement des actions préalablement apprises, d'effectuer des mesures sur l'interface graphique, afin de valider ou d'invalider la conformité du logiciel. Un assistant permet d'effectuer très simplement ces différentes opérations (**Fig. 6**).

Au cours de l'enregistrement, et du paramétrage des points de mesure, du code source est généré dans le projet de test.

```
[CodedUITest]
public class CodedUIFromHomePage
{
```

```
public CodedUIFromHomePage()
{
}

[TestMethod]
public void CodedUIValidateAboutPath()
{
    this.UIMap.StartApplication();
    this.UIMap.ControlHomePage();
    this.UIMap.NavigateAbout();
    this.UIMap.ValidateAboutContent();
    this.UIMap.CloseApplication();
}
// ...
}
```

L'exécution de ce test va automatiquement démarrer l'application, et effectuer chacune des étapes définies lors de l'enregistrement, qu'il s'agisse d'actions, ou de contrôles.

Paramétrage

Mieux, à l'aide de ces tests, il devient possible d'entrer automatiquement des valeurs dans des champs de formulaires. Ces différents jeux de données d'entrées seront stockés dans une base de données, ou dans des fichiers, ce qui permet d'exécuter plusieurs fois un même scénario, avec des valeurs différentes.

La mise en place de ce mécanisme de Binding avec des données externes ne demande que quelques modifications dans le code généré, et le paramétrage de la connexion à la source de données externe.

Performance et charge : la cerise sur le gâteau !

Les tests unitaires et les tests d'interfaces utilisateurs permettent de valider la qualité fonctionnelle de l'application. La conformité de l'application ne se limite néanmoins pas à sa qualité fonctionnelle : il faut s'assurer que l'utilisation restera opérationnelle, dans des conditions réelles d'utilisation, avec un nombre important d'utilisateurs, sur une longue durée d'utilisation. Pour ce faire, il existe d'autres types de tests :

- Les tests de performance Web
- Les tests de charge

Les tests de performance Web

Les tests de performance Web ont ceci en commun avec les tests d'interface utilisateurs qu'ils permettent d'exécuter des scénarios d'utilisation, en effectuant des contrôles durant l'exécution. La différence principale est que dans le cas des tests de performance Web, seuls les échanges HTTP sont analysés, et non leur rendu à l'écran. C'est l'analyse des trames HTTP qui va permettre de contrôler la conformité des retours serveur de l'application (**Fig. 7**). L'un des avantages de cette méthode est sa capacité à être intégrée dans des tests de charge.

Les tests de charge

Le test de charge permet de valider la capacité de montée en charge de l'application, en exécutant les différents tests unitaires ou de perfor-

Résultats des tests			
			5
<input checked="" type="checkbox"/> Série de tests terminé	Résultats : 1/1 réussi(s) ; Éléments vérifiés : 0	Exécuter	Déboguer
Résultat	Nom du test	Projet	Message d'erreur
<input type="checkbox"/> Réussite	AddTest	BusinessLayerTest	

Résultat de test unitaire

1 Enregistrer la séquence

MON APPLICATION ASP.NET

À PROPOS DE

Placez le contenu ici

Requête

- ✓ <http://localhost:2343/>
- ✓ <http://localhost:2343/About.aspx>

Navigateur Web Requête Réponse Contexte Détails

En-têtes :
HTTP/1.1 200 OK

Corps : [Afficher dans l'éditeur HTML](#)

<p>
Placez le contenu ici.

2 Mettre en place des règles de contrôle

3 Exécution du test par analyse HTTP

Mettre en place d'un test de performance Web

mance, à l'aide d'un grand nombre d'utilisateurs virtuels simultanés. Il est possible de paramétriser le comportement des utilisateurs : temps de réflexion, alternance de scénarios, variation du nombre... La durée du test de charge est paramétrée, avec la possibilité de prévoir des temps de chauffe, et un temps de montée en charge. Durant ces tests de charges, différents compteurs tels que mémoire disponible, taux d'utilisation de CPU, nombre d'erreurs, etc... sont échantillonnés, et stockés dans une base de données de résultats.

L'analyse de ces résultats se fait depuis l'interface Visual Studio, qui présente différents rapports graphiques ou des tableaux permettant d'apprécier les résultats des tests. L'analyse peut se faire à postériori, les données étant stockées dans une base de données (Fig. 8).

Comment y aller ?

Méthodologie

La mise en œuvre de ces différents types de tests est un moyen de maîtriser la qualité de ses applications. Cependant, ces outils seuls ne suffiront pas : la mise en place d'une méthodologie adaptée est nécessaire.

Cette méthodologie doit permettre :

- De concentrer l'effort de test sur les parties les plus critiques
- De systématiser le test pour chaque livraison, voire pour chaque archivage de code source

1 Modéliser le comportement

2 Paramétriser les tests

3 Analyser les résultats

Statistique clé : 5 premières pages les plus lentes

URL (Lien vers Plus de détails)	95% du temps de réponse de la page (s)
http://localhost:2343/	0,33
http://localhost:2343/Default.aspx	0,0040
http://localhost:2343/About.aspx	0,0040

Statistique clé : 5 premiers tests les plus lents

Nom	95% de la durée du test (s)
PathHomeToAbout	22,9
AddTest	0,000031

Indicateurs clés

Mettre en place un test de charges

afin de détecter au plus tôt les anomalies, et y remédier.

La mise en place de l'automatisation des tests devient rapidement nécessaire afin d'offrir les garanties proposées par les outils. Par l'utilisation conjointe de Visual Studio, et de TFS, cette automatisation des tests est facilement accessible. Vous rêviez d'un moyen de tester automatiquement vos applications toutes les nuits, que ce soit sur l'IHM, les traitements ou les performances et la charge, de façon à disposer

chaque matin d'une liste d'anomalies à corriger ? La plateforme Visual Studio 2010 et Team Foundation Server 2010 rend ce rêve accessible, avec un coût de mise en place raisonnable.

Sylvain Gross

BU Manager Western Europe chez Neos-SDI

Responsable Technique Offre ALM/PLM

<http://www.neos-sdi.com>

sylvain.gross@neos-sdi.com

@SylvainGrossNeo

L'automatisation de tests sous Visual Studio

Si l'on compare les cycles de vie des projets informatiques de nos jours à ceux d'il y a une ou deux décennies, on constate des délais de livraison de plus en plus rapprochés, des utilisateurs finaux de plus en plus exigeants en termes de qualité, une démultiplication des configurations soft qui doivent être maintenant compatibles.

Les systèmes développés sont de plus en plus complexes et la concurrence ainsi que le « time to market » représentent un danger de plus en plus pressant.

Dans le même temps les outils de test permettant d'automatiser des processus métier ont été contraints d'évoluer pour correspondre aux besoins du marché.

Ils doivent être orientés objets, proposer une interface de programmation puissante et assistée. Dans ces conditions les problèmes de reconnaissance d'objet à l'enregistrement comme à l'exécution devront être réduits au minimum. La maintenance devra, elle, être réduite à son plus strict minimum.

En suivant les mouvements méthodologiques telles que Scrum ou toute autre méthode, les outils devront donc évoluer pour s'intégrer dans la démarche générale du cycle de développement des applications. Nous assistons à une recrudescence de nouveaux outils qui devront nous assister tant au niveau des tests unitaires qu'au niveau des tests fonctionnels utilisateurs, ou bien de performance.

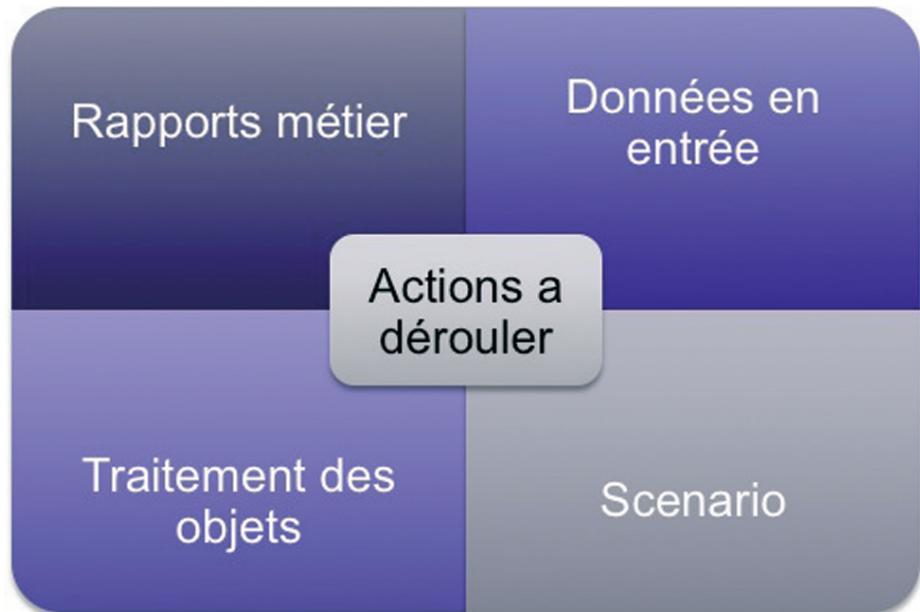
L'automatisation, un challenge à relever en équipe

Nous avons pu constater, en intervenant sur de nombreux projets d'automatisation, que certaines erreurs stratégiques impactent irrémédiablement le succès de la méthode : l'isolement des ressources dédiées à l'automatisation.

Sans une participation active de tous les acteurs dans un projet de développement logiciel, le processus d'automatisation risque d'être long et de ne cibler qu'une partie des points importants à vérifier.

Le partage idéal des tâches permettrait d'utiliser au mieux les compétences des acteurs de la conception logicielle :

- Le développeur pourrait valider les techniques de reconnaissance des objets, les environnements cibles des exécutions, et pourrait donc



ner le support nécessaire pour développer des scripts correspondant aux contenus des versions de builds et environnements cibles.

- L'expert métier pourrait intervenir dans le choix des périmètres fonctionnels à tester, il serait le référent idéal pour concevoir l'ensemble des scénarios ainsi que la matrice des jeux de données à utiliser
- L'ingénieur test et recette pourrait lancer les campagnes de tests automatiques, debugger le niveau le plus simple des échecs d'exécution et analyser les résultats obtenus
- L'expert automatisation aurait pour rôle de mettre en place un framework d'automatisation qui permettrait d'agencer les cas de test, de maîtriser les exécutions, de scripter les processus métiers de manière modulaire et de les mettre à la disposition des experts métiers qui organiseraient les scénarios.

De cette manière la montée en compétence de chacun restera concentrée sur les domaines maîtrisés et ainsi la couverture des tests automatiques ne sera pas remise en question. Un réel gain de temps pourra être alors constaté lorsque chaque acteur aura participé à son élaboration.

L'Automatisation par couche

Pour permettre à chacun de tirer un profit maximal de l'automatisation il est indispensable d'en dissocier ses composantes. (voir diagramme ci-dessus).

De cette manière chacun pourra apporter sa pierre à l'édifice automatique sans interférer sur les autres composantes.

Dans le cas d'un changement d'environnement (donc de jeux de données à modifier), les données étant stockées dans un objet indépendant du script, et le mode opératoire ayant changé, alors la couche de gestion des processus métiers sera maintenue.

Une force de frappe adaptée au besoin

Coded UI est une réponse appropriée à la mouvance du monde automatique.

Profondément ancré dans le socle du développement logiciel, Coded UI s'intègre nativement aux cycles de livraison des versions logicielles produites par Visual Studio. Ainsi grâce à l'interface de configuration de la build nous pourrons facilement lancer les package de tests automatiques correspondant aux tests d'environne-

ment, au bon fonctionnement minimum ou la non Régression au moment où la version est « construite »

Les campagnes de test ciblées sont aussi beaucoup plus simples à définir grâce à TFS. En effet TFS permet d'accéder à toutes les données cruciales du projet, il est trivial de définir les cas de test impactés par un changement dans le code source. Ce niveau de précision est bien plus utile qu'un simple lien à une exigence métier. Cette même fonctionnalité demande dans des outils comparables un développement conséquent dont la stabilité n'est pas toujours à toute épreuve

Fluidité dans l'automatisation

Imaginez de plus que le même moteur permettant aux développeurs de créer des objets (IHM ou autre ADO etc.) soit à votre disposition pour reconnaître puis interagir avec vos applications, et ce en mode Web ou Windows front end.

Une automatisation par couche Native

Durant des années nous avons recherché et développé des solutions qui nous permettaient une dissociation entre les différentes composantes business, technique, Data ou scénarios, avec la solution d'automatisation de Visual studio et de par son architecture anticipée vous pourrez héberger chacune des couches dans des fichiers différents

Un moteur de rapport modulable et puissant

Grâce aux Report services de TFS nous disposons de toutes les informations nécessaires à l'élaboration de rapports explicites, orientés métier ou technique, et portables. Nous pourrons donc consolider les indices de qualité pour faire naître un tableau de bord facilement optimisé, et si besoin, lié à un accès SharePoint.

D'un point de vue technique

La participation de tous les acteurs d'une cellule de qualification logicielle outillée peut être une pierre à l'édifice d'une campagne de tests automatisée. En effet, un testeur manuel a des connaissances métier précieuses pour l'automatique et il convient de les mettre à profit dans l'automatisation des tests. Ce testeur manuel a donc à sa disposition un client lourd lui servant de référentiel de tests et d'assistant à leur bon déroulement : Visual Studio Test Manager. Dès lors qu'il est prêt à exécuter manuellement ses tests, VSTS lui propose d'enregistrer toutes ses actions à la manière des macros Word ou Excel (**Fig. 1**).

Dès lors, son cas de test est enregistré et il peut déjà se vanter : « Je fais de l'automatisation de tests ». Il pourra s'en servir pour rejouer immédiatement son cas de test avec la ligne suivante du jeu de données du test et même lors des prochaines exécutions de celui-ci dans les jours, semaines ou mois à venir en cliquant l'icône Play lors de la prochaine campagne.

Ce n'est pas là leur seul point commun avec ces macros Excel qui ont tant fait pour nous faciliter le travail redondant. En effet, ces actions enregistrées peuvent être récupérées sur l'environnement de développement de Visual Studio. Le code ainsi généré est parfaitement compréhensible, retouchable, et entièrement personnalisable.

Cette génération de code fait apparaître plusieurs nouveaux fichiers dans notre projet .NET : • CodedUITest1.cs : contient la classe représentant notre Coded UI. Assure la couche métier du test dans laquelle on va combiner les différents test steps pour former un test case. Elle

assure également la récupération des lignes des jeux de données de tests depuis Test Manager. A noter que la source de données (Test Manager par défaut) est modifiable vers une base SQL Server, un fichier XML, Excel, Access, txt, etc.

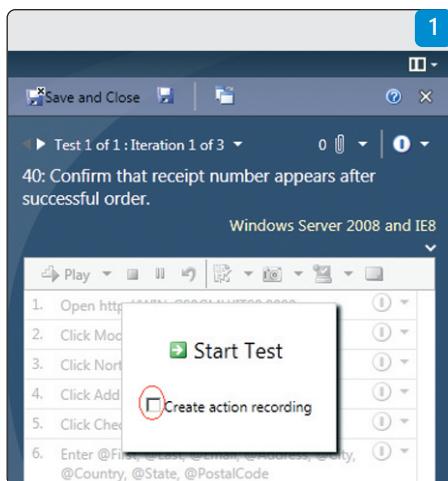
- UIMap.uitest : contient le modèle XML du mapping des objets manipulés par le test.
- UIMap.Designer.cs : contient la représentation de code du fichier XML contenu dans le fichier UIMap.uitest, soit les propriétés utilisées pour identifier les objets de test et les méthodes capturées précédemment. Elles portent le nom des pas de tests choisis dans Test Manager. En effet, chaque test step génère sa propre méthode unitaire afin d'optimiser leur réutilisation par la suite. Ces méthodes sont également commentées.
- UIMap.cs : contient les méthodes de manipulation des objets de test personnalisées. Plus elles sont unitaires, plus elles sont réutilisables.

Jusqu'à ce point, seules des manipulations automatiques d'IHM composent notre test automatique. Faut-il encore vérifier le bon comportement de notre application et donc ajouter des points de vérification : l'**assertion**.

En faisant apparaître le codedUITestBuilder, on va pouvoir enregistrer de nouvelles manipulations à automatiser, ajouter de nouveaux objets de l'application à notre projet afin de pouvoir les manipuler ou vérifier les valeurs de leurs propriétés : ce sont les assertions (**Fig. 2**).

Par un glisser-déplacer depuis l'icône cible du codedUITestBuilder sur un objet composant de l'application à tester, on va le sélectionner. Alors, lui et ses propriétés apparaissent. On peut ainsi sélectionner un couple [propriété – valeur] afin d'ajouter la méthode qui en vérifie le contenu (**Fig. 3**).

En cliquant sur Add Assertion, on va choisir le modèle de comparaison de la valeur : AreEqual, not equal, starts with, contains etc (**Fig. 4**).



This screenshot shows the 'Coded UI Test Builder - Add Assertions: UIClearButton' dialog box. On the left, a tree view shows nodes like 'UIAddtwonumbersWindow', 'UITextBoxValue1Edit', 'UITextBoxValue2Edit', 'UIAddButton', and 'UIClearButton'. The 'UIClearButton' node is selected. On the right, a table titled 'Add Assertion' lists properties and their values. The first row is highlighted with yellow background: 'AutomationId' is set to 'buttonClear', 'ControlType' is 'Button', and 'TechnologyName' is 'UIA'. Other rows include 'HelpText', 'AcceleratorKey', 'AccessKey', 'LabeledBy', 'Shortcut', 'DisplayText' (set to 'Clear'), and 'Generic'.

Pour ajouter l'assertion à notre test automatisé, on va générer le code de cette assertion (Dernier bouton du CodedUITestBuilder). Après avoir choisi le nom de la méthode de l'assertion, elle sera ajoutée à la classe UIMap et l'appel de cette méthode sera lui ajouté à la méthode globale dans le fichier CodedUITest1.cs. Néanmoins, vous pourrez déplacer l'appel de cette méthode à votre guise (**Fig. 5**).

Bien que chaque test case codé dans le codedUITest soit directement exécutable, on peut les combiner dans un Ordered Test pour créer et exécuter une campagne complète à la demande, cet ordered test devenant lui-même un test case (**Fig. 6**).

Sans sortir de notre sujet, on peut ajouter que l'ordered test peut contenir non seulement des tests case automatisés mais également des tests de performance.

TFS inclut un processus complet de build de vos applications. Sans en expliciter les détails ici, on peut déjà dire que construire une build de son application ne se résume plus à une simple compilation d'un projet Visual Studio. On a aujourd'hui besoin, en plus de compiler ses pages de code, de créer un numéro de release, de la différencier en détail de la précédente, d'effectuer toutes sortes d'actions au niveau de la base de données, de créer une nouvelle instance de campagne de tests et beaucoup de choses encore. TFS va nous aider en fournissant un outil de configuration de build qui nous permet, une fois mis en place, de builder l'application en un seul click mais au-delà de cela, de la planifier.

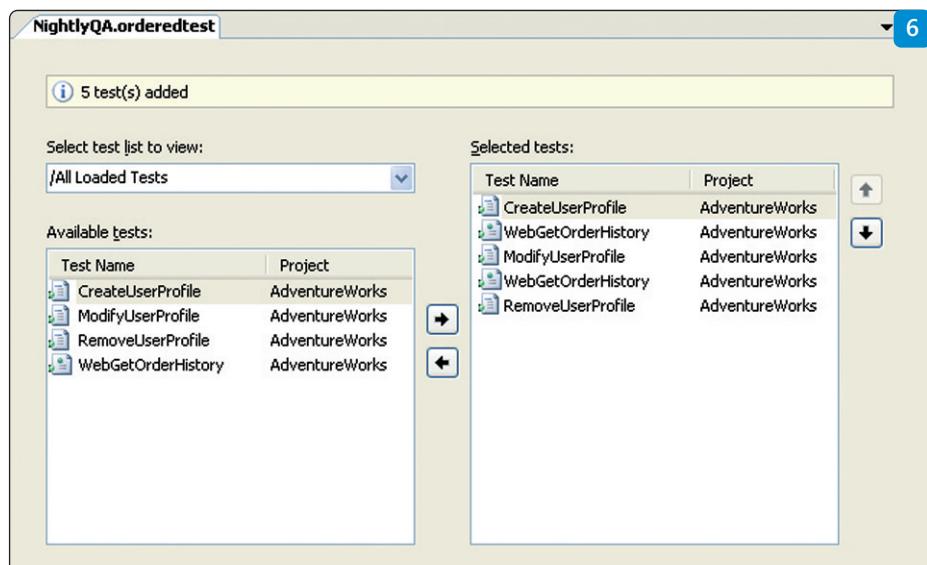
Mais pour revenir à notre sujet, ce processus de build automatique peut imbriquer nos campagnes de tests automatiques sur la build faite et déployée. Le LAB CENTER fourni en complément nous laisse gérer nos environnements pour choisir les emplacements de déploiement et même créer à la volée des machines virtuelles, y déployer nos applications buildées et jouer nos campagnes de tests automatiques et tests de performance (**Fig. 7**).

5

```
public void checkSearchButtonText()
{
    Variable Declarations

    // Verify that 'Search' button's property 'DisplayText' equals 'Search'
    Assert.AreEqual(this.checkSearchButtonTextExpectedValues.UISearchBarDisplayText, UISearchBar.DisplayText);
}

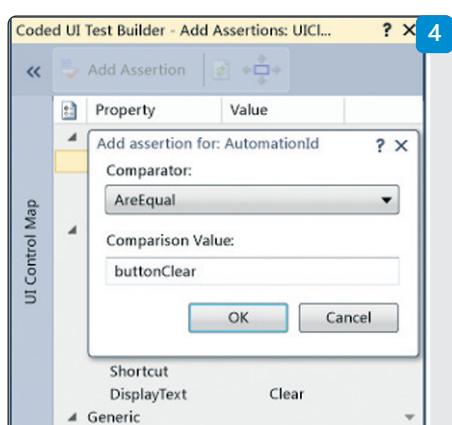
public void CodedUITestMethod1()
{
    // To generate code for this test, select "Generate Co
    // For more information on generated code, see http://
    this.UIMap.SeconnecteràURL();
    this.UIMap.Entrevilledanslechampville();
    this.UIMap.checkSearchButtonText(); ←
    this.UIMap.CliquersurSearch();
    this.UIMap.Fermerlenavigateur();
}
```



7

The screenshot shows the 'Build process template' configuration window. It has sections for 'Build process template:' (set to 'Default Template') and 'Build process parameters'. The 'Build process parameters' section contains several items:

- 1. Required**: Items to Build: Build \$/Demo/Main/Source/MySolution.sln with default platform
- 2. Basic**: Automated Tests: Run tests in assemblies matching ***test*.dll using settings from build definition
- 3. Advanced**: Build Number Format: \$(BuildDefinitionName)_\$(Date:yyyyMMdd)\$(Rev:r)
- Clean Workspace: All
- Logging Verbosity: Normal
- Perform Code Analysis: AsConfigured
- Source And Symbol Server Settings: Index Sources



Bref, Microsoft fournit ici un outil puissant, simple à prendre en main, économique de son temps, et complet. Imaginez : Partir du bureau le soir en ayant terminé ce qui était à faire dans votre journée, et vous le savez grâce aux indicateurs fournis.

Votre infrastructure se met en marche pendant que vous dormez : Votre application se compile, toutes les opérations de build se font, le produit se déploie sur des VM créées au bon moment, les tests automatisés se déroulent, jusqu'aux tests de performance.

Vous arrivez le lendemain matin et trouvez les résultats complets de qualimétrie de vos projets, que de temps gagné !!!

En conclusion

Coded UI est donc un outil d'avant-garde qui peut s'intégrer nativement au cycle de développement de vos applications. N'hésitez pas à nous contacter.

Laurent Abid

CEO, T.T.C Testing Training & Consulting Ltd
Laurent@TTC-testing.com

Didier Davar

Consultant Senior Démarche outillée, T.T.C Testing Training & Consulting Ltd
didier.d@ttc-testing.com

Planifier et organiser les tests avec Microsoft

Microsoft intègre une série d'outils de tests dans sa solution Visual Studio 2010 (Application Lifecycle Management). Un processus complet de suivi des plans de test y est proposé avec des solutions de collaboration entre développeurs et testeurs.

Les objectifs sont de :

- simplifier la compréhension du besoin et du test associé,
- améliorer le dialogue et la visibilité entre le demandeur et le développeur tout au long du cycle de développement,
- faciliter la remontée d'informations sur les anomalies aux développeurs, par exemple, avec l'enregistrement vidéo des exécutions des tests.

Microsoft vise également la démocratisation des tests plus accessibles et économiques. C'est une réponse aux besoins croissants des projets qui recherchent plus de qualité, de performance et une mise en production plus rapide.

La capacité de supporter des processus agiles est un facteur décisif d'adoption des outils de tests de Visual Studio. Près de 80% des entreprises affirment qu'avec son approche Microsoft Agile, Visual Studio est une étape stratégique vers une organisation Agile. Toutefois, cela ne signifie pas que l'on peut se passer de quantifier les bénéfices de l'assurance qualité et des outils de test. Il est vivement recommandé de mettre rapidement en place des indicateurs de qualité et d'amélioration du cycle de développement.

Par où commencer ? Les processus de tests et de transformation de la qualité s'articulent toujours autour de mêmes fondamentaux : la gestion des exigences, du référentiel des tests, des

campagnes d'exécutions, des anomalies et la décision du « Go/no Go » - voir **schéma 1**. Le pilotage et le reporting sont assurés tout au long du cycle des tests selon les indicateurs établis.

Sur la base des exigences de tests, les responsables des tests élaborent un référentiel afin de gérer l'effort pour chaque itération programmée. L'objectif est ainsi d'avoir une vue complète sur tous les éléments (les cas de tests, les configurations). Plusieurs référentiels de tests peuvent être générés pour un projet donné, toutefois, afin de faciliter la capitalisation, il est préférable de n'en prévoir qu'un par projet.

Chaque plan de test correspond à une mise en situation différente. Il permet de définir les conditions spécifiques des tests tels que le nom, le contenu du plan de test, son responsable ou son état d'avancement. Notons qu'avec la fonctionnalité Build de TFS 2010, l'équipe informatique peut optimiser ou automatiser certaines étapes de l'ALM en vue d'une méthodologie agile en accélérant la cadence de montée en production de l'application tout en maintenant un niveau de qualité élevé et en garantissant le contrôle des coûts (**schéma 2**).

A partir d'un plan de test, il est possible d'exécuter des tests manuels et d'automatiser les tests en utilisant Microsoft Test Manager avec Visual Studio Test Professional 2010 ou Visual Studio 2010 Ultimate. Quel que soit le test lancé, l'utilisateur peut sauvegarder les résultats des tests directement dans le projet d'équipe sur Team Foundation Server. Cela permet de voir simultanément où en sont les



tests automatisés et les tests manuels réalisés. Avec Visual Studio 2010 Ultimate, le pilotage de la correction des bugs se réalise soit avec une vision des bugs priorisés soit avec la vision de couverture des exigences de tests, puisque les anomalies et les tests sont liés aux exigences. Cela permet de concentrer son effort de correction sur ce qui est important. Le traitement des bugs est aussi aisément confié à un autre utilisateur, car Visual Studio 2010 donne accès à plus d'informations partagées. Une fois le bug traité, il peut être assigné à un testeur pour vérifier que le problème a été réglé sans incidences secondaires sur l'application.

Les utilisateurs de Visual Studio 2010 Ultimate ou Premium peuvent également automatiser les tests de l'interface utilisateur (coded UI tests). L'automatisation des tests, qui est un projet dans le projet, permet de rejouer, à volonté et plus rapidement que les tests manuels, un ensemble de tests prioritaires et importants à moindre coût. Dans la mesure où la maintenance de ces tests peut être lourde, il faut définir le ROI et le risque du référentiel de tests pour choisir les éléments à automatiser. C'est la prochaine étape pour transformer la qualité, l'efficacité et la performance du développement d'applications de l'entreprise tout en réduisant les coûts.

Salam Elias,

Responsable de l'Alliance Microsoft et Directeur de projets chez Cognizant (<http://www.cognizant.com>). Consulter son blog : <http://salam.hd.free.fr/BlogEngine/>

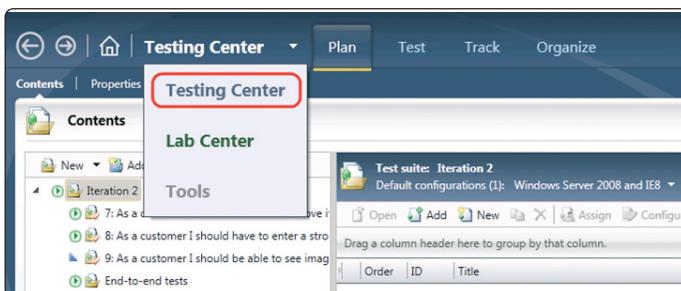


Schéma 2 – Définition des Plans de test à partir de Test Center de Visual Studio 2010 / Microsoft.

Témoignage

Améliorer la qualité logicielle avec Team Foundation Server 2010

La société Log'in Space est un éditeur de logiciels dans le domaine de l'immobilier (gérance, syndic, location saisonnière, etc.). L'éditeur fait partie des leaders dans le domaine en France.

Avec une gamme existante de logiciels développés dans des technologies comme Cobol et 4D, la société a fait le choix stratégique de redévelopper la gamme de logiciels avec les technologies .net, en utilisant entre autres, des technologies comme WPF pour proposer une ergonomie originale et efficace, et WCF pour l'exposition de la logique métier et d'accès aux données sous forme de services. Afin d'encadrer le développement de l'applicatif, le choix de Team Foundation Server a été fait afin de profiter de ses différentes fonctionnalités et de son intégration au sein de Visual Studio. Nous vous proposons au travers de cet article de vous présenter deux éléments mis en oeuvre pour améliorer la qualité de nos produits grâce à TFS 2010.

Les revues de code technique

En plus des sessions de pair programming qui permettent de partager des points de vue sur la manière d'écrire du code, mettre en place des revues de code permet de s'assurer que le code « checkiné » est correct, qu'il ne comporte pas de potentiels bugs, qu'il respecte les normes de développement, etc. Visual Studio 2010 ne propose pas de fonctionnalité permettant de mettre en place de manière naturelle ce type de revue de code (la prochaine version corrigera ce défaut en proposant un outil de revue de code

intégré). Il est cependant possible de mettre en place efficacement ces revues en profitant de la traçabilité offerte entre les Work Items (tâches et bugs) et les changesets. Mettre en place ce type de revue de code est simple. Assurez-vous que l'ensemble des développeurs associent leurs changesets avec un work item lors de leur checkin, en mettant en place une Checkin-Policy ou en indiquant cette obligation dans les normes de développement.

Afin de savoir quels sont les éléments à réviser, créer ensuite une requête de Work Item qui permet de lister l'ensemble des WorkItems qui sont en état Resolved. Pour chaque tâche/bug terminé/résolu par un développeur (Fig. 1).

Il vous est ensuite possible d'effectuer des revues de code quotidiennes en analysant le différentiel de code implémenté pour résoudre les bugs concernés (Fig. 2).

La mise en place de ce type de revue de code a cependant un inconvénient majeur : les revues de code ne sont effectuées que sur du code qui est déjà présent en source control.

La mise en place de tests UI

La mise en place de tests est fondamentale pour parvenir à améliorer la qualité d'un projet, réduire le risque de régressions et travailler sereinement. La mise en place à posteriori de tests unitaires et de tests d'intégration peut être assez complexe et coûteuse en temps. Une première étape assez simple à mettre en œuvre est d'automatiser les tests manuels. Pour cela

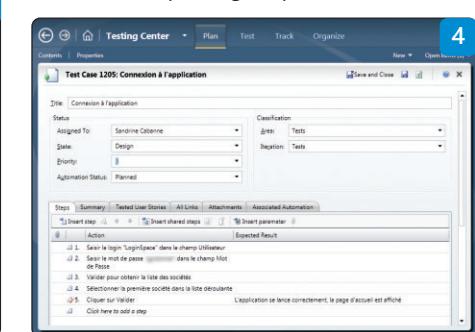
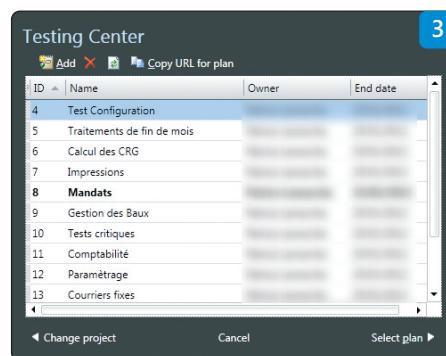
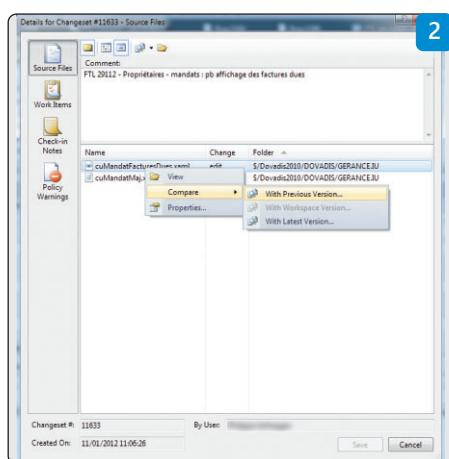
Microsoft propose un outil nommé Microsoft Test Manager. Cet outil permet de gérer des plans de tests qui contiennent des cas de tests destinés à vérifier le bon fonctionnement d'une fonctionnalité (Fig. 3).

L'outil permet d'enregistrer des séquences d'actions effectuées sur l'interface utilisateur afin d'être capable de les reproduire très rapidement sans avoir à effectuer de manipulations manuelles. Ceci implique que le résultat final doit toujours être vérifié manuellement par le testeur en fonction des indications fournies par l'outil (Fig. 4).

Afin d'arriver à effectuer automatiquement les vérifications (les asserts) il est nécessaire d'utiliser Visual Studio afin de générer un Coded UI Test à partir du cas de test auparavant enregistré. Celui-ci va générer le code nécessaire à l'exécution du test, et va permettre d'écrire (ou de générer) le code nécessaire à la définition de l'Assert. A noter que la définition de tests UI ne se fait pas aussi idéalement qu'on le souhaiterait. Il est fort probable que les développeurs aient à revenir sur le code XAML de l'application pour ajouter différentes informations telles que la propriété Name si elle est manquante par exemple. De plus, si la création d'assertions peut être très simple dans certains cas, il n'est pas rare qu'il soit nécessaire d'écrire différents outils pour simplifier l'écriture de celles-ci. Je peux vous recommander la suite d'outils de Jonathan Antoine dans ce domaine : <http://bit.ly/xRBNL9>

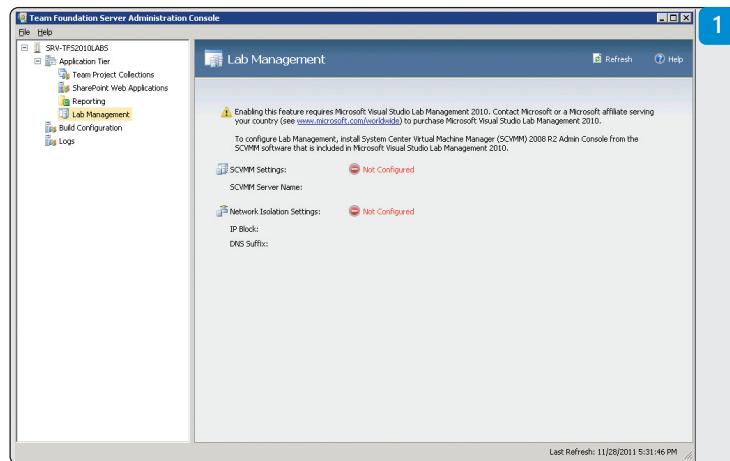
Patrice Lamarche

Leader Technique (Log'in Space)



Intégration continue et tests avec TFS Lab Management

« *Sur ma machine, ça fonctionne pourtant* ». « *Impossible de reproduire ce bug* ». N'avez-vous pas déjà entendu ces phrases dans vos projets de développement ? Avec leur complexité croissante, les outils d'ALM apportent de nouvelles solutions pour assurer une forte qualité logicielle traçable et automatisée. Team Foundation Server propose la fonctionnalité Lab Management afin de faciliter la reproduction des bugs, l'automatisation des tests et des déploiements ainsi que le suivi de la qualité. Dans cet article, nous verrons comment le mettre en place puis son utilisation et ce qu'il peut nous apporter.



Infrastructure et déploiement

La réticence à déployer Lab Management découle essentiellement de la complexité de son infrastructure. Pour gérer les environnements virtuels des projets d'équipe, TFS requiert un serveur System Center Virtual Machine Manager 2008 R2.

On peut résumer très synthétiquement SCVMM comme un centre de contrôle pour hyperviseurs. Avec la mouture actuelle, vous pouvez gérer de manière agrégée le déploiement et la maintenance de machines virtuelles réparties sur plusieurs hôtes aussi bien Hyper-V que VMware, par une interface graphique ou avec du PowerShell. Cet article n'a pas pour but de détailler le déploiement d'un SCVMM mais il faut savoir que c'est en lui-même un projet à part entière.

Quand on ouvre les propriétés de Lab Management dans la console d'administration de TFS, on constate qu'on ne peut rien faire tant que la Console d'administration de SCVMM n'a pas été installée. (**Fig.1**)

Il faut donc l'installer depuis le disque/iso d'installation de SCVMM et s'assurer que le compte de service de TFS dispose du rôle administrateur dans SCVMM.

Une fois ceci fait, on peut commencer à configurer les options de base de Lab Management, à savoir référencer le serveur SCVMM par son nom complet puis les paramètres d'isolation réseau. L'isolation réseau permet à TFS d'instan-

cier des VM sans polluer le réseau existant mais aussi d'avoir plusieurs environnements d'un même projet en même temps (pour plus de détails : <http://msdn.microsoft.com/en-us/library/ee518924.aspx#NetworkIsolationChecklist>).

Il y a plusieurs choix de topologies possibles mais qui ont des complexités différentes en fonction de l'infrastructure existante. Après avoir configuré les bases au niveau global TFS, il faut maintenant configurer plus spécifiquement au niveau de chaque collection de projets (**Fig.2**).

Ces options sont le cœur du Lab Management puisque c'est ici que l'on va configurer les bibliothèques de VM, les groupes d'hôtes (hyperviseurs) sur lesquels elles seront déployées et le compte de service de workflow qui sera utilisé pour les builds et les tests. L'intérêt des groupes d'hôtes est que l'on n'a pas à se soucier de savoir où sont placées les VM : SCVMM le fait pour nous en calculant, d'après les préférences établies par l'administrateur, l'hyperviseur le mieux noté pour la tâche en question.

On profite là de la première itération des outils MS pour le Cloud privé appliquée au développement. L'intérêt que ces paramètres soient au niveau collection de sites réside bien entendu dans le fait de pouvoir avoir différentes configurations. On peut ainsi avoir une collection de projets prioritaires qui bénéficiera d'hyperviseurs plus véloces et de VM avec plus de res-

sources par rapport aux projets moins importants.

N.B. : actuellement, TFS 2010 ne supporte pas les hôtes en cluster pour le provisionnement automatique des VM MAIS il fonctionne parfaitement pour des VM provisionnées et configurées manuellement ou avec VM prep tool .

Afin de pouvoir dispatcher les tests automatisés, on doit installer un contrôleur de tests, idéalement sur un serveur dédié, provenant du même disque que les agents.

L'installation et la configuration est très simple et la capture (**Fig.3**) parle d'elle-même.

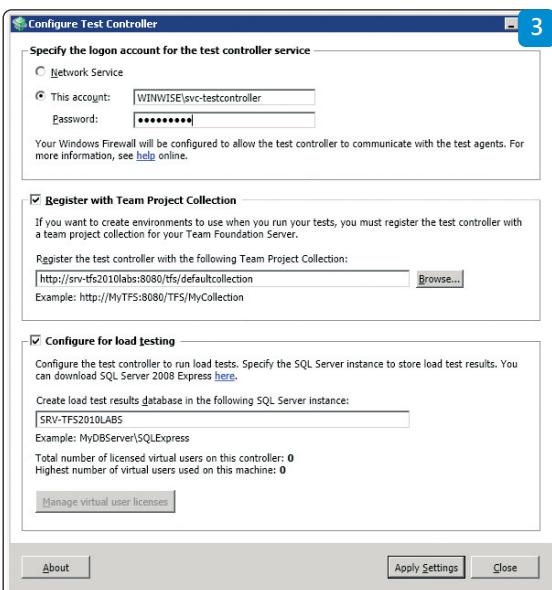
Enfin, pour avoir une configuration supportée et bénéficier des dernières fonctions de TFS, il faut installer un hotfix disponible à l'adresse suivante : <http://support.microsoft.com/kb/983578>.

Utilisation de Test Manager

Maintenant que la plateforme est mise en place, on ne va pas se priver d'utiliser l'ensemble des fonctionnalités offertes par le Lab Management !

L'outil au cœur de l'utilisation du Lab Management est le Test Manager. Si vous avez déjà l'habitude de définir des plans de test, vous avez déjà eu affaire à lui.

Difficile de choisir par quoi commencer, l'outil est vraiment complet et répond aux problématiques rencontrées le plus fréquemment au cours de la vie d'un logiciel. Nous allons aborder les plus courantes et voir comment le Lab



Management, couplé au reste de la plateforme TFS bien évidemment, y répond.

« Je dois tester le déploiement de mon application sur différents environnements cibles »

Sans le Lab Management, on est obligé de gérer soi-même les machines sur lesquelles on déploie l'application et cela devient vite fastidieux si on doit couvrir un nombre important de systèmes d'exploitation.

Avec, tout devient plus facile :

Commencez par créer à travers SCVMM autant de machines virtuelles que d'environnements différents sur lesquels vous souhaitez tester le déploiement de votre application. Installez également les agents de Build TFS 2010, tests et lab sur chaque machine. Faites un snapshot sur chaque environnement une fois que vous avez installé tous les prérequis nécessaires à votre logiciel.

N.B. : pour simplifier ces tâches, un outil existe : VM prep tool disponible ici :

<http://archive.microsoft.com/vslabmgmt>

Stockez ensuite ces VM dans la librairie de SCVMM. Vous aurez le choix entre les stocker telles quelles ou les enregistrer en tant que template. Vous choisissez la deuxième option si vous avez besoin d'instancier plusieurs fois une même VM pour vos tests.

Importez dans votre projet TFS les VM dont vous avez besoin à partir de SCVMM. Cela se passe dans le Test Manager. Notez qu'étant donné que les VM sont stockées dans SCVMM, une seule VM peut être réutilisée dans plusieurs Team Projects.

Créez un environnement (toujours via Test Manager) contenant toutes les VM sur lesquelles vous souhaitez déployer votre application.

Créez une build utilisant le template « LabDefaultTemplate.xaml ». Dans l'onglet Process vous aurez accès à un wizard de configuration vous permettant de sélectionner l'environnement précédemment créé.

Il ne nous manque plus qu'à indiquer quelles lignes de commande vous souhaitez exécuter sur chaque VM de votre environnement afin de déployer les différents composants de votre application.

Vous pouvez maintenant lancer votre build et observer les résultats pour contrôler le bon déploiement de votre application sur les différents environnements.

On vient de voir rapidement l'automatisation du déploiement d'une application sur différents environnements cibles. Cela peut paraître un peu long à mettre en place, c'est vrai à l'initialisation, mais vous gagnez :

- L'automatisation du déploiement de votre application, peu importe les modifications de code
- L'évolution simple de vos environnements cibles (il suffit de changer les VM de votre environnement)
- La possibilité de vous connecter sur les VM une fois l'application déployée pour effectuer des tests

Les possibilités en termes de paramétrage des environnements et des build de déploiement sont immenses. Le template de build fourni par défaut peut être modifié (comme tout template de build). Vous pouvez gérer finement les paramètres réseau des VM : fonctionnement dans un réseau isolé, interaction avec le réseau de l'entreprise). Tout a été prévu pour permettre la définition des environnements correspondant au contexte réel de votre application, aussi complexe soit-il. Cela est déjà intéressant mais on peut aller beaucoup plus loin.

« Je veux tester mon application sur différents environnements cibles »

Déployer son application automatiquement, c'est un bon début mais cela ne garantit pas qu'elle fonctionne. Allons un peu plus loin dans l'automatisation et rajoutons des tests à tout cela. La plus grosse partie du travail se passe dans l'outil Test Manager.

Si vous êtes déjà familiarisé avec la création de plans de test vous n'aurez aucune difficulté à définir l'ensemble des tests que vous souhaitez exécuter de manière automatisée.

Il vous faut en effet définir un plan de test composé de suites de tests composées elles-mêmes de cas de tests (automatisés ou non).

Une fois que vos plans de test sont définis, il ne vous reste plus qu'à les associer à la build qui effectue déjà le déploiement de l'application. En relançant le wizard de configuration évoqué précédemment et en suivant l'étape « Test » vous pourrez sélectionner les plans/suites de tests/cas de tests que vous souhaitez exécuter. Voilà, votre build qui déployait votre application exécute désormais les tests automatisés dans la foulée.

Les résultats sont consultables via le rapport de build ou dans le Test Manager en consultant les résultats des plans de test.

Les tests exécutables de cette manière sont :

- Les tests unitaires
- Les tests web
- Les tests codés d'UI

C'est bien me direz-vous mais comment fait-on pour les tests manuels ?

Rien de plus simple en utilisant les options fournies. On peut en effet indiquer à notre build qui déploie puis teste l'application de faire un snapshot de l'environnement après avoir déployé

QUALITÉ LOGICIELLE

l'application et avant d'avoir exécuté les tests. En passant par le Lab Center nous pouvons accéder à notre environnement virtualisé et décider de restaurer le snapshot créé par l'exécution de la build puis nous connecter à l'environnement. Nous nous trouvons donc sur notre environnement cible sur lequel la version de l'application que nous voulons tester est installée. Il ne reste plus qu'à lancer le Test Center depuis cette même machine, de parcourir le plan de test concerné, de sélectionner le cas de test manuel à dérouler et de cliquer sur « Run ». Nous sommes donc en train d'exécuter un test manuel dans les mêmes conditions que les tests automatisés qui ont été déclenchés par la build. Petite cerise sur le gâteau, si jamais vous détectez une anomalie lors du test manuel, vous pouvez prendre un snapshot de l'environnement. Un lien vers ce snapshot sera alors automatiquement ajouté lorsque vous créerez le bug associé au cas de test manuel.

Le partage de snapshot peut aussi se faire en exportant un fichier .lvr qui peut être envoyé par email à un autre collaborateur (**Fig.4**).

Ce fichier ne contient qu'une simple ligne de connexion (ex « CommandUrl=levr://ConnectTo FwdLink?CollectionURI=http://srv-tfs2010labs: 8080/tfs/DefaultCollection&ForwardLink=1 »). Cela permet à la personne en charge de la résolution du bug de constater ce qui s'est produit et de pouvoir analyser les logs système et applicatifs générés par votre test (**Fig.5**).

Comme cela, fini le souci du « chez moi ça marche » qui ne fait pas avancer la résolution du bug. Vous pouvez tester directement dans les mêmes conditions que la personne qui a détecté l'anomalie.

Conclusion

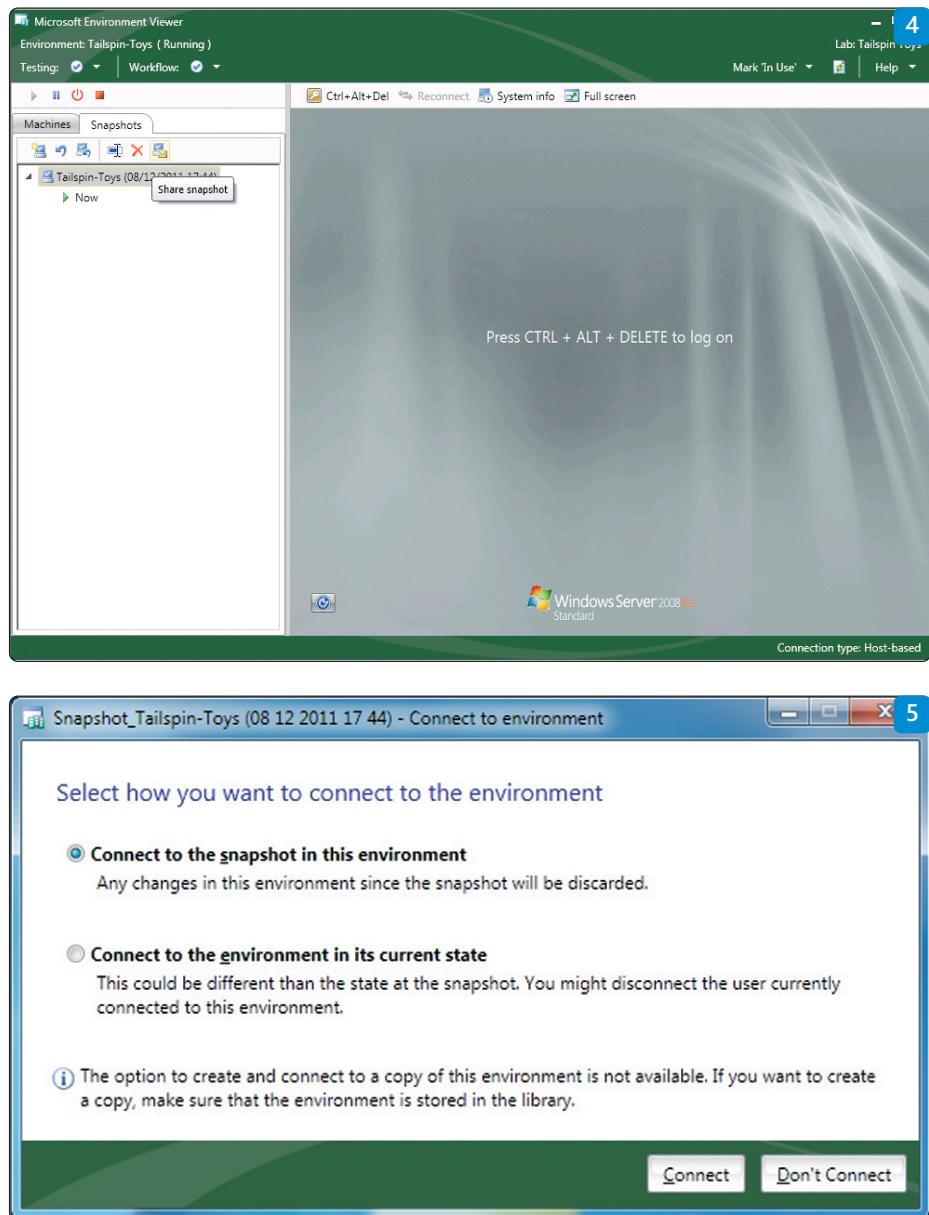
Nous avons vu dans ces quelques pages une présentation rapide de Lab Management dans TFS mais il faudrait un livre entier pour réellement être exhaustif sur le périmètre fonctionnel de l'outil.

Pour des projets conséquents et, avec de l'investissement en assurance qualité, le TCO arrive à rester modéré.

Qu'en est-il de l'avenir ?

Les gammes System Center 2012 et TFS 11 sont d'ores et déjà en préversions publiques et apporteront leurs lots d'améliorations.

Concernant la prochaine version Lab Management, Brian HARRY, le père de TFS, a déjà annoncé qu'il serait plus simple à mettre en place en ne nécessitant plus d'être intégré à un



SCVMM pour fonctionner (<http://blogs.msdn.com/b/bharry/archive/2011/10/31/lab-management-improvements-in-tfs-11.aspx>).

Les projets pourront ainsi être déployés et testés sur des machines physiques existantes qui auront été configurées en conséquence.

Bien entendu, pour des infrastructures plus conséquentes, l'outil pourra exploiter SCVMM 2012. Il n'y a, à l'heure où ces lignes sont écrites, aucune annonce sur les évolutions prévues sur ce point. On espère que les clusters d'hyperviseurs et les clouds privés pourront être intégrés avec TFS.

Comme pour tous les déploiements, vous devez vous demander s'il est préférable d'attendre avant de mettre en place les prochaines versions. La réponse dépend de la volumétrie à mettre en place. Une forte volumétrie (plusieurs

dizaines voire centaines de VM) impliquant un projet d'une durée supérieure ou égale à 10/12 mois pourra justifier d'attendre.

Liens utiles

Utilisation d'un lab virtuel pour le cycle de vie de votre application : <http://msdn.microsoft.com/fr-fr/library/dd997438.aspx>

Le blog d'équipe de Lab Management (en anglais) : http://blogs.msdn.com/b/lab_management/

Les vidéos d'apprentissage MSDN (en anglais) : <http://msdn.microsoft.com/en-us/vstudio/gg537361#lab>

Pierrick Catro-Brouillet

Expert SharePoint et Infrastructure, Winwise
<http://blogs.developpeur.org/spbrouillet>

Cédric Noureau

Consultant/Chef de projets, Winwise

Aperçu de la prochaine version

Collaboration, agilité, gestion du cycle de vie, tests, quatre piliers essentiels qui sont aujourd’hui les fondations de TFS et de Visual Studio. Microsoft travaille activement à la prochaine version de Visual Studio.

La philosophie demeure identique : couvrir toujours mieux le cycle de vie, améliorer le travail d'équipe, mettre des interactions toujours plus efficaces. Voyons ensemble quelques éléments de cette prochaine version. Pour cette future version, les équipes Microsoft ont concentré leur travail sur quatre éléments :

- collaboration : concentrer les flux entre les membres des équipes en valorisant ce que ces flux peuvent apporter aux collaborateurs
- « feedback actionnable » : il arrive qu'un commentaire, un retour utilisateur soit nécessaire à l'équipe pour corriger, modifier des éléments d'un projet. Mais comment faire concrètement ? L'idée est de pouvoir utiliser directement le commentaire. Par exemple, un testeur pointe un défaut de développement, il

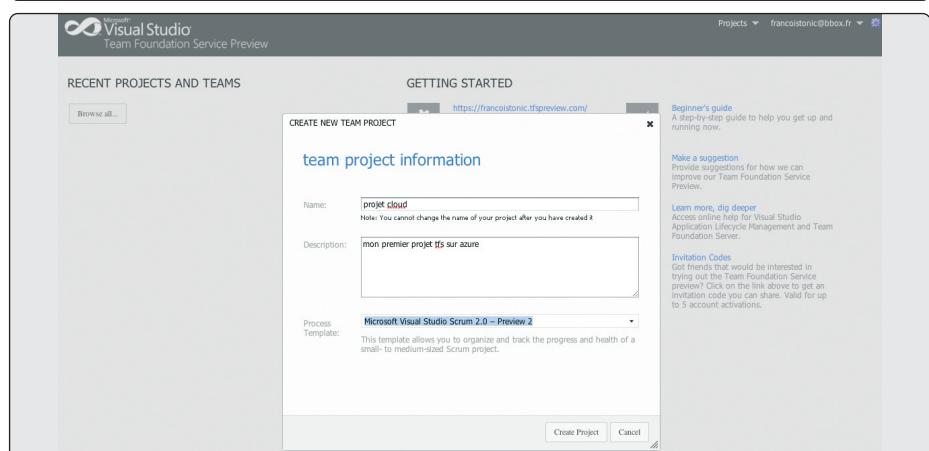
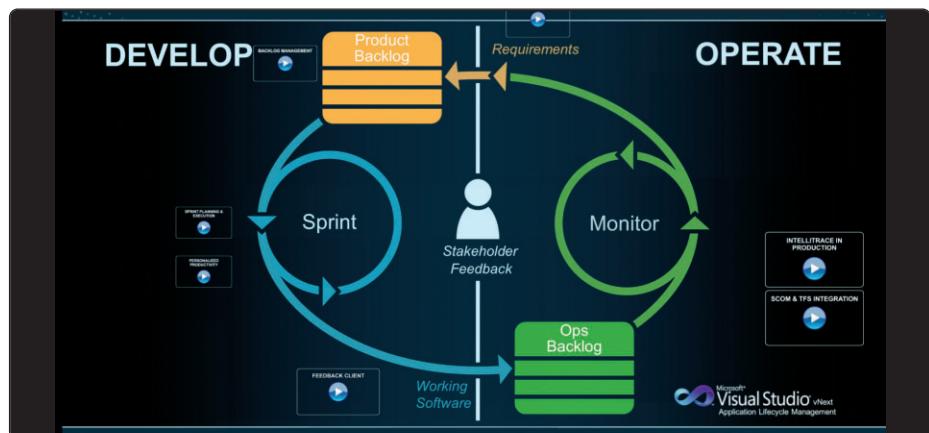
TFS prend le virage cloud !

Dévoilé officiellement en septembre 2011, Team Foundation Server sur Windows Azure (préversion) est un portage TFS sur la plate-forme Windows Azure de Microsoft. Il s'agit d'un service hébergé TFS. En quelques minutes, il est possible de mettre en place un projet et surtout de disposer d'un TFS sans avoir besoin de le déployer localement. Si à l'heure actuelle, ce TFS n'est pas complet, il propose tout de même des fonctions très complètes :

- contrôle de source
- suivi des work item
- gestion des projets agiles (nouveauté de TFS 11)
- outils de tests
- automatisation des builds

TFS pour Windows Azure se veut ouvert et peut s'interconnecter avec Visual Studio 11, Team Explorer Everywhere, VS 2010 et Test Professional 2010. Le démarrage se déroule en trois grandes étapes :

- 1 - URL du compte pour accéder à son TFS Azure
- 2 - création de l'équipe projet
- 3 - installation des clients nécessaires pour connecter l'équipe à TFS...



devrait pouvoir agrémenter son commentaire de vidéos, images, informations précises sur la configuration, un log Intellitrace. Bref, tout ce qui serait utile aux développeurs pour résoudre, corriger un problème.

- Les styles de travail : adapter les meilleurs outils pour les différents membres des équipes. Car les attentes, les besoins d'un développeur, architecte, testeur... ne sont pas identiques et leurs manières de travailler sont parfois très différentes.
- Processus agiles transparents : que toute l'équipe travaille sur une source unique vérifiée et qu'on puisse avoir à tout moment l'exacte situation du projet, des tâches, des sources.

Une collaboration toujours plus fluide

La prochaine version doit supprimer toutes les petites choses qui détournent testeurs, développeurs et l'ensemble des équipes de leurs tâches, du projet. Il s'agit aussi d'améliorer les flux d'informations entre les utilisateurs et l'équipe et entre le Product Owner (Scrum) et les utilisateurs. Cela passe par une transparence dans les flux, la planification. Il s'agit aussi de pouvoir récolter et utiliser les commentaires, les

retours utilisateurs le plus rapidement possible et de les utiliser immédiatement. Pour le développement, il s'agit d'augmenter l'assurance qualité par une amélioration du code, de la revue de code, des frameworks de tests unitaires, dans l'exploration des tests et des tâches au quotidien. Parmi les nombreuses nouveautés (voir page suivante), le lab management sera plus simple à utiliser et à mettre en œuvre. Pas la peine de disposer d'un serveur Hyper-V et de System Center Virtual Machine Manager. Un simple serveur (virtualisé ou physique, avec tous les éléments nécessaires) sera requis ! Et ça, c'est une super nouvelle ! Les préversions VS 11 et TFS 11 sont disponibles en téléchargement. Enjoy !

Ressources

Site officiel : <http://www.microsoft.com/visual-studio/en-us/roadmap>

Blog en Français pointu sur le sujet : <http://blogs.developpeur.org/etienne/>

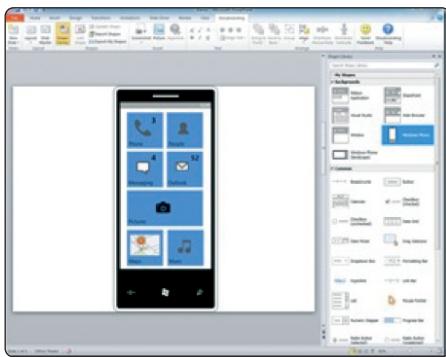
Blog sur Lab Management : <http://blogs.msdn.com/b/bharry/archive/2011/10/31/lab-management-improvements-in-tfs-11.aspx>

François Tonic

Les principales nouveautés VS 11 sur le cycle de vie

1 GESTION DES EXIGENCES "LÉGÈRE"

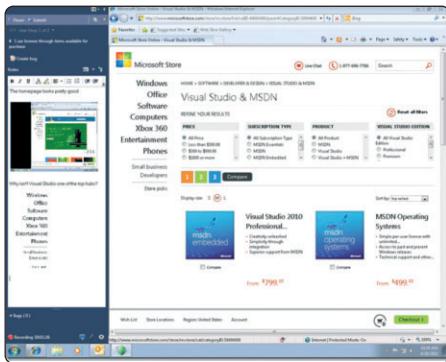
Afin de pouvoir gérer de manière simple et rapide les exigences on aura un plugin pour Powerpoint permettant de rapidement créer une maquette de l'application. L'avantage, outre le fait que Powerpoint est un outil familier, est, qu'une fois la maquette finie (ou suffisamment avancée), on dispose automatiquement d'une présentation simple et efficace permettant de récupérer des commentaires des stakeholders (intervenants)



2 COMMENTAIRES/RETOURS DES STAKEHOLDERS

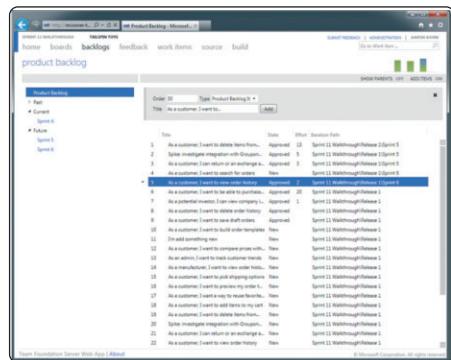
Le maquettage dans Powerpoint permet un premier feedback avant le développement de l'application. Afin de permettre d'obtenir du feedback lors du développement, Microsoft va proposer un outil se basant sur le TestRunner de VisualStudio Test Professional 2010. Une fois lancé, l'outil collectera des données (vidéo, son, actions) pendant que l'utilisateur se servira de l'application. Il pourra à tout moment faire un commentaire qui se traduira par la création d'un work item dans TFS contenant l'ensemble des données collectées.

Au niveau du web access on retrouvera aussi un



onglet dédié au feedback pour permettre une meilleure communication entre équipe de développement et les intervenants.

3 PLANIFICATION AGILE



Les méthodologies Agile sont de plus en plus utilisées. Microsoft a donc revu son interface web access pour simplifier la partie planification. On va donc retrouver :

- Une gestion du product backlog avec possibilité de réordonner facilement les items par glisser/déposer.
- La gestion de la capacité directement dans l'interface web (il fallait passer par les workbooks Excel en 2010)
- Un taskboard afin de suivre et mettre à jour par glisser/déposer le statut des work items.

Comme vous pouvez le voir aussi sur ces captures d'écran, le style complet du site web access a été repris pour suivre les guidelines METRO. L'utilisation de METRO, du HTML et d'AJAX font que le site est très réactif et devient vraiment utilisable au jour le jour.

4 TESTS UNITAIRES CONTINUS

L'utilisation des tests unitaires n'est plus à démontrer. Par contre dans la pratique cela peut être contraignant car il faut penser à les exécuter, en fonction du nombre, cela peut prendre du temps etc. La prochaine version de Visual Studio apportera plusieurs améliorations :

- L'interface de tests a été revue afin de faire ressortir ce sur quoi le développeur doit se concentrer.
- Les tests sont aussi maintenant exécutés en permanence en tâche de fond.
- L'explorateur de tests unitaires supportera de nouveaux framework de tests !! On retrouvera en plus de MSTest, NUnit et XUnit. Si votre fra-

mework préféré n'est pas un de ceux-là, pas de problème, Microsoft a prévu un point d'extensibilité permettant à qui le veut de rajouter le support d'un autre framework ☺

5 TESTS EXPLORATOIRES

Il arrive que certains bogues soient trouvés en dehors de l'exécution des tests de recettes. On parle alors de tests exploratoires où l'utilisateur est libre de ses actions et peut ainsi découvrir des bogues lors d'un scénario non prévu. Il est souvent difficile dans ce cas de reproduire le bug. Pour améliorer cela, la prochaine version de Visual Studio va supporter ce type de test en proposant un Test Runner "libre" (sans étape). Comme pour le test runner classique, les données et actions de l'utilisateur seront enregistrées.

6 INTÉGRATION AVEC SYSTEM CENTER

Afin de supporter les interactions avec les opérations, Microsoft va fournir un nouveau connecteur pour System Center Operation Manager afin de pouvoir créer des work items dans TFS directement depuis SCOM lorsqu'un problème intervient en production. Pour tous les fans de l'IntelliTrace, sachez que Microsoft prévoit de fournir une version pour la production !

7 MEILLEURE EXPÉRIENCE UTILISATEUR

D'une manière générale, l'onglet Team Explorer de Visual Studio va être repris et devenir un Team Navigator. Il sera plus simple de voir les informations et cela apportera une grande nouveauté : la gestion du contexte de travail. Qu'est-ce que c'est ? Lorsqu'un développeur voudra travailler sur une tâche, il indiquera via l'interface qu'il commence à travailler dessus. Visual Studio pourra alors suivre les modifications qui sont faites et les liera à la tâche en cours. Si le développeur doit changer de tâche (pour corriger un bug par exemple), Visual Studio prendra l'ensemble du contexte en cours et le sauvegardera en le liant à la tâche. L'utilisateur pourra ensuite travailler sur autre chose et lorsqu'il indiquera qu'il reprend son travail sur la première tâche, son contexte sera automatiquement restauré !

Guillaume Rouchon

Wygwam

<http://blog.qetza.net/fr/tag/team-foundation-server-vnext/>

Suite de la page 38

fonction malloc() du langage C. Pour calculer la taille de nos matrices (`nbCol * nbRow * sizeof(type)`), nous multiplions le nombre de colonnes par le nombre de lignes (ici c'est une matrice carrée) et par la taille de l'item, ici un nombre flottant.

```
void MatrixMultiplyHost(float* A, float* B, float* C, int size)
{
    float* deviceA;
    float* deviceB;
    float* deviceC;
    size_t sizeAlloc = size * size * sizeof(float);

    cudaMalloc((void**)&deviceA, sizeAlloc);
    cudaMalloc((void**)&deviceB, sizeAlloc);
    cudaMalloc((void**)&deviceC, sizeAlloc);
}
```

Les variables noyau et leurs allocations

On passe à la fonction `cudaMalloc` l'adresse du pointeur mémoire devant être alloué et la taille du buffer à allouer. Les matrices A et B sont les valeurs d'entrées. La matrice C représente la matrice résultat en sortie.

2 - Copier les paramètres vers la mémoire GPU

Après avoir alloué les variables GPU, nous devons recopier respectivement le contenu des matrices A, B, C dans la variable deviceA, deviceB, deviceC. Nous utilisons la fonction `cudaMemcpy` et le drapeau `cudaMemcpyHostToDevice` afin d'assurer le transfert des données dans le sens CPU vers GPU.

```
cudaMemcpy(deviceA, A, sizeAlloc, cudaMemcpyHostToDevice);
cudaMemcpy(deviceB, B, sizeAlloc, cudaMemcpyHostToDevice);
cudaMemcpy(deviceC, C, sizeAlloc, cudaMemcpyHostToDevice);
```

Copier les variables hosts dans les variables noyaux

Arrivé à ce stade, les trois matrices « noyaux » deviceA, deviceB et deviceC sont allouées et initialisées et recopiées dans la mémoire globale GPU.

3 - Appel de la fonction noyau

Nos variables sont prêtes et transmises au code noyau, nous pouvons donc appeler la fonction noyau. Cependant, nous devons préciser combien de fois la carte graphique devra exécuter notre algorithme. En effet en programmation CUDA C, nous devons préciser combien de fois le code noyau devra être exécuté. Nous touchons un point très important de la programmation parallèle sur GPU. En programmation CUDA C, les boucles sont prises en charge par le moteur d'exécution CUDA et c'est donc le code appelant, côté host, qui se doit de préciser le nombre d'itérations à réaliser.

```
MatrixMultiply<<<numBlocks, threadsPerBlock>>>(deviceA, deviceB,
deviceC, size);
```

Remarque : La programmation parallèle GPU permet de résoudre des problèmes de performance sur des algorithmes orientés données. D'un point vue programmation, cela revient à optimiser les boucles « for » des implémentations et dans notre cas nous sommes dans le cadre de deux boucles imbriquées.

Le lancement de notre méthode noyau doit respecter la syntaxe suivante ; nom de la méthode, suivi de trois chevrons ouvrant, le nombre de blocs nécessaires, une virgule, le nombre de threads par

blocs, suivi de trois chevrons fermants. Le reste de la signature ne présente pas de différence avec une fonction C standard. Si vous souhaitez lancer 512 fois votre fonction, vous pouvez par exemple vous exprimer ainsi :

```
MatrixMultiply <<<1, 512>>>(deviceA, deviceB, deviceC, size);
```

Remarque : La syntaxe pour appeler une fonction noyau ne peut être reconnue par un compilateur C standard car la syntaxe avec les chevrons ne sera pas reconnue. Pour que votre compilateur CUDA reconnaisse ce code, vous devez le placer dans un fichier de type « .cu »

Dans notre exemple, nous souhaitons calculer le produit matriciel de matrices 1024 * 1024. Comment faire pour exécuter **1 048 576** threads exigés par notre traitement. Rappelez-vous de l'organisation des threads sur l'architecture CUDA, nous disposons de nombreux blocs. Dans les faits, nous disposons d'un cube de blocs (1024, 1024, 64) par grille. Pour calculer le nombre d'itérations des deux boucles externes de notre algorithme, nous utiliserons deux variables **CUDA** de type `dim3` supportant une configuration cubique (x, y, z). Dans notre cas nous n'utiliserons que deux dimensions des structures `dim3`: `threadsPerBlock` pour définir combien de threads seront créés par bloc et `numBlocks` pour définir combien de blocs seront utilisés. Cette technique de répartition entre les blocs et le nombre de threads par bloc permet d'exploiter la topologie de threads CUDA efficacement. Par exemple, nous définissons pour chaque bloc de threads la taille suivante : `BLOCK_SIZE * BLOCK_SIZE`.

```
dim3 threadsPerBlock(BLOCK_SIZE, BLOCK_SIZE);
dim3 numBlocks(adjust(size) / threadsPerBlock.x, adjust(size) /
threadsPerBlock.y);
```

Déclaration des variables spécifiant le nombre de threads à lancer

Sachant que la valeur de `BLOCK_SIZE` est de **32** dans notre programme, ce qui signifie que nous définissons un plan de 1024 threads par bloc. Côté blocs, nous définissons un plan de $1024/32 * 1024/32 : 1024$ blocs. Finalement, nous obtenons les 1 048 576 threads attendus. En procédant ainsi, nous répartissons notre demande de calcul sur l'architecture CUDA sans difficulté.

Cependant, l'utilisation de `BLOCK_SIZE`, peut poser un problème. Pour éviter des soucis, si la valeur size n'est pas modulo de la valeur de `BLOCK_SIZE`, nous passons par l'intermédiaire de la fonction utile `adjust()`, qui nous assure un résultat positif, mais parfois un peu supérieur à notre demande.

```
int adjust(int n)
{
    if (n < BLOCK_SIZE) { return n; }
    return (n / BLOCK_SIZE + (n % BLOCK_SIZE == 0 ? 0 : 1)) *
BLOCK_SIZE;
}
```

Ajuster le nombre de threads vis-à-vis de la valeur BLOCK_SIZE

Nous verrons, lorsque nous étudierons le code noyau, comment rectifier ce petit excès potentiel dans le nombre d'itérations.

```
MatrixMultiply<<<numBlocks, threadsPerBlock>>>(deviceA, deviceB,
deviceC, size);
```

Lancer le traitement sur la carte graphique

Remarque : si vous lancez un nombre de threads supérieurs à la capacité de votre carte graphique, le moteur d'exécution CUDA retournera une erreur de type : « invalid configuration argument ». Il est donc important d'être vigilant sur le paramétrage de fonction noyau afin d'éviter de mauvaises surprises.

Après le lancement de la fonction noyau, le code noyau va s'exécuter sur l'architecture CUDA. Noter qu'il n'est plus possible de modifier le nombre de threads une fois, le traitement lancé.

4 - Exécuter l'algorithme sur l'architecture CUDA de votre carte graphique

Du côté de l'implémentation de notre calcul matriciel, vous allez sans doute noter que le code de l'algorithme a un peu changé.

```
__global__
void MatrixMultiply(float* A, float* B, float* C, int size)
{
    int i = blockIdx.y * blockDim.y + threadIdx.y;
    int j = blockIdx.x * blockDim.x + threadIdx.x;

    if (j < size && i < size)
    {
        float sum = 0;
        for (int k = 0; k < size; ++k) {
            sum += A[i * size + k] * B[k * size + j];
        }
        C[i * size + j] = sum;
    }
}
```

Code de notre algorithme décliné en une fonction kernel

Le mot clef `__global__` en préfixe de la définition de la fonction `MatrixMultiply` « Noyau » permet de signaler au compilateur CUDA, `nvcc`, que cette fonction est globale et donc appellable depuis le host, mais s'exécute exclusivement sur la carte graphique.

Les extensions du compilateur CUDA C offrent plusieurs types de décorations pour les fonctions. Le tableau ci-dessous, résume les usages possibles des différents types de décoration.

	S'exécute sur le	Appelable seulement depuis le
<code>_device_</code> <code>_double*</code> <code>SubCompute(...)</code>	device	device
<code>__global__</code> <code>void MatrixMultiply(...)</code>	device	host
<code>host</code> <code>void MatrixMultiplyCudaSquare(...)</code>	host	host

Si votre fonction kernel doit être décomposée en sous-fonctions devant être appelables depuis le périphérique, les sous-fonctions doivent être décorées avec le préfixe le mot clef `_device_`. Pour les fonctions s'exécutant exclusivement sur le CPU, le préfixe `_host_` est utilisable, mais pas utile pour compiler correctement.

Contrairement à la version séquentielle où nous bouclions sur les lignes et les colonnes des matrices A et B, l'exécution GPU assure ces boucles en répartissant le code de notre algorithme (code noyau). Sur chacun des threads GPU réclamés (**1 048 576** threads), chaque instance de la fonction noyau se trouve dupliquée sur chacun des threads impliqués dans notre traitement. Ainsi, chaque code noyau obtiendra automatiquement les informations sur les indices du thread GPU qui l'exécute.

```
int i = blockIdx.y * blockDim.y + threadIdx.y;
int j = blockIdx.x * blockDim.x + threadIdx.x;
```

Révision de nos variables vis-à-vis du moteur d'exécution CUDA

Les variables multidimensionnelles `threadIdx`, `blockDim` et `blockIdx` permettent de retrouver les indices du thread courant, afin de renseigner nos variables d'origines, `i` et `j`. Dans notre exemple de produit matriciel, nous disposons de deux boucles externes, indicées respectivement par `i` et `j`. On note que suite au calcul sur le nombre de threads réclamé par le code appelant, le code noyau se protège via l'instruction `if`, pour ne traiter que les indices valides vis-à-vis de la taille de nos matrices. L'architecture CUDA, permet de répartir le calcul sur une grille de threads GPU. Sur la [Fig.5], nous comparons le code séquentiel et le code noyau CUDA. À gauche, le code séquentiel pris en charge par un thread CPU, à droite le code noyau pris en charge par **1 048 576** threads GPU. Dans le code noyau, les deux boucles externes ont disparu, mais les variables globales `blockIdx.y * blockDim.y + threadIdx.y` et `blockIdx.x * blockDim.x + threadIdx.x`, sont renseignées par le moteur d'exécution pour chaque instance de thread CUDA utilisée. Enfin, nous le signalons dans l'étape 3, nous devons protéger notre code d'itérations excédentaires en filtrant les indices avant de lancer un calcul.

```
if (j < size && i < size)
{
}
```

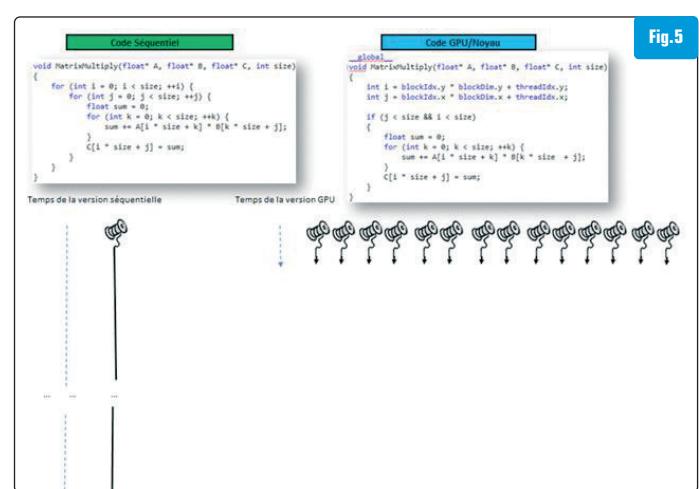
Ceci peut vous surprendre, mais le coût d'un thread GPU est si faible, qu'un petit excès d'itération ne change pas grand-chose sur le plan des performances.

5 - Retour du mode GPU/noyau vers le mode CPU/host

L'enchaînement des différentes étapes réalisées pour lancer une fonction noyau n'est pas vraiment correct. En effet, les appels noyaux, sont par défaut asynchrones et de ce fait le retour d'un appel noyau est immédiat. Ce mode d'appel est parfaitement justifié, car il permet d'enchaîner plusieurs appels noyaux depuis le code CPU/host sans attendre l'exécution effective des calculs. En d'autres mots, le code GPU s'exécute parallèlement sans aucune contrainte de synchronisation.

6 - Copier les résultats en mode CPU/Host vers la matrice C

Dans l'étape précédente, nous avons précisé que le code noyau s'exécutait de manière asynchrone, mais alors que se passe-t-il, lorsque la fonction `cudaMemcpy()` est appelée ?



Comparatif d'exécution entre le code séquentiel sur CPU et le code parallèle sur GPU

```
cudaMemcpy(C, deviceC, sizeAlloc, cudaMemcpyDeviceToHost);
Copier la matrice deviceC en provenance du noyau vers la matrice C allouée
côté host
```

La réponse est simple, l'appel à la fonction **cudaMemcpy()** provoque un point de synchronisation avec le code GPU. Le drapeau **cudaMemcpyDeviceToHost** assure la copie des octets en provenance de la fonction **kernel** vers la mémoire host.

7 - Détruire la mémoire des buffers utilisés par la fonction noyau

La mémoire allouée dans la mémoire globale GPU doit être libérée sous peine de créer une fuite mémoire GPU.

```
cudaFree(deviceA);
cudaFree(deviceB);
cudaFree(deviceC);
}
```

Libérer toutes les matrices noyau en fin de traitement

La fonction **cudaFree()** reprend le contrat de la fonction **free()** en langage C. Si vous disposez de tous les codes présentés au sein d'un fichier **.cu** dans un projet C++ qui respecte le paramétrage présenté dans l'article précédent, alors vous devriez obtenir un binaire opérationnel pour réaliser une exécution.

EXÉCUTION DU PRODUIT MATRICIEL

Après avoir compilé notre code en mode Release en 64bit, nous sommes enfin prêts pour lancer notre produit matriciel avec le Framework **CUDA C**. Nous pouvons lancer à la fois notre traitement **CUDA**, que nous appellerons **CUDA Simple**, précédé des traitements avec un seul cœur et un traitement sur quatre cœurs via la méthode **PPL parallel_for** présentée dans le premier article [Fig.6]. Les résultats sont très encourageants, la technologie **CUDA C** améliore considérablement les performances de notre calcul matriciel [Fig.7]. Les performances du code **CUDA** sont extrêmement bonnes vis-à-vis du code **parallel_for** sur 4 cœurs et incomparablement plus rapide que l'exécution sur un seul cœur. Si vous étiez encore sceptiques sur l'intérêt des GPU, je crois qu'il n'y a plus de doute possible lorsque votre algorithme est massivement orienté données.

CODES SOURCES

L'initiation à la programmation GPU est un sujet délicat qui doit être réalisé avec des codes sources afin de mieux comprendre « comment ça marche ». Je vous invite à télécharger les sources qui m'ont permis d'écrire cet article. Le fichier ZIP contient la solution Visual Studio 2010 « **MulMatrix.sln** », contenant deux projets : « **MulMatrix** » et « **MulMatrixCuda** ».

- Le projet **MulMatrix** est une simple console, il ne contient que du code CPU, comme le code séquentiel, parallèle avec PPL et l'appel au code CUDA.

Fig.6

NVIDIA Quadro 2000M

Matrices A(1024,1024) x B(1024,1024) - single core
Elapsed time: 32835.1 ms

Matrices A(1024,1024) x B(1024,1024) - parallel_for
Elapsed time: 12191.5 ms - PASSED

Matrices A(1024,1024) x B(1024,1024) - CUDA Simple
Elapsed time: 184.031 ms - PASSED

Press any key to continue . . .

Exécution de nos calculs matriciels

- Le projet **MulMatrix** est un projet type DLL, il contient à la fois du code host et du code **device** dans le fichier **MulMatrix.cu**.

Point d'entrée de la démonstration.

```
int _tmain(int argc, _TCHAR* argv[])
{
    try
    {
        print_device();

        Demo demo;

        demo.run(Demo::SERIAL);

        demo.run(Demo::PARALLEL_FOR, false);
        demo.run(Demo::PARALLEL_FOR);

        demo.run(Demo::CUDA, false);
        demo.run(Demo::CUDA);

    }
    catch (std::exception* e)
    {
        cerr << e->what();
    }
    return 0;
}
```

Pour éviter des mesures de performance altérées par des phénomènes d'initialisation (chargement de DLL, Runtime ...), les traitements sont exécutés deux fois. Une première exécution avec un nombre d'itérations réduites est lancée sans afficher de résultats. Dans le code ci-dessus la méthode **run** permet d'indiquer si vous souhaitez mesurer ou pas le traitement.

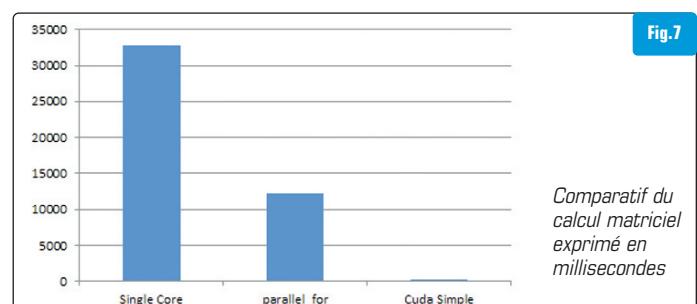
EN CONCLUSION

Les architectures CPU et GPU sont fortement différentes et les conséquences pour les développeurs sont importantes. Nous avons appris comment implémenter notre programme matriciel sur une architecture CUDA. Nous avons appris à lancer un traitement noyau en exploitant efficacement la topologie CUDA. Nous avons aussi montré les différentes étapes permettant de lancer, d'exécuter puis de terminer une fonction noyau vis-à-vis de son programme hôte. Et enfin, nous avons comparé les exécutions de la version CUDA de notre algorithme vis-à-vis des versions séquentielles et parallèles sur CPU.

Bruno Boucard

Indépendant, expert en parallélisme. MVP Technical Computing

<http://blogs.msdn.com/b/devpara/>



PhpStorm : accélérer les développements PHP



Pour la réalisation de projets conséquents, et dans l'optique d'optimiser la productivité des développeurs, l'utilisation maîtrisée d'un EDI (Environnement de Développement Intégré) complet est un plus indéniable. Dans le monde PHP, les deux plus connus sont très certainement les EDI libres Eclipse et Netbeans, à l'origine destinés au langage Java.

Bien que très complets, ces ateliers souffrent parfois de lacunes, en termes notamment de configuration, d'encombrement mémoire ou encore d'ergonomie. Un EDI concurrent, propriétaire cette fois, comble efficacement ces lacunes et propose par la même occasion des fonctionnalités complémentaires : il s'agit de PhpStorm de JetBrains. Cet éditeur de logiciels s'est tout d'abord fait connaître dans le monde Java avec son premier produit, l'EDI IntelliJ IDEA. PhpStorm reprend l'interface et les fonctionnalités d'IntelliJ pour les adapter au langage PHP. Nous profitons de la sortie récente de la version 3.0 pour présenter cet outil taillé pour les projets de grande envergure. Dans un premier temps, nous en décrivons l'interface en général, et dans un second temps, nous présentons quelques fonctionnalités choisies pour leur nouveauté ou leur intégration particulièrement efficace.

PRÉSENTATION GÉNÉRALE

On attend principalement d'un EDI qu'il regroupe dans une même interface les fonctionnalités indispensables à la réalisation et au suivi du projet tout au long de son cycle de vie. On y retrouve donc bien évidemment un éditeur performant simplifiant l'écriture de code, mais aussi idéalement l'intégration d'un débogueur,

l'interfaçage avec un outil de gestion de versions, l'interfaçage avec un outil de gestion de tâches, la prise en charge des tests unitaires ou encore des outils de déploiement. PhpStorm intègre tout ceci de façon intuitive et efficace. Nous présentons ici son interface et ses principales fonctionnalités.

> Ecran d'accueil

L'écran d'accueil [Fig.1] présente les raccourcis habituels de création ou d'ouverture de projets, mais il propose également par défaut la création d'un nouveau projet à partir d'un outil de gestion de versions [Fig.2] là où chez les concurrents il faut passer par l'installation et la configuration d'un plugin. La création d'un nouveau projet peut aussi se faire en ouvrant simplement un répertoire contenant des fichiers sources. A contrario de la création à partir de sources existantes qui offre la possibilité de récupérer des sources sur un FTP ou un partage réseau et présente des options complémentaires de déploiement, aucune configuration n'est ici nécessaire. Le répertoire est ouvert

comme un projet, et après sa fermeture, il apparaît simplement dans la liste des projets récents.

> Présentation de l'interface

Une fois le projet ouvert, on retrouve le même type d'interface que pour les EDI concurrents [Fig.3]. L'interface se décompose en différentes fenêtres, que l'utilisateur peut configurer à sa convenance. Par défaut, on observe un découpage en trois fenêtres principales :

1. La plus grande, centrale, consacrée à l'édition des fichiers sources à proprement parler. Les différents fichiers ouverts apparaissent sous forme d'onglets portant le nom du fichier (dont la couleur change suivant le statut en cas d'utilisation d'un outil de gestion de versions).
2. Une fenêtre sur la gauche consacrée à la navigation à l'intérieur de la solution et des fichiers. Plusieurs choix sont offerts (sous forme d'onglets sur la gauche). La navigation peut se faire dans l'arborescence des fichiers de la solution ou par « structure », comme par exemple dans l'arborescence d'un fichier XML ou HTML ou parmi les classes PHP, leurs méthodes, etc.
3. Une fenêtre en bas utilisée pour le débogage, l'interface avec l'outil de gestion de versions ou encore la liste des « TODO » ici aussi disponibles sous forme d'onglets.

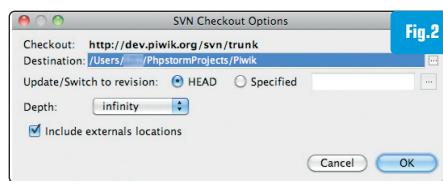


Fig.2

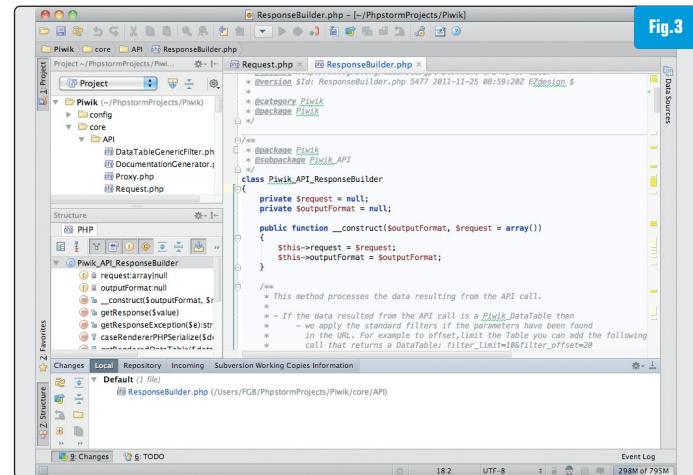
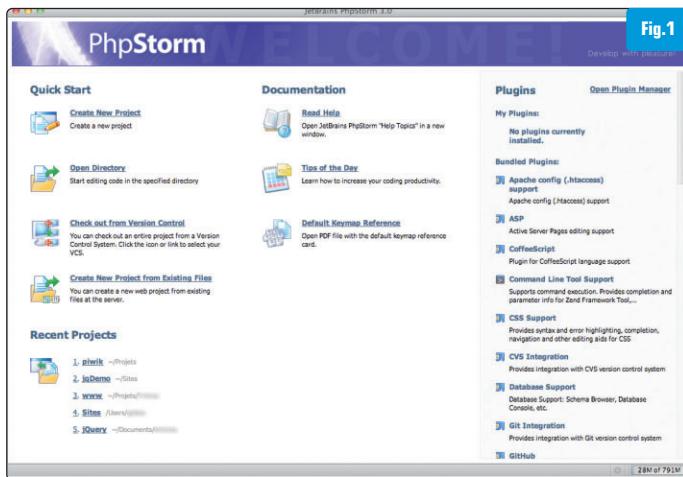


Fig.3



Au premier abord rien d'exceptionnel donc, mais l'interface est intuitive et truffée de petites fonctionnalités pratiques, comme la localisation, dans l'arborescence du projet, du fichier en cours d'édition. De plus, l'interface est entièrement personnalisable. La plupart des fonctionnalités présentées ci-après sont accessibles par l'utilisation de raccourcis également personnalisables. Un PDF présent dans l'aide propose une synthèse des principaux raccourcis. Par défaut, une popup « trucs et astuces » s'ouvre à chaque démarrage pour présenter une fonctionnalité et son raccourci associé, s'il existe, ce qui permet de se familiariser rapidement avec l'interface. On regrettera cependant qu'il ne soit pas possible d'ouvrir plus d'un projet à la fois dans une fenêtre en utilisant le navigateur de projet sur la gauche pour passer de l'un à l'autre. La seule possibilité est d'utiliser une fenêtre différente par projet. Un autre aspect peut être gênant, pour les allergiques à la langue de Shakespeare, l'interface n'est pas disponible en français.

> Fonctionnalités principales

Pour répondre aux besoins très variés rencontrés tout au long du cycle de vie d'une application, l'EDI doit intégrer de nombreuses fonctionnalités. Comme NetBeans ou Eclipse, PhpStorm a choisi d'utiliser un système de plugins. Il est donc livré avec de nombreux plugins qu'il est possible de désactiver, mais on peut aussi en ajouter pour répondre à de nouveaux besoins.

L'édition de code

L'activité principale du développeur est l'écriture de code. C'est donc aux fonctionnalités d'édition que nous nous intéressons tout d'abord. PhpStorm propose bien évidemment les fonctionnalités courantes de coloration syntaxique, d'auto-complétion, d'indentation automatique, de masquage de portions de code, ou encore d'affichage de la

documentation des méthodes et fonctions. PhpStorm n'est d'ailleurs pas uniquement compatible avec PHP. En effet, il reconnaît très bien par exemple le HTML, le JavaScript, le CSS ou encore le SQL. La fonction d'auto-complétion ne se contente pas de la classique complétion de code, elle permet également de compléter automatiquement les classes CSS ou encore les chemins vers des fichiers. Dans le même esprit, avec l'utilisation de petites icônes et d'un code couleur judicieux, PhpStorm met en évidence les portions de code inutiles, erronées, qui peuvent être simplifiées ou encore qui comportent des erreurs orthographiques.

Pour générer automatiquement les portions de code répétitives, PhpStorm propose un certain nombre de templates pour créer simplement des squelettes de classes, de méthodes, de boucles, etc. L'utilisateur peut bien sûr ajouter ses propres templates. D'autres fonctions, à l'instar de « entourer de », du déplacement de portions de code par l'intermédiaire de raccourcis claviers, ou encore de la génération de code [Fig.4] permettent d'éviter des copier/coller fastidieux. PhpStorm dispose également d'un outil de comparaison et de fusion qui s'utilise au choix sur deux fichiers locaux, sur un fichier par rapport à une version donnée sur l'outil de gestion de version, sur un fichier par rapport à un fichier sur un FTP ou bien sur une portion de code par rapport au contenu du presse-papier [Fig.5].

La navigation au sein du code est simplifiée grâce à l'utilisation de fonctions de recherche ciblées (dans tous les fichiers du projet, dans tous les fichiers d'un répertoire donné, dans tous les fichiers de production, tous les fichiers de test, etc.), employées pour trouver l'implémentation d'une méthode, ses différentes utilisations, ou simplement une chaîne de caractères.

Enfin, les principales tâches de refactoring telles que le renommage ou la suppression

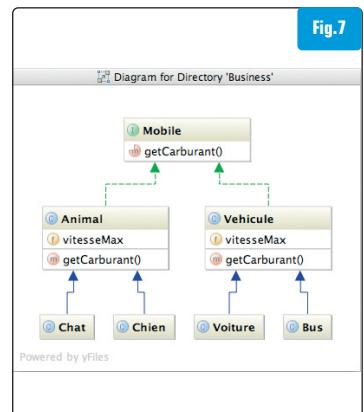
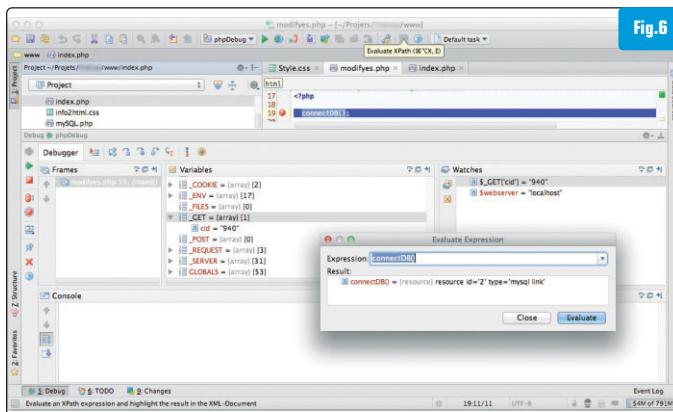
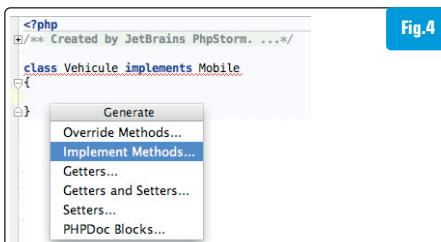
intelligente sont disponibles et prennent en compte, si nécessaire, les commentaires ou le changement automatique des noms de fichiers (lors du renommage d'une classe par exemple).

Versions et Gestion des tâches

Par rapport à ses concurrents, PhpStorm a l'avantage d'intégrer un interfaçage avec les principaux outils de gestion de versions (Git, SVN, etc.) et de gestion de tâches (Redmine, YouTrack, etc.) sans installation ou configuration supplémentaire. En plus des fonctionnalités standard d'archivage, la création d'une nouvelle branche ou l'import d'un nouveau projet dans l'outil de gestion de version sont accessibles directement depuis l'EDI, ce qui permet d'éviter au maximum le recours à un client extérieur. De même, l'intégration avec l'outil de gestion de tâches propose l'ajout de nouvelles tâches, la consultation ou encore la résolution directement depuis l'interface. Il est par ailleurs également possible d'associer un archivage à une tâche de façon à la résoudre automatiquement, ou simplement y faire référence.

Qualité du code et débogage

Pour aider les développeurs à fournir un code propre et à déboguer leur application, PhpStorm dispose de nombreux outils. L'inspection de code a pour but de mettre en évidence les erreurs ou alertes des différents scripts. On peut en automatiser le lancement avant chaque archivage pour alerter l'utilisateur le cas échéant. De façon plus graphique, au même niveau que l'ascenseur de navigation dans le fichier ouvert dans l'éditeur, on retrouve une petite colonne permettant de localiser les différentes erreurs ou alertes dans le code. Chaque notification est matérialisée par un petit rectangle coloré par lequel on peut directement se rendre à l'endroit correspondant dans le fichier. PhpStorm gère aussi les tests unitaires



PHPUnit. Il gère notamment le lancement des tests, leur débogage ainsi que la visualisation des résultats.

Déboguer une application PHP avec PhpStorm s'avère très simple. Il n'est point besoin ici de configuration complexe du serveur de développement ou de l'EDI. Il suffit d'activer le débogage avec Xdebug dans le fichier de configuration de PHP : tout est déjà configuré et prêt à fonctionner du côté de PhpStorm [Fig.6]. Le débogage de JavaScript est également intégré à l'EDI, ainsi que le débogage d'applications à distance. Un profiler est inclus dans la dernière version. Par contre, si l'on souhaite déboguer à la fois le JavaScript et le PHP, deux instances de débogage au profil différent sont nécessaires, nous obligeant à passer de l'une à l'autre constamment.

FONCTIONNALITÉS AVANCÉES ET NOUVEAUTÉS

PhpStorm se distingue de la concurrence soit en proposant de nouvelles fonctionnalités, soit en proposant une implémentation plus efficace ou plus complète de certaines fonctionnalités existant dans d'autres EDI. Un accent tout particulier est mis sur l'ergonomie. Loin d'être exhaustifs, les exemples présentés ci-dessous ont pour but de montrer ce qui donne à PhpStorm ce petit plus qui le rend tout de suite confortable à utiliser et rapidement indispensable.

> Diagramme de classes

Un premier exemple, introduit avec la version 3.0 de PhpStorm, est la génération des diagrammes des classes PHP [Fig.7]. Il est par ailleurs possible d'initier des tâches de renommage à partir de ce diagramme. Par contre nous aurions aimé avoir d'autres options de refactoring, comme l'ajout ou la suppression de méthodes ou variables, ou encore la modification de l'héritage, à partir de ce même diagramme.

> Interfaçage avec la base de données

Un autre exemple intéressant est l'interfaçage avec la base de données qui offre l'exécution de scripts sur la base de données tout en simplifiant leur écriture grâce à l'auto-complétion [Fig.8]. Les modifications sur la structure de la base ou sur les données peu-

vent s'effectuer depuis l'interface, graphiquement ou par l'exécution de requêtes, ce qui permet la plupart du temps d'éviter l'utilisation d'outils externes. Malheureusement cette intégration n'est pas complète, l'auto-complétion des noms de champs et de table ne fonctionne pas dans les scripts SQL inclus dans des chaînes de caractère PHP. Il existe cependant des plugins pour répondre à ce besoin.

> Simplifiez-vous le CSS

Certaines fonctionnalités peuvent sembler accessoires mais permettent un gain de temps important. Par exemple, PhpStorm présente les feuilles de style CSS avec des carrés colorés en marge des lignes comportant un code couleur. En plus de donner un aperçu rapide de la couleur correspondante, ces carrés donnent accès, d'un simple clic, à un mélangeur de couleurs épaulé d'une pipette permettant de sélectionner la couleur de n'importe quel pixel affiché à l'écran [Fig.9]. Il est également possible d'afficher tous les styles s'appliquant à une balise HTML donnée (qu'ils soient appliqués directement ou hérités) comme on le ferait avec un outil tel que Firebug. Par ailleurs, le refactoring de CSS est simplifié. Le renomage d'une classe dans la feuille de style provoque la mise à jour du code HTML l'utilisant, mais l'inverse est aussi vrai.

> Intégration avec les frameworks

En complément de la complétion et de l'affichage de l'aide, l'outil de ligne de commande intégré peut être configuré pour reconnaître et exécuter les commandes de différents frameworks. Par défaut, et si le framework correspondant est déjà installé sur le système, les commandes de Zend Framework et de Symfony peuvent être activées. On peut également utiliser l'auto-complétion pour

faciliter l'écriture de ces commandes et les exécuter directement depuis PhpStorm.

> Historique Local

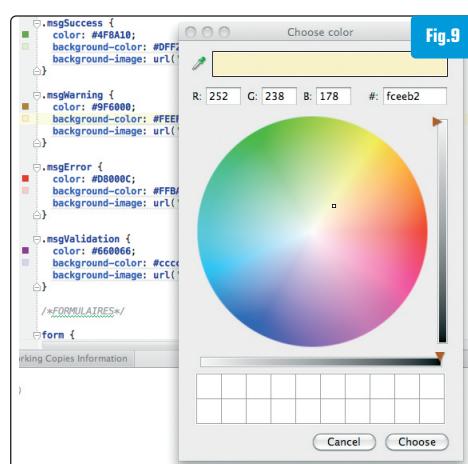
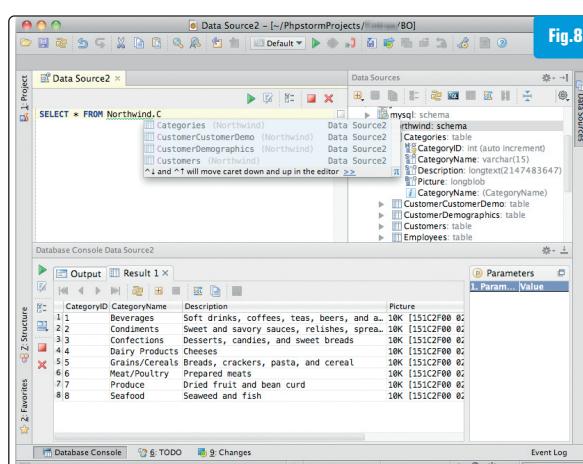
Parallèlement à l'outil de gestion de versions, PhpStorm propose un historique local. Les fichiers sont enregistrés automatiquement, ce qui peut surprendre au départ, mais permet de faire une comparaison et une fusion par rapport à une version antérieure [Fig.10]. On utilise par défaut l'échelle du temps pour retrouver la version souhaitée, mais l'utilisation d'étiquettes est aussi possible.

EN GUISE DE CONCLUSION

Les fonctionnalités présentées dans cet article sont celles installées par défaut avec PhpStorm. Cependant, le système de plugins permet d'ajouter de nouvelles fonctionnalités. Au-delà des fonctionnalités elles-mêmes, l'interface est soignée et intuitive et s'intègre élégamment sous les systèmes d'exploitation Windows, Linux (GTK) ou encore OS X. Techniquement, peu de fonctionnalités proposées par la concurrence sont manquantes. Reste la barrière du prix des licences qui, si elles sont rapidement rentabilisées par les sociétés par une productivité améliorée, peuvent sembler excessives pour un développeur à titre personnel. On peut regretter qu'il n'existe pas une version « community » distribuée librement comme IntelliJ, mais il est cependant possible d'obtenir une licence gratuite si l'on participe à un projet open source. Dans tous les cas, un essai de 30 jours est offert pour tous les outils de JetBrains. Pour plus d'informations, vous pouvez consulter le site de l'éditeur :

<http://www.jetbrains.com/phpstorm/>.

Gilles Bougenière,
consultant Osaxis (www.osaxis.fr)



Bien débuter avec Visual C++

Initialement, le développement sous Windows en C++ n'était possible qu'au travers d'un outil payant comme Visual C++ ou via l'acquisition de la suite Visual Studio. En plus de l'environnement de développement intégré (IDE), le package inclut les différentes en-têtes et librairies pour développer sous Windows alias Windows SDK, anciennement nommé Platform SDK.

> Windows SDK

Le Windows SDK fournit les outils, les compilateurs, les en-têtes et les librairies nécessaires à la création d'application sur les plateformes Windows. Il existe aujourd'hui deux versions majeures de kits de développement « Windows SDK » : Windows SDK pour Windows 7 et .NET Framework 3.5 SP1 ou Windows SDK pour Windows 7 et .NET Framework 4.0. Les deux versions de Windows SDK sont disponibles en téléchargement gratuit. La première version redistribue le compilateur disponible dans Visual C++ 2008 SP1 (VC9) et la deuxième distribue le compilateur de Visual C++ 2010 SP1 (VC10). Pour lancer une compilation en mode de commande, il faut positionner quelques variables d'environnements comme PATH, INCLUDE et LIB. La définition exacte de l'environnement est décrite dans le fichier vcvars32.bat disponible sous %ProgramFiles%\Microsoft Visual Studio 10.0\VC\bin. Le kit Windows SDK fournit aussi un fichier qui permet de créer son environnement de build en utilisant SetEnv.cmd disponible sous %ProgramFiles%\Microsoft SDKs\Windows\v7.1\Bin et dont l'utilisation est la suivante :

```
Usage : Setenv [/Debug | /Release][/x86 | /x64 | /ia64 ][/vista
| /xp | /2003 | /2008 | /win7][-h | /?]
```

Bien qu'il soit possible de compiler directement depuis la ligne de commande, l'utilisation d'un IDE apporte un plus grand confort de travail. Le plus simple est d'utiliser Visual C++ 2010 Express qui est gratuit et fournit le même environnement que Visual Studio 2010. VC++ Express distribue la runtime C, la librairie standard C++ (STL) et d'autres librairies comme PPL et les fichiers du Windows SDK. **Important** : Il faut installer Visual Studio 2010 SP1 après l'installation de Visual C++ 2010 Express. La version Express de Visual C++ 2010 possède quelques limitations comme l'absence des librairies MFC et ATL, pas de compilation 64 bit, pas de gestionnaire de fichiers de ressources (RC editor) pour désigner les interfaces graphiques, pas de support pour OpenMP ni pour les add-ins Visual Studio. Cependant, ces restrictions ne sont pas trop importantes pour quelqu'un qui veut démarrer dans le développement Visual C++. L'utilisation d'une bonne documentation est fondamentale. Il est possible de télécharger la documentation ou de l'utiliser en ligne. Je conseille pour ceux qui débutent une autre option qui est la documentation "MSDN Library pour Visual Studio 2008 SP1". Disponible gratuitement, cette image ISO de 2.2 GB qui contient la fameuse MSDN Library qui est la véritable Bible pour les programmeurs Windows. Pour bien démarrer, il suffit parfois de trouver un programme existant et de l'adapter. Les programmes d'exemples sont disponibles gratuitement. Pour tester notre environnement, tentons de compiler en ligne de commande sur un projet C++ sourceforge comme Scintilla, disponible sur <http://sourceforge.net/projects/scintilla>. Trouvons les deux makefiles et lançons la compilation via Microsoft Visual Studio 2010 Express -> Visual Studio Command Prompt.

```
D:\Files_GNU\scite300\scintilla\win32>nmake -f scintilla.mak
D:\Files_GNU\scite300\scite\win32>nmake -f scite.mak
```

Le fichier makefile contient les différentes commandes (rc, cl, ln) et options pour compiler la solution.

```
rc -fo.\ScintRes.res ScintRes.rc
cl -Zi -TP -W4 -EHsc -Zc:forScope -Zc:wchar_t -D_CRT_SECURE_
NO_DEPRECATED=1 -O1 -MT -DNDEBUG -GL -I../include -I../src -I..
/lexlib -c -Fo.\ ..\src\AutoComplete.cxx ..\src\CallTip.cxx ..
\src\CellBuffer.cxx ...
link -OPT:REF -LTCG -DEBUG -DEF:Scintilla.def -DLL -OUT:..\bin
\Scintilla.dll ..\AutoComplete.obj ...
```

Si la compilation via le makefile est réussie, alors votre environnement Visual C++ est prêt à l'emploi.

> Introduction à C++

Visual C++ supporte le langage C++ et le langage C++/CLI. Le langage C++ standard ISO/IEC, dont la version C++11 vient d'être publiée par Herb Sutter et son équipe au mois de mars 2011 après de nombreuses années d'effort sur la standardisation. C'est le langage C++ standard, celui qui est disponible sur les environnements Windows, Linux et que l'on utilise avec un compilateur comme GCC ou Visual C++. Le deuxième langage C++ supporté est un langage C++/CLI standardisé ECMA qui comporte des extensions spécifiques pour produire du « code managé » à base de C++. Cet article ne traite que du langage C++ standard et n'adresse pas C++/CLI. Dans certains articles, revues ou webcasts, on peut entendre que C++ est un langage « unsafe », non sécurisé et dangereux. Ceci est faux. C'est une ruse pour favoriser l'adoption d'autres langages. C++ est un langage type-safe, qui fournit des mécanismes d'abstractions évolués (classe, template) et qui ne sacrifie pas le pouvoir et les performances.

La Runtime C

Le langage dispose d'une librairie nommée Runtime C qui contient les routines de base pour réaliser des applications. Cette runtime est implémentée en langage C et fournit les fonctions communément définies dans le C ANSI, qui fait l'objet du livre K&R « C ANSI ». Avec Visual C++, le code source de la runtime est fourni dans le répertoire %ProgramFiles%\Microsoft Visual Studio 10.0\VC\crt\src. Dans Visual C++ 2010, la CRT contient 1300 fichiers pour 15 MB de source code. La runtime C fournit les catégories de fonctions suivantes : [tableau 1](#).

La librairie standard C++ (STL)

La librairie STL est constituée de classes et de templates définis en 6 parties : containers, container adapters, iterators, algorithms, function objects et function adapters. Les éléments de la famille « containers » sont équivalents aux collections. Ces classes permettent de stocker,

supprimer ou d'accéder à certains éléments. Voici la liste des containers : **tableau 2**. Les iterators ou itérateurs en français permettent l'accès aux containers. Ils fonctionnent comme des pointeurs et peuvent être de différents styles : input, output, forward, bidirectional, random access. Chaque container utilise un type d'itérateur spécifique. La documentation associée à chaque classe indique lequel utiliser. Un exemple illustre comment utiliser les containers et les itérateurs.

```
#include <vector>
#include <string>
#include <iostream>
using namespace std;
...
```

Tableau 1

Catégorie	Fonctions C	Fichier d'en-têtes
gestion des arguments	va_arg, va_end, va_start	stdarg.h
manipulation de buffer	memcpy, memchr, memcmp, memmove, memset, swab	ctype.h
classification de byte et caractère	isalpha, isalnum, ismbbkalnum, ismbbkprint, isdigit, islower, isupper, isprint, ispunct, isspace, isxdigit...	ctype.h, wchar.h
alignement de données	aligned_malloc, aligned_free, aligned_mszie, aligned_realloc...	malloc.h
conversion de données	abs, atof, atoi, atol, ecvt, fvt, itoa, mbtowc, strtod, strtol, toascii, tolower, toupper...	stdlib.h, math.h
routines de débogage	_ASSERT, _ASERTE, _CrtDbgBreak, _CrtDbgReport, _CrtIsValidPointer, malloc_bdg, free_dbg, realloc_dbg...	crtdbg.h
contrôle de répertoire	chdir, getcwd, getdrive, getdiskfree, mkdir, rmdir, searchenv...	direct.h, errno.h
gestion des erreurs CRT	assert, clearerr, _RPT, set_error_mode...	crtdbg.h
gestion des exceptions et robustesse	terminate, unexpected, set_terminate, set_unexpected...	eh.h
gestion des fichiers	chsize, filelength, fstat, get_osfhandle, locking, setmode, access, chmod, fullpath, makepath, mktemp, remove, rename, splitpath, umask, unlink...	io.h, errno.h
support des flottants	abs, acos, asin, atan, atof, cos, div, floor, fpclass, frexp, hypot, max, min, mod, pow, rand, sin, sqrt, srand, tan...	stdlib.h, math.h
entrées et sorties	Stream I/O: fclose, feof, ferror, fflush, fgetc, fgetpos, fgets, fopen, fprintf, fputc, fputs, fread, fscanf, fseek, fwrite, setbuf... Low-level I/O: close, commit, creat, dup, eof, lseek, open, read, sopeb, tell, umask, write... Console I/O: gets, printf, puts, scanf, getch...	stdio.h, io.h, sys/stat.h, sys/types.h
internationalisation et unicode	setlocale, _UNICODE; _MBCS...	locale.h
gestion de la mémoire	calloc, malloc, delete, expand, mszie, realloc...	stdlib.h, malloc.h
processus et environnement	abort, atexit, _beginthreadex, cwait, execv, execvpe, getenv, getpid, _pipe, _popen, raise, signal, system...	process.h, errno.h
recherche et tri	bsearch, lfind, lsearch, qsort...	stdlib.h, search.h
manipulation de chaînes	strcol, strdec, strncol, strspnp, strcat, scanf, sprintf, strchr, strcmp, strcpy, strdup, strerror, strftime, strlen, strlwr, strcat, strspn, strstr, strtok, strupr... string.h	string.h
appel système	findclose, findfirst, findnext...	io.h
gestion du temps	asctime, clock, ctime, difftime, gmtime, localtime, mktime, strdate, strftime, time...	time.h

```
void routine2()
{
    vector<string> vector1;
    // Ajout d'éléments
    vector1.push_back("C++ Renaissance");
    vector1.push_back("Going Native");
    vector1.push_back("Why C++ ?");
    // Obtention de l'itérateur -> vector<string>::iterator it = vector1.begin()
    for(vector<string>::iterator it = vector1.begin(); it!=vector1.end(); it++)
    {
        cout << "value: " << *it << endl;
    }
    // Itération dans une boucle for classique
    for(size_t i=0 ; i<vector1.size() ; i++)
    {
        cout << "value: " << vector1[i] << endl;
    }
}
```

Pour ajouter des éléments dans un container `vector<T>`, on utilise la méthode `push_back(T)`. Pour itérer dans le container, on obtient l'itérateur propre au container, puis on itère en utilisant l'opérateur `++`. L'accès à l'élément se fait en utilisant l'opérateur `*`. Une itération peut aussi se faire via l'utilisation du `vector<T>` comme un tableau en utilisant l'opérateur `[]`. La STL fournit plusieurs moyens pour réaliser les opérations et ainsi permet plusieurs styles d'écriture de code. Il existe un container très sympa à utiliser qui permet d'associer une clé avec une valeur : `map<K,T>`. L'ajout d'éléments se fait soit en utilisant l'opérateur `[]` ou bien la méthode `insert` qui prend un type `pair<first, second>`. L'itération dans le container se fait en utilisant les membres `first` et `second`. La recherche se fait en utilisant la méthode `count()`. Exemple de code :

Tableau 2

Container	Fichier d'en têtes	Description
vector	<vector>	vector<T> est un tableau qui grossit automatiquement. C'est le container le plus utilisé
deque	<deque>	deque<T> est une double-ended queue. C'est comme un vector mais plus rapide pour l'insertion en début ou en fin
list	<list>	list<T> permet l'ajout et la suppression rapide d'éléments
map	<map>	map<K,T> stocke des couples key/types uniques
multimap	<map>	multimap<K,T> permet que les couples soient identiques
set	<set>	set<T> est un ensemble d'éléments uniques
multiset	<set>	multiset<T> permet que les éléments soient identiques
hash_map	<hash_map>	hash_map<K, T> est un map rapide en accès
hash_multimap	<hash_map>	hash_multimap<K, T> est un multimap rapide en accès
hash_set	<hash_set>	hash_set<T> est un set rapide en accès
hash_multiset	<hash_set>	hash_multiset<T> est un multiset rapide en accès
stack	<stack>	stack<T> implémente LIFO (last in, first out) donc dernier entré, premier sorti
queue	<queue>	queue<T> implémente FIFO(first in, first out) donc premier entré, premier sorti
priority_queue	<queue>	priority_queue<T> est une queue dans laquelle l'élément le plus haut est le premier dans la queue

```
#include <map>
...
void routine3()
{
    map<string, string> dico;
    dico[«VB»] = «Visual Basic»;
    dico[«C#»] = «Visual C# -> Productivity»;
    dico[«C++»] = «Visual C++ -> Power and Performance !»;
    // Ajout en utilisant la fonction make_pair
    dico.insert(make_pair(«C++11», «Fast and Fluid ! See you later
in Windows 8»));
    // Obtention de l'itérateur via container<T>::iterator
    for( map<string, string>::iterator it = dico.begin() ; it!=
dico.end() ; it++)
    {
        cout << «key: » << it->first << « value:» << it->second << endl;
    }
    // count détermine si une clé existe -> retourne 0 ou 1
    if(dico.count(«C++») != 0)
    {
        cout << «C++ value: » << dico[«C++»] << endl;
    }
}
}
```

Les containers de la STL sont génériques, donc on peut les utiliser avec n'importe quel type. Exemple de map avec comme clé un type string et comme valeur un type LangageProp :

```
class LangageProp
{
private:
    string desc, core_value;
public:
    LangageProp(string d, string cv ) : desc(d), core_value(cv) {}
    string Display() { return desc + « is » + core_value; }
};

void routine4()
{
    map<string, LangageProp> dico;
    dico.insert(make_pair(«VB», LangageProp(«Visual Basic», «for the
kids ?»)));
    dico.insert(make_pair(«C#», LangageProp(«Visual C#», «for
productivity»)));
    dico.insert(make_pair(«C++», LangageProp(«Visual C++», «for Power
and Performance !»)));
    // Obtention de l'itérateur via container<T>::iterator
    for( map<string, LangageProp>::iterator it = dico.begin() ; it!=
dico.end() ; it++)
    {
        cout << «key: » << it->first << « object:» << it->second.
Display() << endl;
    }
}
```

Dans cette exemple, on remarquera que le membre second de l'itérateur est de type LangageProp et on peut faire appel directement à la méthode Display() de cette classe. La STL fournit aussi un ensemble de fonctions templates appelées algorithmes disponibles dans le

fichier d'en-têtes <algorithm>. Ce fichier contient plusieurs dizaines de fonctions génériques qui savent traverser les containers et y opérer un certain traitement. La plus connue est sans doute for_each qui permet de faire un même traitement pour l'ensemble des éléments d'un container. Nous reprenons comme base la collection ci-dessus. L'appel à for_each se fait sur la base d'un itérateur de début, d'un itérateur de fin et d'une routine, OnItem, qui va être appelée sur chaque élément du container.

```
void OnItem(pair<string, LangageProp> value)
{
    // retrieve the LangageProp object reference
    LangageProp &prop = value.second;
    cout << «OnItem...» << prop.Display() << endl;
}

void routine5()
{
    map<string, LangageProp> dico;
    dico.insert(make_pair(«VB», LangageProp(«Visual Basic», «for the
kids ?»)));
    dico.insert(make_pair(«C#», LangageProp(«Visual C#», «for
productivity»)));
    dico.insert(make_pair(«C++», LangageProp(«Visual C++», «for
Power and Performance !»)));
    for_each(dico.begin(), dico.end(), OnItem);
}
```

Programmation moderne avec C++11

Écrire du C++ moderne ça veut dire quoi ? Par exemple, on ne va plus explicitement faire de simples new/delete sur nos objets. On va construire et manipuler les objets avec des « smart pointer » via shared_ptr<T>. Ce template implémente un mécanisme de compteur de références qui permet de partager des objets entre plusieurs clients, et permet de stocker des pointeurs dans des containers en toute sécurité.

```
#include <memory>
...
void routine6()
{
    shared_ptr<LangageProp> lp1(new LangageProp(«C++11», «writing
modern C++ code»));
    cout << lp1->Display() << endl;
    shared_ptr<LangageProp> lp2(lp1);
    cout << «use_count should be 2. use_count() ==» << lp2.use
_count() << endl;
}
```

L'abstraction weak_ptr<T> est le meilleur compagnon de shared_ptr<T>. Cela permet de casser les références circulaires. Le template weak_ptr<T> implémente la pattern Observer pour les shared_ptr<T>. Avec un weak_ptr<T>, dès qu'un shared_ptr<T> libère ses ressources, l'information est propagée à tous les weak_ptr<T> afin qu'il ne puisse plus utiliser un pointeur invalide.

Christophe Pichaud - .NET Rangers by Sogeti
 Consultant sur les technologies Microsoft
 christophepichaud@hotmail.com - www.windowscpp.net

TestNG : le framework de tests nouvelle génération !

Créé à la fin des années 90, JUnit est rapidement devenu le standard de facto pour le développement des tests unitaires en Java. Néanmoins, l'outil souffre d'un certain nombre de limitations obligeant les développeurs à mettre au point des contournements pas toujours très heureux. Fort de ce constat, un architecte Java a pris le taureau par les cornes en mettant au point un outil venant corriger ces problèmes. Nommé TestNG, ce framework ne se contente pas de cela puisqu'il apporte une nouvelle approche pour la réalisation de tests en Java. Présentation et prise en main de cet outil destiné aux applications nouvelle génération.

Au départ, il y a un architecte Java français du nom de Cédric Beust qui petit à petit prend conscience des limites de JUnit début 2004. Pis encore, il s'aperçoit que celles-ci ont en fait été implémentées sciemment lors du design du framework. Le premier exemple mis en lumière apparaît lorsque l'on souhaite maintenir un état entre les différents tests contenus au sein d'une classe de tests JUnit. Par défaut, ce dernier ne le permet pas puisqu'il va ré-instancier la classe après l'exécution de chacune des méthodes de tests la composant. La garantie de l'indépendance du contexte d'exécution entre chacun des tests d'une classe étant l'argument mis en avant par les créateurs de JUnit pour justifier cette ré-instantiation systématique. Argument quelque peu fallacieux puisque la méthode setup invoquée entre chaque test doit permettre de garantir cette indépendance ! Pour Cédric Beust et plusieurs autres observateurs du monde Java, il paraît nécessaire de pouvoir conserver une connexion JDBC, un contexte applicatif Spring ou bien le résultat du parsing d'un gros fichier XML afin de minimiser la durée d'exécution des tests.

Afin de contourner cette limitation, les créateurs de JUnit ont conseillé l'utilisation d'un pattern consistant à utiliser une variable de classe statique initialisée au sein d'une méthode de setup évidemment statique appelée avant l'exécution des méthodes de la classe de tests concernée. Poussant à l'utilisation de méthodes statiques venant modifier l'état de la classe, ce pattern se révèle être une très mauvaise solution. Outre les risques intrinsèques de thread-safety lors d'accès multi-threads sur les champs statiques de la classe, un problème plus mystérieux se pose lors de l'exécution des tests JUnit via la tâche Ant dédiée. En effet, sans le positionnement du flag fork à yes sur cette dernière, l'exécution des tests échoue sans raison apparente après plusieurs tentatives. Enfin, ce contournement viole des règles simples de l'orienté objet et met en exergue le caractère intrusif de JUnit qui oblige les développeurs à adapter leur code pour pallier ses manquements internes.

Outre ces problèmes, beaucoup n'étaient pas satisfaits à cause des contraintes imposées par JUnit jusqu'aux versions 3.x quant aux méthodes de tests au niveau de leur nommage et de leur signature. En sus, le modèle de définition des classes de tests via héritage de la classe TestCase imposé par JUnit à l'époque s'avérait bien trop contraignant. Non content d'avoir décrit et critiqué les restrictions imposées par JUnit, Cédric Beust a décidé dans la foulée de proposer une alternative en travaillant sur son propre framework de tests

Java apportant une nouvelle approche au domaine, tout en tenant compte des limitations de JUnit. Profitant de l'arrivée des annotations au sein du langage Java, TestNG a innové avec un modèle de tests basé sur des annotations au centre desquelles on trouve @Test. Preuve du caractère avant-gardiste de TestNG, JUnit 4 en a adopté une grande partie par la suite. On peut ainsi citer le système de définition des tests basé sur des annotations, les tests dont le résultat attendu est une exception, les tests limités dans le temps ou bien le rajout d'un certain nombre de méthodes de configuration. Cette mise en évidence des différents problèmes liés à JUnit ne saurait remettre en cause le rôle de pionnier du framework dans la prise en compte de l'importance des tests unitaires dans le processus de développement logiciel. JUnit aura ainsi démocratisé la pratique en la faisant passer du monde de la QA (Quality Assurance) au monde des développeurs, installant par là-même les tests unitaires en pilier du développement logiciel. Pour le créateur de TestNG, JUnit aura été la première étape dans la démarche de prise en compte des tests dans le monde Java. TestNG doit maintenant permettre de passer à la vitesse supérieure en autorisant le test d'un système entier via la facilitation des tests d'intégration, fonctionnels et d'acceptation. De fait, le scope de TestNG est bien plus large que celui de JUnit utilisé bien souvent pour des besoins dépassant le cadre de sa conception. Ceci ayant rendu fort logiquement l'outil inadapté et compliqué, car non prévu pour des cas d'utilisation hors du cadre strict des tests unitaires.

FONCTIONNALITÉS DE BASE

Disponible à présent en version 6.3, TestNG se compose d'un ensemble d'annotations permettant de configurer les tests. Novateur à sa sortie, ce modèle s'affranchit des contraintes liées aux méthodes de tests tout en bénéficiant de la puissance et de la souplesse des annotations Java. L'outillage autour du framework est excellent avec notamment l'existence de plugins pour les principaux IDE du marché parmi lesquels celui pour Eclipse que nous allons utiliser. Son ergonomie ressemble fortement à ce que l'utilisateur connaît avec JUnit offrant une prise en main immédiate. Un premier exemple trivial va nous donner un aperçu des fonctionnalités de base de TestNG :

```
public class DequeTest {
    private Deque<Integer> deque;
```



```
public DequeTest() {
    out.println("Constructor");
}

@BeforeClass
public void oneTimeSetUp() {
    out.println("Setup");
}

@AfterClass
public void oneTimeTearDown() {
    out.println("Cleanup");
}

@BeforeMethod
public void setUp() {
    deque = new ArrayDeque<Integer>();
    out.println("Method setup");
}

@AfterMethod
public void tearDown() {
    deque.clear();
    out.println("Method cleanup");
}

@Test
public void testEmptyDeque() {
    assertEquals(true, deque.isEmpty());
    out.println("empty deque");
}

@Test (expectedExceptions = NoSuchElementException.class)
public void testRemoveWhenEmpty() {
    out.println("remove when empty");
    deque.remove();
}

@Test (enabled = false)
public void testPop() {
    // pop not yet implemented
    deque.push(5);
    assertEquals(5, deque.pop());
    out.println("pop");
}

@Test (timeOut = 1000)
public void testTimeOut() {
    out.println("timeout");
    while(true);
}
```

La sortie console de l'exécution de ce cas de test [Fig.1] permet de valider certains comportements de base de TestNG. Tout d'abord, le cas de test est bien instancié une seule et unique fois. Ensuite, les méthodes de setup et cleanup globales n'ont pas besoin d'être déclarées statiques. Garantissant un contexte d'exécution stable entre

chaque test, les méthodes de setup et cleanup locales sont déclarées via les annotations @BeforeMethod et @AfterMethod. Le jeu d'annotations de configuration des tests est dans l'ensemble plus complet et plus clair avec TestNG qu'il ne l'est avec JUnit.

Ce cas de test met également en avant les innovations amenées par TestNG, reprises au sein de JUnit 4 par la suite, comme la possibilité de faire des tests devant résulter en exceptions, celle de désactiver un test via la propriété enabled de @Test et les tests asynchrones devant s'exécuter dans un certain laps de temps défini en renseignant la propriété timeOut avec une valeur en millisecondes.

ORGANISATION DES TESTS

Le framework TestNG offre aux développeurs une grande liberté pour regrouper leurs tests suivant leurs propres critères. En premier lieu, le regroupement peut s'effectuer au sein d'une suite de tests définie au sein d'un fichier de configuration XML. Au sommet de sa hiérarchie, on trouve la balise <suite> qui peut contenir plusieurs cas de tests matérialisés par la balise <test>. Cette dernière regroupant plusieurs classes de tests pointées via la balise <class>. L'utilisation d'un fichier XML de configuration permet, entre autres, de pouvoir faire intervenir la QA pour modifier la configuration d'exécution des suites de tests sans avoir à recompiler le code des tests. En outre, ce fichier permet de filtrer de manière très fine les méthodes de tests qui devront être jouées.

Ces fonctionnalités classiques de regroupements sont complétées par le concept très puissant des groupes. Un groupe sert à rassembler des tests suivant leurs fonctionnalités. Cette catégorisation s'effectue dans le code des tests via la propriété groups de @Test. En reprenant l'exemple de code précédent, on peut ainsi regrouper les méthodes de tests effectuant des opérations sur la deque au sein d'un groupe functional via @Test(groups="functional") et placer la méthode testTimeOut au sein d'un groupe example puisqu'elle ne sert qu'à illustrer l'utilisation des tests avec limite de temps. Exécuter seulement les tests fonctionnels de ce cas de test pourra ainsi être réalisé grâce au fichier de configuration testng.xml suivant :

```
<suite name="Deque Tests Suite">
    <test name="Test Case 1">
        <groups>
            <run>
                <exclude name="example" />
```

Constructor
Setup
Method setup
@Test - empty deque
Method cleanup
Method setup|
@Test - remove when empty
Method cleanup
Method setup
@Test - timeout
Method cleanup
Cleanup
PASSED: testEmptyDeque
PASSED: testRemoveWhenEmpty
FAILED: testTimeOut
org.testng.internal.thread.ThreadTimeoutException:
Method org.testng.internal.TestNGMethod.testTimeOut()
didn't finish within the time-out 1000

Fig.1

Résultat de l'exécution du cas de test



```

</run>
</groups>
<classes>
    <class name="fr.app.demo.DequeTest" />
</classes>
</test>
</suite>

```

Le résultat de l'exécution de la suite de tests montre bien que seuls les tests du groupe functional ont été joués [Fig.2].

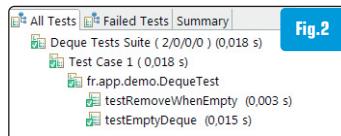


Figure 2 : Résultat de l'exécution de la suite de tests

La flexibilité introduite par ces groupes est sans limite puisqu'il est également possible de composer des groupes de groupes au sein du fichier testng.xml. Enfin, le framework propose des annotations dédiées pour exécuter certaines méthodes avant et après l'exécution d'une suite de tests (@BeforeSuite / @AfterSuite) et celle d'un groupe de tests (@BeforeGroups / @AfterGroups).

DÉPENDANCES

A contrario de la philosophie de JUnit, TestNG permet d'imposer un ordre d'exécution pour les tests. En effet, il arrive bien souvent que lorsqu'un test échoue on sache qu'un certain nombre d'autres tests vont forcément échouer à la suite. De fait, est-ce bien nécessaire d'exécuter plus d'une centaine de tests qui vont finir en échec si on le sait avant leur lancement ? La réponse est évidente et en permettant d'appliquer des stratégies d'ordonnancement des tests, TestNG offre un gain de temps significatif. Les tests sautés étant marqués en tant que tels dans le rapport de tests généré. L'ordre peut être spécifié par rapport à d'autres tests via la propriété dependsOnMethods de @Test ou bien par rapport à des groupes comme nous allons le voir dans cet exemple de tests en liaison avec un serveur d'applications :

```

public class ServerTest {
    @Test(groups="init")
    public void launchServer() { }

    @Test(dependsOnGroups="init", groups="deploy-apps")
    public void deploy() { }

    @Test(dependsOnGroups="init", groups="deploy-apps")
    public void deployAuthentication() { }

    @Test(dependsOnGroups="deploy-apps", groups="functional")
    public void testLoginOK() { }

    @Test(dependsOnGroups="deploy-apps", groups="functional")
    public void testLoginFail() { }
}

```



Ordre des dépendances pour les tests serveur

L'ordre des dépendances [Fig.3] tient bien compte du fait que les tests fonctionnels de vérification du login n'ont d'intérêt que si le déploiement a été réalisé avec succès et de même ce dernier ne pourra avoir lieu que si le serveur a bien été démarré.

DATA DRIVEN TESTING

Repris par JUnit 4, les tests paramétrés constituent une fonctionnalité essentielle de TestNG et rendent aisément le Data Driven Testing. Le premier mode d'utilisation consiste à définir les paramètres au sein du fichier testng.xml et à les appliquer sur les méthodes de tests via @Parameters(value="param"). Procéder de la sorte présente l'avantage de rendre modifiable par la QA la valeur des paramètres avant l'exécution des tests. Néanmoins, seuls les paramètres de type chaînes de caractères ou types primitifs sont supportés ici. Pour des cas nécessitant des objets complexes, il faut passer par l'annotation @DataProvider et toucher au code. Considérons la classe RomanNumeral qui possède une méthode intToRoman convertissant un nombre au format arabe en son équivalent au format romain. Au lieu d'écrire plusieurs dizaines de tests pour vérifier son comportement sur un échantillon significatif de valeurs, il est profitable de passer par un test paramétrisé qui fournit à la méthode de tests les données qu'elle doit considérer en entrée. Ceci peut être fait de la sorte :

```

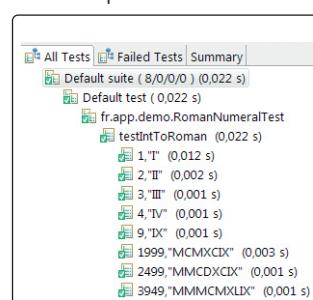
public class RomanNumeralTest {
    // ...
    @DataProvider
    public Object[][] romanNumeralData() {
        return new Object[][] {
            {1, "I"}, {2, "II"}, {3, "III"}, // ...
            {3949, "MMMCXLIX"} };
    }

    @Test(dataProvider="romanNumeralData")
    public void testIntToRoman(int arabic, String roman) {
        assertEquals(roman, romanNumeral.intToRoman(arabic));
    }
}

```

L'exécution du cas RomanNumeralTest conduit à l'appel de la méthode testIntToRoman pour chacun des échantillons fourni par la méthode annotée par @DataProvider [Fig.4]. Il est bon de préciser que pour des données provenant de sources externes, longues à charger, TestNG propose un pattern permettant de faire du chargement paresseux via l'utilisation d'un itérateur.

En outre, le fournisseur de données peut être implémenté dans une classe séparée du cas de test pour plus de modularité. En développant la méthode inverse romanToInt, il faut ajouter



Exécution du test paramétrisé

dans le cas de test une nouvelle méthode de tests liée au fournisseur de données. Cependant pour ce type de cas où de multiples méthodes doivent être appelées avec en entrée la même source de données, TestNG propose le concept



de Factory de tests. Héritant son nom du design pattern éponyme, ce concept permet de faire jouer des tests créés dynamiquement. Pour ce faire, il est nécessaire de créer un cas de test comme suit :

```
@Test
public class RomanNumTest {

    private int arabic;
    private String roman;
    private RomanNumeral romanNumeral;

    public RomanNumTest(int a, String r) {
        arabic = a;
        roman = r;
    }

    @BeforeClass
    public void setup() {
        romanNumeral = new RomanNumeral();
    }

    public void testIntToRoman() {
        assertEquals(roman, romanNumeral.intToRoman(arabic));
    }

    public void testRomanToInt() {
        assertEquals(arabic, romanNumeral.romanToInt(roman));
    }
}
```

Ce cas de test réutilisable est annoté directement au niveau de la classe par `@Test`, ce qui implique que toutes les méthodes de la classe, non configurées par ailleurs avec des annotations TestNG, seront considérées comme des tests. Le développement de la Factory se fait via l'annotation `@Factory` placé sur une méthode au sein d'une classe dédiée :

```
public class RomanNumeralTestFactory {

    private Object[][] data = new Object[][] { { 1, "I" }, ... };

    @Factory
    public Object[] createInstances() {
        Object[] result = new Object[data.length];

        for (int i = 0; i < result.length; i++) {
            result[i] = new RomanNumTest(Integer.parseInt(data[i][0].toString()), data[i][1].toString());
        }

        return result;
    }
}
```

L'exécution de la Factory permet de jouer les tests sur l'échantillon de données contenu au sein de l'objet `data`, ce qui permet d'imaginer des scénarios d'utilisation très puissants.

TESTS CONCURRENTS

Pour certaines applications, il s'avère précieux de vérifier si le code de certaines classes est bien thread-safe. Si aucun outil ne peut garantir à 100% le comportement d'une portion de code accédée simultanément par plusieurs dizaines de threads, TestNG permet néanmoins de mettre au point des tests concurrents du code. Pour ce faire, il est possible de spécifier qu'une méthode de test doit être invoquée à de multiples reprises avec un pool de threads de taille fixée et ce dans un laps de temps défini :

```
@Test(threadPoolSize=10, invocationCount=30, timeOut=10000)
public void testServer() { }
```

Le second bénéfice apporté par ce support de la concurrence concerne la rapidité d'exécution des suites de tests puisqu'il est possible de préciser, via l'attribut `parallel` de `<suite>`, que les tests devront être exécutés en parallèle avec un nombre de threads alloué pour cette exécution configurable. La granularité est assez fine puisqu'il existe un mode d'exécution parallèle par classes, par tests et par méthodes.

REPORTING

Les rapports produits par TestNG ne sont pas extraordinaires du point de vue esthétique. Le framework permet de produire des rapports de tests au format HTML, XML ou bien au format JUnitReport ce qui le rend de facto interopérable avec les différents outils prévus pour JUnit. Les données produites par ces rapports contiennent l'ensemble des tests qui ont été joués avec leur résultat, leur ordre d'exécution et leur temps d'exécution. Ils sont classés par suites et groupes en précisant les groupes éventuellement exclus lors d'une exécution. Afin de rendre extensible ces capacités de base, le créateur de TestNG a exposé une API Reporting. Celle-ci a permis à des développeurs de mettre au point des plugins de génération de rapport beaucoup plus présentables visuellement. ReportNG, le plus connu d'entre eux, présente l'avantage d'être open source tout en étant facilement intégrable aux outils de build (Ant, Maven) et d'intégration continue (Hudson, Jenkins) les plus couramment utilisés dans le monde Java.

CONCLUSION

Moins connu et moins utilisé que JUnit qui reste le standard de facto pour les tests unitaires dans le monde Java, TestNG est un produit qui gagne à être connu et qui se révèle plus puissant pour des grosses suites de tests comportant plusieurs milliers de tests. Après l'avoir essayé une fois, la plupart des développeurs ne peuvent plus s'en passer. Son intégration avec les IDE et les différents outils utilisés pour l'industrialisation des développements est excellente et ne peut constituer une raison de discrédit. En outre, il offre la compatibilité avec les tests JUnit 4 pour une migration en douceur en gardant opérationnelle une base de tests JUnit existante. Ses fonctionnalités plus avancées permettent d'envisager le test complet d'un système en partant des tests unitaires pour aller jusqu'aux tests d'acceptation. Sa portée est donc plus large que celle de JUnit et TestNG se pose donc comme une solution à choisir pour les besoins des applications de nouvelle génération.

Sylvain Saurel – Ingénieur d'Etudes Java / JEE
sylvain.saurel@gmail.com

Prenez le contrôle de votre ordinateur à partir de votre Smartphone

Cet article a pour objet d'aborder les fondements de la programmation réseau et plus spécifiquement la programmation des sockets à travers la réalisation d'un programme proposant de réaliser différentes tâches sur un ordinateur distant à partir d'une application exécutée sur un Smartphone. Nous utiliserons à cet effet le langage Java qui confère une interopérabilité certaine et jouit d'une sorte de "standardisation" pour les fonctions réseaux et leur utilisation.

LA THÉORIE AVANT LA PRATIQUE

Une socket est utilisée pour assurer la communication de données entre un serveur et un ou plusieurs clients au sein d'un réseau de type OSI, comme par exemple les réseaux TCP/IP. Pour cela, le système d'exploitation met en place une ligne directe (un flux de données) entre ces deux entités dont les extrémités correspondent schématiquement aux sockets. Pour illustrer notre propos, nous pouvons évoquer le travail des standardistes qui, autrefois, branchaient leurs fils dans les bonnes prises pour établir la communication entre deux personnes. Fidèle à la philosophie propre au monde Unix, une socket se manipule de la même manière qu'un fichier. A cet effet, chaque socket est associée à un descripteur, autrement dit une structure de données à laquelle peuvent se lier indifféremment un fichier sur disque, un terminal, une imprimante, etc. Pour manipuler les sockets, il se montrerait donc tout à fait envisageable d'exploiter les primitives « `read()` », « `write()` » et « `close()` » habituellement dévolues aux opérations sur les fichiers. Notons simplement qu'il existe des fonctions davantage spécifiques à l'utilisation des sockets et donc d'usage plus simple et plus efficace. Le mode de connexion employé par les sockets se décline en deux versions : le mode datagramme et le mode connecté.

> Le mode datagramme - « **datagram sockets** »

Il arrive parfois qu'un programme ait besoin d'envoyer une certaine quantité d'informations sans pour autant attendre un éventuel "accusé de réception". C'est ainsi le cas des programmes à usage industriel qui subissent des contraintes fortes dues aux faibles ressources mises à leur disposition. Dans la pratique, cette quantité d'informations est encapsulée à l'intérieur d'une structure appelée datagramme et à laquelle vient s'ajouter l'adresse du destinataire. La communication en mode datagramme se trouve régie par le protocole UDP (« User Datagram Protocol ») et ses applications restent relativement spécifiques (TFTP en est un exemple). En outre, les sockets en mode datagramme sont dites "sans connexion". En effet, la connexion réseau ne s'avère pas nécessaire au moment de l'envoi et dans ces conditions, ni la réception ni l'ordre de réception des paquets émis ne sont assurés: Ce mode de transmission demeure relativement contraignant et ne sera pas utilisé dans le cadre de cet article. Pour fixer clairement les idées, l'émission en mode datagramme peut se comparer à celle des lettres postales : il n'y a pas de contact direct entre "émetteur" et "récepteur", l'ordre de récep-

tion n'est pas toujours assuré, et le courrier est même susceptible de se perdre en route.

> Le mode connecté - « **stream sockets** »

Le mode connecté trouve quant à lui une certaine analogie dans le domaine de la téléphonie. Ainsi, "émetteur" et "récepteur" figurent en contact direct, l'échange est dit bidirectionnel et l'ordre des données est respecté. La communication en mode connecté est régie par le protocole TCP (« Transmission Control Protocol »), et son usage se montre beaucoup plus important qu'en mode datagramme. L'exemple le plus parlant reste sans doute le protocole HTTP (HyperText Transfer Protocol) qui donne accès aux pages HTML (HyperText Meta Language). A cet égard, vous pouvez tester directement l'usage des sockets en exécutant la commande suivante :

```
c:\> telnet www.yahoo.fr 80
```

En tapant "GET ./index.html HTTP/1.0" puis en validant deux fois avec la touche Entrée, vous verrez alors le code HTML de la page d'accueil s'afficher en retour, le travail du navigateur Internet restant alors d'interpréter ce code et d'afficher la page web correspondante.

LA PRATIQUE À PRÉSENT

Après avoir appréhendé la base théorique, nous allons à présent pratiquer l'utilisation des sockets à travers la réalisation d'un double programme client et serveur qui permettra de prendre le contrôle d'un ordinateur et d'y exécuter toute une série de commandes à partir d'un téléphone mobile connecté sur le même réseau que cet ordinateur (cas d'un réseau domestique disposant d'une BOX par exemple).

A cet effet, nous utiliserons l'environnement de développement Netbeans auquel nous ajouterons les kits de développement « **JDK Standard Edition** » et « **JDK Mobile Edition** ». Il s'agit en effet d'un environnement particulièrement bien adapté au développement d'applications pour mobile avec notamment un éditeur graphique puissant qui permet en quelques clics de définir les différents écrans de l'application ainsi que les interactions entre ces écrans.

En synthèse, nous baserons notre exemple sur l'utilisation des technologies suivantes :

- **JDK Standard Edition (SE) 1.7**
- **JDK Mobile Edition (ME)**
- **Netbeans 7.0 avec extension Java Mobile**

> Le développement du serveur

En préambule, nous attirons votre attention sur le fait que le programme présenté dans le cadre de cet article ne met pas en œuvre une gestion multithread des connexions au socket, ce qui serait indispensable si le programme devait autoriser plusieurs connexions simultanées. En effet, les méthodes associées aux sockets sont synchrones et lorsque le serveur a accepté une connexion, celui-ci ne rend la main que lorsque la connexion se termine. Aussi, au titre de la simplification, nous développerons notre serveur pour qu'il accepte une seule connexion à un instant donné.

La création de la socket

L'utilisation des méthodes d'accès aux sockets nécessite l'import du package `java.net.*`.

```
import java.net.*
```

En premier lieu, il convient de disposer d'un objet de type `ServerSocket` qui constitue la socket principale et qui nous servira tout au long du programme pour accepter les connexions clientes. Le constructeur de la classe requiert en paramètre d'entrée le port associé à la socket : il s'agit d'un numéro supérieur à 1024 ; les premiers ports étant strictement réservés par le système d'exploitation et ses applicatifs réseaux.

```
static final int port = 8082;

ServerSocket servSocket;
servSocket = new ServerSocket(port);
```

Nous pouvons à présent utiliser les différentes méthodes de la classe « `ServerSocket` ».

L'attente de connexion

L'objet étant à présent instancié, il convient de préparer la socket à toute nouvelle connexion sur le port préalablement défini. Pour cela, la classe « `ServerSocket` » dispose de la méthode « `accept()` » qui fournit en retour la socket directement associée à la connexion qui a été détectée puis acceptée par la méthode « `accept()` ».

Comme indiqué précédemment, la méthode `accept()` est synchrone et bloque le programme dans l'attente d'une connexion. Une fois cette connexion établie, le serveur n'est plus en attente d'une nouvelle connexion.

Pour permettre des connexions simultanées, il serait nécessaire de placer la gestion de la socket au sein d'un thread java qui serait exécuté à chaque nouvelle connexion.

```
Socket conn = null;
conn = servSocket.accept();
```

Nous disposons à présent de la connexion socket « `conn` » qui lie notre serveur à notre client et qui nous permettra d'assurer la communication entre ces deux entités.

La communication entre client et serveur

Pour assurer la communication entre le client et le serveur, la classe `Socket` met à la disposition du développeur les deux méthodes d'entrées/sorties suivantes :

```
conn.getInputStream() ; // pour les entrées
conn.getOutputStream() ; // pour les sorties
```

Nous allons à présent utiliser ces méthodes pour créer les objets qui nous permettront d'une part d'envoyer les données vers le client et d'autre part de recevoir les données issues de celui-ci.

```
PrintWriter out;
BufferedReader in;

out = new PrintWriter(conn.getOutputStream(), true);
in = new BufferedReader(new InputStreamReader(conn.getInputStream()));
```

Les classes « `PrintWriter` » et « `BufferedReader` » disposent des méthodes standard d'entrées/sorties telles que « `print` », « `println` », « `write` » ou encore « `read` », « `readLine` »,...

Le contrôle de l'ordinateur par le serveur

Lorsque le serveur reçoit une demande de connexion de la part d'un client sur le port prédéfini, il crée dans un premier temps un objet de type `Socket` disposant des différentes méthodes d'accès aux flux d'information en entrée et en sortie.

L'objectif de notre serveur est de recevoir de la part d'un client une ou plusieurs instructions, de les interpréter puis de les exécuter sur l'ordinateur exécutant le programme serveur.

A l'exception d'une nouvelle connexion, le serveur se met donc en attente d'une chaîne de caractères qui constitue l'instruction à exécuter.

```
out = new PrintWriter(conn.getOutputStream(), true);
in = new BufferedReader(new InputStreamReader(conn.getInputStream()));

while ((inputLine = in.readLine()) != null)
{
    if (inputLine.equals("quit"))
        break;
    else if (inputLine.equals("calc"))
    {
        Message ("Lancement de la calculatrice");
        Runtime.getRuntime().exec("calc");
    }
    else if (inputLine.equals("cmd"))
    {
        Process child = Runtime.getRuntime().exec("cmd.exe /C dir");
        InputStream inCmd = child.getInputStream();
        int c;
        while ((c = inCmd.read()) != -1)
        {
            System.out.print((char)c);
        }
        inCmd.close();
    }
    else
        Message ("Commande <" + inputLine + "> inconnue");
}
```

A ce stade, vous pouvez exécuter le serveur et tester la bonne exécution des instructions en utilisant le programme telnet qui permet la connexion sur un ordinateur distant sur un port passé en paramètre.

```
C:\>telnet localhost 8082
```

Notre serveur est prêt à recevoir les instructions du client qui, dans le cadre de notre exemple, prendra la forme d'une application Java mobile.

> Le développement du client

Pour développer l'application mobile, nous allons utiliser l'environnement de développement Netbeans qui dispose d'une interface simple et rapide d'apprehension notamment pour la conception d'application Java Mobile.

En complément du package de base Netbeans, il convient d'ajouter l'extension « Mobility » qui n'est pas disponible par défaut dans le package de base. Pour installer cette extension, rendez-vous dans le panneau de gestion des Extensions : « Tools > Plugins », cochez l'extension « Mobility » dans l'onglet « Available Plugins », puis cliquez sur le bouton « Install » proposé au bas du panneau.

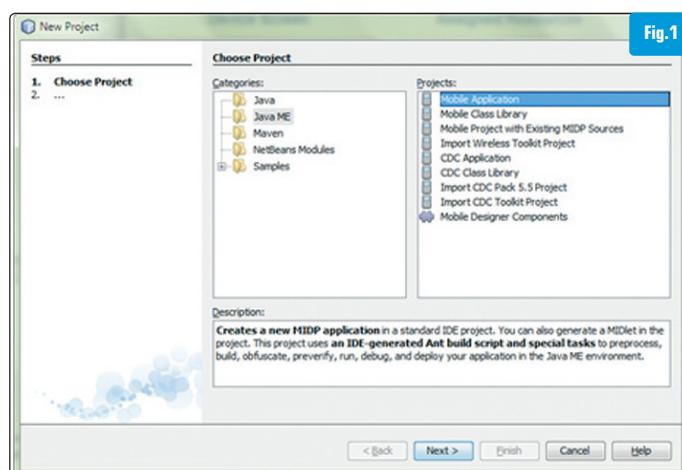
L'application mobile

Dans le cadre de notre exemple, nous créerons une application simple composée d'un unique écran qui listera les commandes à exécuter par le serveur.

A partir de Netbeans, nous allons créer un nouveau projet de type « Mobile Application » disponible dans la catégorie Java ME. Pour rappel, il aura été nécessaire d'avoir installé préalablement le kit « JDK Mobile Edition », disponible gratuitement sur internet [Fig.1].

La seconde étape consiste à présent à créer notre écran principal, à savoir l'écran listant les différentes instructions à exécuter sur le serveur. A cet effet, nous utiliserons l'éditeur graphique accessible en cliquant sur le bouton « Flow ». Il s'agit d'un outil très pratique utilisé pour définir la cinématique de l'application : nous retrouvons dans la fenêtre principale les différents écrans ainsi que les conditions de passage entre ces différents écrans [Fig.2].

Un glisser/déposer d'un écran (« displayables ») de type « List » à partir de la palette vers l'éditeur graphique permet de rajouter notre écran principal Liste. Nous pouvons, à ce stade, supprimer l'écran « Form » créé par défaut.



Création d'un projet "Application Mobile"

Notre écran Liste doit comporter une commande « ExitCommand » qui sera utilisée pour fermer l'application. Pour définir les enchaînements entre l'écran et les actions de l'utilisateur, nous créons d'une part le lien entre l'état « Started » qui correspond au démarrage de l'application et l'affichage de l'écran Liste et d'autre part le lien entre la commande « exitCommand » et l'état « Resumed » qui correspond à la fermeture de l'application [Fig.3].

A présent, notre application est composée d'un écran Liste auquel il convient d'ajouter les éléments de liste qui constitueront les différentes instructions à exécuter sur le serveur : à partir de la palette, nous déposons successivement plusieurs « List Element » vers notre écran Liste.

L'exécution d'une commande

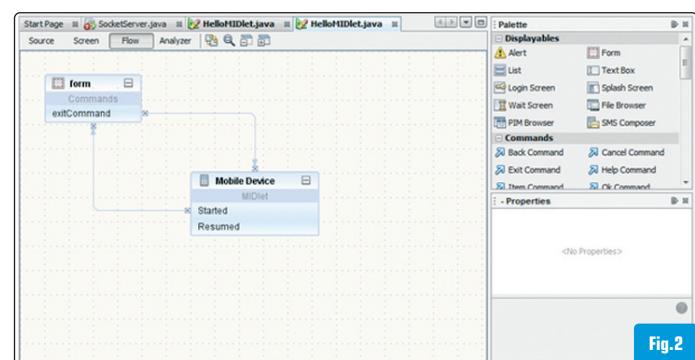
A partir de l'éditeur de flux, nous allons ajouter un écran de type « form » qui sera associé à une commande de la liste proposée sur notre écran principal Liste. Dans notre exemple, nous nommons cet écran « CheckDisk » car il sera associé à l'exécution de l'instruction « Lancer le vérificateur de disque. Une fois l'écran ajouté dans la zone d'édition, il est nécessaire de le lier avec la commande en tirant un lien entre celle-ci et le nouvel écran. Nous finalisons cet écran « CheckDisk » en ajoutant un contrôle de type « StringItem » qui sera utilisé pour afficher le résultat retourné après envoi de l'instruction vers le serveur.

L'accès au code source associé à la gestion de l'écran « CheckDisk » est accessible en cliquant sur le bouton « Source », proposé à côté du bouton « Flow » dans le ruban de l'éditeur. La méthode « Form getCheckDisk () » est exécutée au chargement de l'écran, nous allons donc développer ici la procédure de connexion et d'envoi de l'instruction vers le serveur par le biais d'une connexion de socket.

```
public Form getCheckDisk () {
...
// Connexion et execution de l'instruction
StringItem.setItemLabel(Connexion());
...
return CheckDisk;
}
```

La socket de connexion

La création d'une socket de connexion s'effectue à travers l'appel de la méthode « open() » proposée par la classe « Connector ». Cette



Éditeur de Flux (Flow) proposé par Netbeans

méthode prend en paramètre d'entrée une chaîne de connexion contenant l'adresse IP du serveur ainsi que le port de la socket et fournit en retour un objet de type « SocketConnection ».

```
SocketConnection sc;
sc = (SocketConnection) Connector.open("socket://127.0.0.1:8082");
```

L'adresse IP proposée dans cet exemple correspond à l'adresse localhost. Vous pouvez naturellement utiliser n'importe quelle machine de votre réseau pour exécuter le programme serveur et indiquer ici l'adresse IP de cette machine.

La communication avec le serveur

Une fois la connexion établie grâce à la méthode « open() », l'accès aux flux d'entrées/sorties est disponible au travers des méthodes « openInputStream() » et « openOutputStream() » de la classe « SocketConnection ».

Pour simplifier le flux de communication de notre programme, nous développons deux méthodes « recupererString () » et « envoyerString (string) » qui nécessitent comme paramètres d'entrée les objets de classes « InputStream » et « OutputStream » retournées respectivement par les méthodes « openInputStream() » et « openOutputStream() ».

```
public StringBuffer recupererString(InputStream is) {
    StringBuffer sb = new StringBuffer();
    int c = 0;
    try {
        while ((c = is.read()) != 10) {
            sb.append((char) c);
        }
    } catch (Exception e) {
        sb.append("Erreur");
    }
    return sb;
}

public void envoyerString(OutputStream os, String s) throws IOException {
    try {
```

```
        os.write(s.getBytes());
    } catch (IOException ex) {
        ex.printStackTrace();
    }
    os.flush();
}
```

L'envoi des commandes au serveur

Nous disposons à présent des écrans de l'application et des méthodes de communication avec le serveur.

Il convient donc de mettre ces méthodes en application afin de soumettre nos différentes instructions au serveur :

```
sc = (SocketConnection) Connector.open("socket://127.0.0.1:8082");
is = sc.openInputStream();
os = sc.openOutputStream();

// envoi de la commande
envoyerString(os, "chkdsk\n");
```

La méthode « envoyerString » envoie la chaîne de caractère correspondant à l'instruction à exécuter par le serveur.

Le serveur, en fonction du bon déroulement ou non de l'instruction, enverra en retour une chaîne de caractère que l'application mobile récupérera au travers de la méthode « recupererString() ».

Des exemples de commandes exécutables à distance

Les programmes présentés dans les paragraphes précédents jettent les bases d'une application de prise de contrôle à distance d'une machine par une autre machine, à l'instar des programmes largement répandus tels que VNC, pcAnywhere, « Bureau à distance », ... Voici à titre d'exemples quelques commandes pouvant être exécutées par le serveur sur demande de l'application cliente :

Commande	Description
logoff	Fermer la session Windows
shutdown	Arrêt de l'ordinateur (Windows)
chkdsk	Analyse le disque
sfc /scannow	Analyse les fichiers systèmes et corrige les erreurs détectées
tasklist	Affiche la liste des tâches exécutées sur la machine
cleanmgr	Lance le nettoyeur de disque

CONCLUSION

Nous avons vu à travers cet article les techniques permettant la communication entre deux machines présentes sur un réseau de type TCP/IP (comme internet) par le biais des sockets.

Ces techniques sont abondamment mises en œuvre dans les architectures informatiques actuelles : envoi et réception de mails (port 25 et 110), accès aux pages web (port 80), ... et seront également mises fortement à contribution avec l'avènement de la répartition multi-sites de l'information telle que promue par les architectures de type « Cloud » de plus en plus présentes...

Vincent Perdereau

vincentperdereau@gmail.com

Enchainement des écrans et commandes de notre application mobile

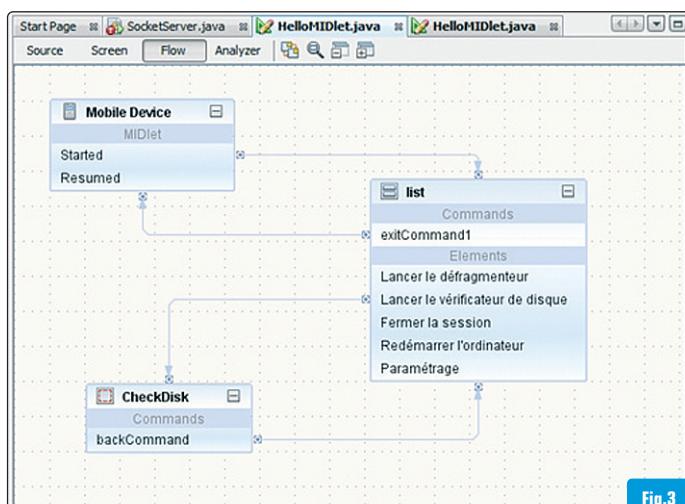


Fig.3

Personnaliser un thème sous Drupal 6

On vient de vous confier un site sous Drupal. Et votre client vous demande de modifier rapidement la structure des pages affichées. Difficile ? Pas tant que cela...

Tout webmaster s'est trouvé, au moins une fois dans sa carrière, dans la situation de reprendre un site créé par un tiers, et ceci avec des outils avec lesquels il est peu familiarisé. Le moment ne tardera pas à arriver où le client demandera une modification. Il faut pouvoir réagir rapidement. Cet article part de l'hypothèse que le site que vous prenez en charge est sous le CMS Drupal 6 et que l'on vous demande de modifier la structure des pages affichées. Nous allons voir ensemble comment ce travail, avec l'outil extrêmement puissant et complexe qu'est Drupal 6, se révèle finalement facile à mener à bien, même sans grande connaissance du système. Le tout est d'avoir une idée générale du fonctionnement de l'outil. Bien sûr, on suppose que le lecteur de cet article est habile avec PHP et qu'il a une bonne expérience du développement Web. L'apprentissage des manipulations de base avec Drupal est laissé au lecteur. Cet article est très facilement transposable à Drupal 7.

1 POSONS LE PROBLÈME

Nous partons de l'hypothèse que le site à modifier utilise le thème Riebel. Au moment de la rédaction de cet article, ce thème se trouvait justement en cours de développement. Nous l'avons donc choisi en raison de son caractère inachevé qui se prête bien à notre propos. Vous trouverez sur notre site ce thème en l'état où il était au moment de la rédaction de l'article, et vous trouverez, bien sûr le thème modifié. Notre but est de définir deux emplacements de bannières publicitaires. Une bannière horizontale en haut de chaque page, et une bannière verticale à droite de chaque page, comme illustré [Fig.1].

2 LES OUTILS

Comme nous ne travaillerons pas sur un site en production pour nous faire la main, nous supposons que le lecteur dispose, sur sa machine de travail, d'un site installé en local sous Drupal 6. Le thème par défaut sera donc ce fameux thème Riebel. On prendra la précaution de définir un autre thème comme thème de l'interface de l'administration. Le thème habituel Garland, par exemple. Même si le travail que nous voulons faire est simple, il est très pertinent d'installer des outils de développement dans le site. Ces outils sont respectivement les modules Devel et Theme Developer. Ce dernier étant lui-même au stade de développement au moment de la rédaction de cet article, recherchez-le éventuellement dans cette catégorie, sur le site drupal.org. Installez et activez ces deux modules, puis, en ce qui concerne le module Devel faites apparaître (au moins) le bloc Dévelopement. Ce bloc est magique. En un clic de souris, il permet non seulement de consulter des variables, mais aussi et surtout, de reconstruire la table de routage des menus, de réinstaller des modules et de vider le cache du système. Ces trois derniers points sont très importants. Si on les oublie, on peut s'arracher les cheveux à essayer de comprendre pourquoi une modification que l'on vient

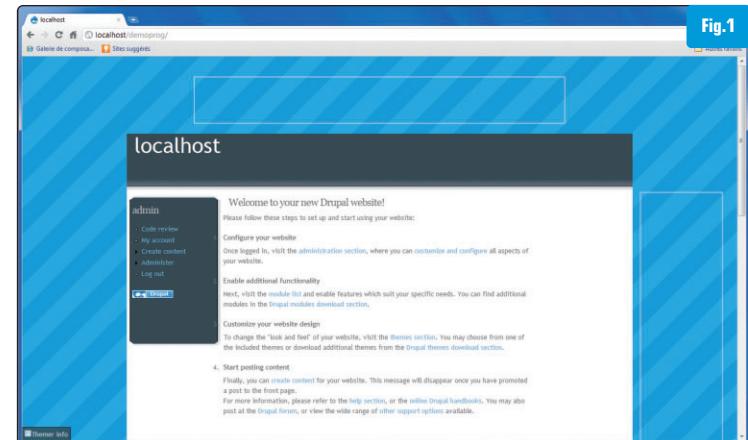


Fig.1

Nous voulons définir deux emplacements de bannières publicitaires.

d'apporter dans le code reste sans effet. Ceci en raison de certaines particularités dans le fonctionnement du cœur de Drupal. Pour les exemples de cet article, on pensera tout spécialement à vider le cache avant d'essayer les modifications. Le module Theme Developer est lui aussi magique car il permet, pour tout élément d'une page, de savoir quel modèle (template) a servi à le générer, quelles fonctions de prétraitement ont été appelées lors de la génération, et quelles variables sont connues et accessibles par le template. Un dernier mot en ce qui concerne le thème Riebel. Afin de simplifier cet article nous allons bidouiller sa feuille de style avant de commencer à travailler. Cherchez, dans le fichier style.css du thème, l'élément body et mettez quatre lignes en commentaire, comme ceci :

```
body {
font: normal 0.8em «Trebuchet MS», sans-serif;
line-height: 1.6em;
color: #666666;
width:995px;
/*
border-right-width:2px;
border-right-style:solid;
border-left-width:2px;
border-left-style:solid;
*/
```

3 APERÇU DU MÉCANISME DE GÉNÉRATION D'UNE PAGE AVEC DRUPAL

Nous donnons ici une idée générale et simplifiée, qui dans la pratique peut comporter plusieurs variations. Drupal sépare les contenus de leurs rendus. D'un côté nous avons les contenus qui sont en interne des nœuds ou nodes (avec des exceptions comme les commentaires par exemple qui ne sont pas des nœuds) et qui sont traités par des modules. De l'autre côté nous avons un moteur de template, et un thème, qui comporte des templates (ou documents modèles) et une ou plusieurs feuilles de styles. Drupal supporte trois moteurs de template, à savoir Smarty, PHPTal et PHPTemplate. En pratique PHPTemplate est utilisé presque exclusivement. Il suffit de connaître HTML et PHP pour l'utiliser, ce qui est sans aucun doute une des raisons de sa popularité. Sous Drupal chaque élément d'une page est dit 'themable'. Ceci signifie



que pour chaque élément le thème apporte un template dédié. Ainsi les contenus ou nœuds auront leur template, les commentaires auront le leur, de petits éléments comme les breadcrumb en auront un également, et les blocks aussi. Les blocks sont des petites zones rectangulaires d'une page, généralement positionnés dans les colonnes gauche et droite. Il est encore possible de définir des éléments de pages particuliers, qui auront alors eux aussi leurs templates. Les emplacements de bannières que nous voulons ajouter seront des blocks. Lorsque Drupal traite une URL, la table de routage des menus se charge d'invoquer les fonctions de modules, dont le rôle est de récupérer les données en base de données et de constituer les structures de données que le thème devra rendre. Pour le rendu, une fois les données préparées, le système appelle une fonction pivot s'il en est, la fonction theme(), et à partir de là tout le mécanisme de rendu s'enclenche. Globalement ce mécanisme appelle des fonctions de prétraitement lorsqu'il y en a, puis injecte les contenus, qui devront être les éléments rendus, dans les templates, ceux-ci les transformant en morceaux de page HTML. Finalement les morceaux sont assemblés entre eux et une page HTML complète est constituée, pour être retournée au navigateur client.

4 ANATOMIE D'UN THÈME

Un thème Drupal comporte une ribambelle de fichiers. Un ou plusieurs fichiers .css bien évidemment et des fichiers d'extensions .tpl.php qui sont les fameux templates dont nous avons parlé. Leurs noms les attribuent de facto aux éléments à rendre. Ainsi node.tpl.php sert à rendre les noeuds au sens le plus général du terme, block.tpl.php sert à rendre les blocs, etc. Au centre de tout cela, vient le fichier .info qui déclare le thème au système. Voici celui, ici généreusement amputé par votre serviteur, du thème Riebel :

```
name = Riebel
description = Online Magazine and Social Networking Theme.
core = 6.x
engine = phptemplate
screenshot = screenshot.png
regions[sidebar_left] = Left sidebar
regions[sidebar_right] = Right sidebar
regions[content] = Content
regions[footer_left] = Footer left
regions[footer_middle] = Footer middle
regions[footer_right] = Footer right
features[] = logo
features[] = name
features[] = slogan
features[] = mission
```

Ce fichier commence par décrire le thème par son nom, et par le moteur de template qu'il utilise. A la fin de l'extrait ci-dessus nous voyons énumérées un certain nombre de fonctionnalités (features). Ces fonctionnalités de base sont standard, et leur simple mention dans ce fichier fait que Drupal présentera les options de configuration correspondantes dans l'interface d'administration du thème [Fig.2].

5 LES RÉGIONS

Nous arrivons maintenant à la définition de régions, qui nous intéressent au plus haut point car ce sont des zones dont le rôle est de récep-

tionner des blocs. Ces zones sont matérialisées par des rectangles jaunes dans l'interface d'administration des blocs, [Fig.2] et bien sûr, elles n'apparaissent dans les pages du site que si elles contiennent effectivement des blocs. En fait il est plus juste de dire que ce sont les blocs qui apparaissent plutôt que les régions. Comme on le voit sur la figure, notre thème, qui est assez rudimentaire en l'état où nous l'avons trouvé, ne propose aucune région qui nous convienne. Nous allons donc tout de suite déclarer deux nouvelles régions dans le fichier .info:

```
regions[hbanner] = Banniere horizontale
regions[vbanner] = Banniere verticale
```

hbanner et vbanner sont les noms que le système utilisera, dans le travail avec les templates. *Banniere horizontale* et *Banniere verticale* seront les noms de régions affichées dans l'interface d'administration. Mais pour l'instant cette interface n'affiche rien du tout, car ici une seule déclaration ne fera pas de miracle.

6 MODIFIER DES TEMPLATES

Nous devons maintenant retoucher un template. La question est de savoir lequel. Comme nous voulons que les bannières s'affichent à chaque page, le bon sens est de les intégrer à la page entière et non à un élément susceptible de venir s'insérer dans une page. Nous regardons donc le fichier page.tpl.php. Nous découvrons avec satisfaction qu'un template PHPTemplate n'est ni plus ni moins qu'une page HTML avec du code PHP dedans. Nous ne sommes donc pas dépayrés :-) Nous devons ajouter un peu de code. Commençons par la bannière horizontale. Celle-ci devant se situer tout en haut de la page, nous ajoutons le code en début de page, comme ceci :

```
<body>
    <!-- banniere horizontale -->
    <div id="hbanner">
        <?php
            if ($hbanner) {
                print $hbanner;
            }
        ?>
    </div>
    <!-- fin banniere horizontale -->
    <!-- Header Start -->
```

Les régions, zones destinées à accueillir des blocs, sont matérialisées par des rectangles jaunes dans l'interface d'administration.

```
<div id="header">
<div id="headerInner">
```

Nous commençons par définir une boîte div. Son id est le même que celui de la région, mais ceci n'est absolument pas obligatoire. Ensuite nous testons si une variable \$vbanner est non vide. Cette fois c'est bien du nom de la région, obligatoirement, dont il s'agit. Cette variable est non vide dans deux cas. Premier cas, nous sommes dans l'interface d'administration des blocs. Alors le système matérialise les régions, comme nous l'avons vu plus haut. Second cas, la région contient des blocs. Alors les templates de blocs, comme nous l'avons dit plus haut, ont créé des morceaux de code HTML. Tous ces morceaux sont regroupés dans la variable, nous n'avons plus qu'à les afficher. La question est maintenant: comment se présentera cette région, cette div, dans le navigateur client ? La réponse est donnée tout simplement, dans la feuille de style.

Commençons par une déclaration fort pauvre pour l'instant :

```
#vbanner {
    text-align: center;
}
```

Ici le text-align ne sert qu'à positionner le nom de la région dans l'interface d'administration. Pour afficher quelque chose concrètement, nous créons un bloc sans titre, en HTML (format admin dans l'interface) avec un style incorporé (vous avez parfaitement raison, c'est mal ... :-) Mais nous ferons mieux un peu plus tard)

```
<div style=> width:728px; height:90px; border:1px solid white;
margin-bottom: 5px; margin-top: 5px; margin-left: auto; margin-right: auto;>
</div>
```

Continuons la modification de notre template, en mettant en place la seconde région. Un coup d'œil dans le débogueur de Chrome (par exemple) montre qu'il peut être intéressant de placer la région dans la div d'id maincontainer. Ce que nous faisons :

```
<!-- Main Start -->
<div id=>maincontainer<>
<div id=>vbanner<>
```

```
<?php
if ($vbanner) {
    print $vbanner;
}
?>
</div>
```

Pour ne pas compliquer cet exemple, nous choisissons un positionnement absolu dans la feuille de style :

```
#vbanner {
    position: absolute;
    margin-left: 1010px;
    margin-top: -50px
}
```

Et nous créons un nouveau bloc avec un style incorporé (et oui, c'est encore mal :-)

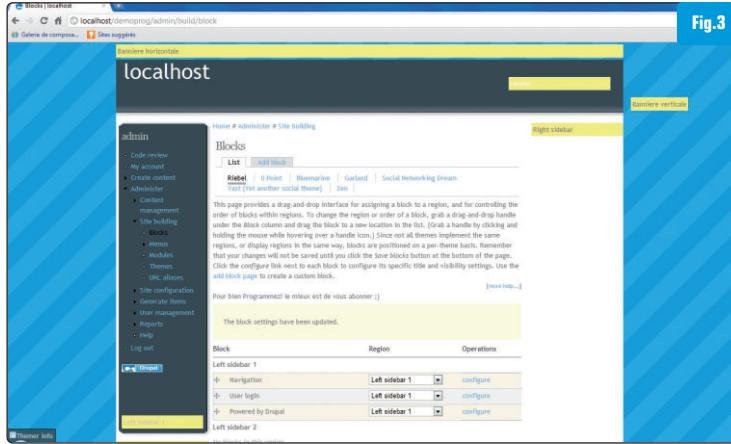
```
<div style=> height:600px; width:160px; border:1px solid white; >
</div>
```

C'est terminé nous pouvons admirer le résultat. Dans l'interface nous voyons nos deux régions matérialisées, et nous constatons aussi que la feuille de style influe directement sur cette matérialisation [Fig.3]. Placez maintenant chacun des blocs dans leurs régions respectivement et vous obtiendrez quelque chose de semblable à l'illustration [Fig.4], ce qui finalement ne correspondant pas réellement à nos attentes. La bannière horizontale présente des marges qui ne sont pas prévues.

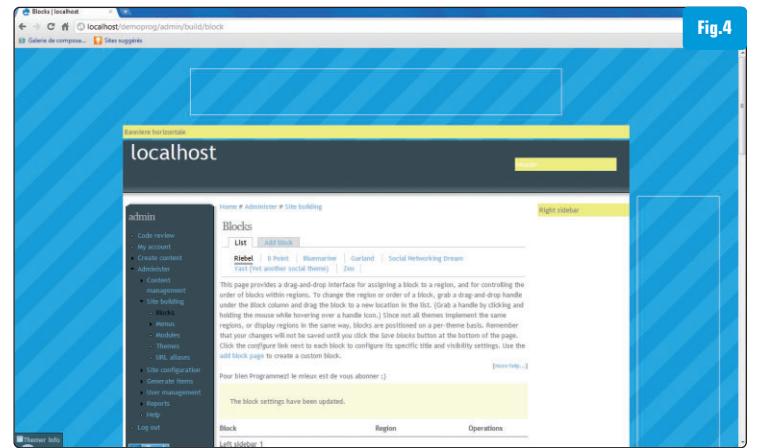
7 DÉFINIR UN NOUVEAU TEMPLATE

L'explication de ce phénomène déplaisant se trouve dans le template de blocs du thème Riebel, block.tpl.php

```
<?php // $Id:      ?>
<div class=>block block-<?php print $block->module; ?>> id=>block-<?php print $block->module; ?>-<?php print $block->delta; ?>>
<h2 class=>title<><?php print $block->subject; ?></h2>
<div class=>content<><?php print $block->content; ?></div>
</div>
```



Nos nouvelles régions apparaissent dans l'interface d'administration.



Voici qu'apparaissent nos bannières, à l'issue d'un premier jet.

Nous voyons, par exemple, que ce template affiche le titre (subject) du bloc, même quand il n'y a pas de titre, ce qui fait apparaître la marge horizontale supérieure. Nous allons donc créer un template dédié à notre bannière horizontale. Il suffit pour cela de créer un fichier en respectant une convention de nommage. En l'occurrence, nous créons le fichier block-hbanner.tpl.php dont voici le code :

```
<?php // $Id:    ?>
<div id=>hbanner-tpl><div class=>content><?php print $block->content; ?></div>
</div>
```

Pour faire les choses bien, nous allons cette fois créer un vrai style dans le fichier style.css :

```
#hbanner-tpl {
  text-align: center;
  width: 728px;
  height: 90px;
  border: 1px solid white;
  margin-bottom: 5px;
  margin-top: 5px;
  margin-left: auto;
  margin-right: auto;
}

#hbanner-tpl .content p, #hbanner-tpl p {
  text-align: center;
  color: white;
}
```

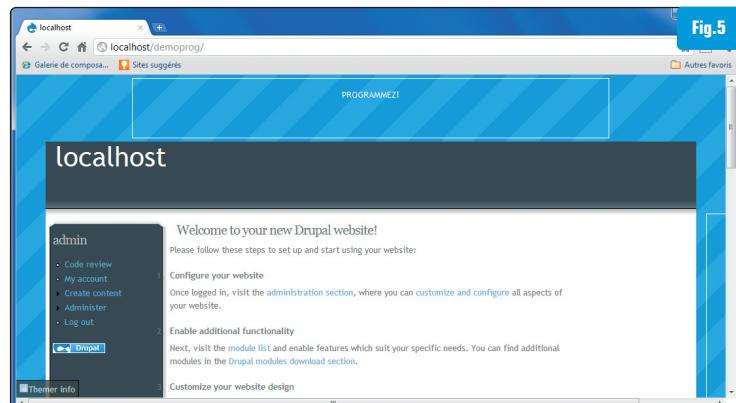
Ce qui nous permet de créer un nouveau bloc, sans HTML ce qui est beaucoup plus sain, et réduit à sa plus simple expression

PROGRAMMEZ!

Nous obtenons le résultat, satisfaisant cette fois de l'illustration [Fig.5].

8 PASSER DES VARIABLES À UN TEMPLATE

Le système passe des variables aux templates qu'il invoque. Ainsi le template block.tpl.php reçoit un objet \$block. Les variables accessibles à chaque template sont documentées à drupal.org. Elles sont généralement nécessaires et suffisantes, mais il n'est pas interdit



Avec un nouveau template, notre bannière est maintenant satisfaisante.

d'avoir des besoins supplémentaires. Avant que les templates ne soient invoqués, il y a une phase de prétraitement au cours de laquelle un jeu de fonctions aux noms conventionnels est invoqué, si elles existent. La convention de nommage et l'ordre d'invocation sont documentés à drupal.org, mais aussi dans le fichier système inclu de /theme.inc, fonction theme.

Ces fonctions reçoivent un tableau contenant des variables. Il est possible d'en modifier les valeurs ou d'en ajouter d'autres au passage. Si plusieurs fonctions touchent en cascade à une même variable, c'est la dernière fonction qui a raison. Une fois la phase de prétraitement terminée, le tableau est détruit et les variables existent individuellement en mémoire. Nous allons donc définir deux fonctions de prétraitement, déclarant chacune une variable. Le bon endroit pour placer ces fonctions est le template «générique» template.php. Nous y ajoutons donc ce code :

```
function phptemplate_preprocess(&$variables) {
  $variables['programmez'] = «PROGRAMMEZ!»;
}

function phptemplate_preprocess_block(&$variables) {
  $variables['abonnement'] = «ABONNEZ-VOUS EN BLOCK ! :-»;
}
```

Pour illustrer le fonctionnement nous créons un dernier bloc, au contenu très simple cette fois encore

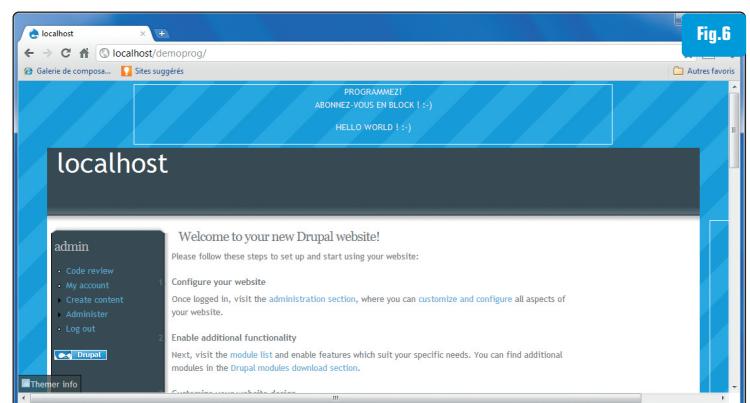
HELLO WORLD!

Puis nous modifions notre template block-hbanner.tpl.php afin qu'il exploite nos deux variables programmez et abonnement.

```
<?php // $Id:    ?>
<div id=>hbanner-tpl>
  <?php echo '<p>' . $programmez . '</p><p>' . $abonnement . '</p>' ; ?>
  <div class=>content><?php print $block->content; ?></div>
</div>
```

Le résultat des opérations est illustré [Fig.6]. Nous en avons terminé. Si Drupal est un outil complexe, il est aussi puissant et souple, ce qui nous a permis d'atteindre notre but finalement très facilement.

Frédéric Mazué - fmazue@programmez.com



Notre template reçoit maintenant des variables.

Construire une distribution de sources GNU

Dans l'univers GNU/Linux les applications binaires viennent sous deux formes. Des paquets de binaires ou des archives de sources facilement compilables. Nous examinons aujourd'hui comment construire de telles archives.

Toutes les grandes distributions Linux ont un système de package qui permet d'installer des applications toutes prêtes à fonctionner. Les packages peuvent être au format RMP (Red Hat et distributions dérivées), au format DEB (Debian et distributions dérivées, etc.). Ces systèmes de package ont des avantages indéniables, à commencer par les facilités qu'ils présentent à l'utilisateur final. Mais ils ont aussi leurs inconvénients. Par exemple, il arrive fréquemment que les dépendances pour un paquet particulier soient excessivement lourdes. Ils présentent encore l'inconvénient de fournir des binaires compilés pour le plus grand nombre de machines. Par exemple pour i386, mais vous, vous voudriez bien tirer parti de toute la puissance du CPU de votre nouvelle machine :-) Ils manquent encore de souplesse. Avec eux, vous ne pouvez pas avoir une application dotée seulement de certaines fonctionnalités et pas d'autres. Ils présentent encore le défaut de manquer d'universalité. Par exemple, les RMP ne s'installent pas sur toutes les distributions.

1 L'UNIVERSALITÉ DES AUTOTOOLS GNU

En alternative aux systèmes de packages, viennent les archives de sources GNU que tout le monde a rencontré au moins une fois dans l'univers de Linux/UNIX. C'est très simple à mettre en œuvre. On décomprime l'archive de sources, on tape dans un terminal la mythique séquence de commandes :

```
. /configure  
make  
make install
```

Et c'est réglé :-) Ce système présente un avantage énorme: l'universalité. Vous pourrez vous en servir sur n'importe quel Linux, même le plus exotique, et au-delà de cela, sur n'importe quel UNIX qui dispose des outils GNU, c'est-à-dire sur tous les UNIX :-) Ces outils, incroyablement puissants et généralement appelés de façon simplifiée les 'Autotools' présentent un inconvénient de taille, la difficulté. Ils sont très difficiles à apprendre et à mettre en œuvre. Fonctionnement complexe, syntaxe lourde, manquant d'homogénéité, absconse, etc. Cependant, si des outils de build alternatifs remarquables existent, comme Scons ou CMAKE, ce dernier étant pourtant l'outil de build de KDE, ils ne parviennent pas à s'imposer. Les habitudes de l'univers d'UNIX sont coriaces. Alors on continue à utiliser les autotools. Mais franchement, quelle satisfaction lorsqu'on a réussi à les mettre en œuvre et qu'on publie ainsi ses sources comme un pro :-) Nous allons donc voir comment résoudre les difficultés présentées par les autotools dans les cas de figure classiques. Nous supposons que le lecteur possède d'assez bonnes connaissances en administration Linux/UNIX et en Shell, qu'il sait compiler du code C/C++, et qu'il sait manipuler des makefiles de base. Les autotools doivent bien entendu être présents sur votre système. Installez-les si besoin depuis votre gestionnaire de packages.

2 PRÉSENTATION DES AUTOTOOLS

Il s'agit d'un ensemble de scripts frontaux (écrits en Shell, et Perl... :) qui lisent des fichiers déclaratifs et génèrent les scripts nécessaires à la configuration et la compilation du projet, les scripts configure et les Makefile notamment. Voici succinctement qui fait quoi :

Script	Action
Aclocal	Dépose dans l'archive de sources les macros m4 qui seront exécutées lors de l'exécution du script configure.
Autoheader	Génère les fichiers en-tête de configuration qui seront lus lors de la phase de compilation.
Automake	Génère les makefiles
Autoconf	Génère le script configure
Libtools	Aide à la construction de bibliothèques portables.

Il y a deux façons d'envisager l'utilisation des autotools. La première est de créer son application et la compiler en écrivant les makefiles manuellement au fur à mesure, puis passer aux autotools à postériori pour créer une distribution. La seconde, plus classieuse, consiste à utiliser les autotools conjointement à la construction de l'application, et même s'appuyer sur eux pour tester divers cas de configuration. Nous appliquons ici la seconde démarche.

3 POUR SE FAIRE LA MAIN

Nous allons donc créer une archive GNU d'une très belle application de type helloworld. En voici le code qui n'a pas encore été compilé.

```
#include <config.h>  
#include <iostream>  
  
using namespace std;  
  
int main(int argc, char* argv[]){  
    cout << "Hello World! ";  
    cout << "du package " << PACKAGE_STRING << endl;  
    cout << "le fichier de configuration de cette application est: "  
        << CONF_FILE << endl;  
    return 0;  
}
```

En l'état, ce code ne peut d'ailleurs pas être compilé. Le fichier en-tête config.h est inconnu. Les macros PACKAGE_STRING et CONF_FILE sont inconnues également. Nous construisons donc notre distribution conjointement à notre application, ce qui va automatiquement combler les manques. Vous trouverez tous nos exemples sur notre site. Ce premier exemple réside dans le répertoire simplehelloworld. Une distribution GNU propre sur elle obéit à des conventions. Ainsi on s'attend à trouver certains fichiers. Donc dans notre répertoire simplehelloworld nous faisons :

```
touch NEWS README AUTHORS ChangeLog
```

et l'on songera à ne pas laisser ces fichiers désespérément vides...
:-) On s'attend également à trouver certains sous-répertoires:

```
mkdir doc po src
```

Et bien entendu, le code de notre application simplehelloworld.cpp sera déposé dans le sous-répertoire src. Le point d'entrée de la construction d'une distribution GNU se trouve dans un fichier déclaratif conventionnellement nommé configure.ac. Il est pertinent d'éditer un tel fichier avec un outil qui connaisse la syntaxe des macros m4, Emacs ou Gedit par exemple. Voici notre configure.ac

```
AC_INIT([simplehelloworld], [1.0], [fmazue@programmez.com])
AC_PREREQ([2.59])
AM_INIT_AUTOMAKE([simplehelloworld], [1.0], [foreign -Wall -Werror])
AC_PROG_CXX
AC_CONFIG_HEADERS([config.h])
AH_TEMPLATE([CONF_FILE], [Le fichier de configuration de simplehelloworld.])
AC_DEFINE([CONF_FILE], [/etc/simplehelloworld.conf"])
AC_CONFIG_FILES([Makefile src/Makefile])
AC_OUTPUT
```

Ce fichier énumère un certain nombre de macros m4, dans un ordre qui est globalement indifférent, quoique pas toujours :-) Ainsi l'on doit commencer par AC_INIT qui définit le nom de l'application et sa version. La macro suivante, AC_PREREQ, va s'assurer que les autotools présents sur le système cible sont dans une version suffisante, ici 2.59. Vient ensuite AM_INIT_AUTOMAKE qui initialise le frontal automake. On se doute que cette macro doit toujours être là :-) Arrêtons-nous un instant pour faire une remarque importante: les macros sont préfixées selon l'outil qu'elles concernent. Ainsi AC concerne autoconf, AH autoheader et AM automake. Il est très vite arrivé, lorsqu'on écrit un fichier configure.ac de se tromper de préfixe, pour après ne plus comprendre ce qui se passe. Il est donc pertinent de toujours être bien attentif à ce point. AC_PROG_CXX fait que le script configure recherchera la présence du compilateur C++ sur le système cible. Pour un compilateur C on aurait mis AC_PROG_CC. AC_CONFIG_HEADERS définit l'en-tête de configuration. Ici notre fichier config.h pour l'instant manquant, mais qui sera maintenant automatiquement générée et remplie. On remarque que les macros ont une syntaxe de type liste. On pourrait ainsi demander plusieurs en-têtes :

```
AC_CONFIG_HEADERS([config.h monautreconfig.h encoreunconfig.h])
```

Ensuite AH_TEMPLATE et AC_DEFINE permettent la déclaration d'une macro, au sens C/C++ du terme, dans config.h. Ici, nous donnons l'emplacement et le nom d'un hypothétique fichier de configuration. Tout code source de notre application, pourvu qu'il inclue config.h, pourra donc accéder à cette macro. Enfin AC_CONFIG_FILES, comme son nom ne l'indique pas, énumère les makefiles, inexistant à ce stade, qui seront nécessaires à la compilation de l'application. AC_OUTPUT quant à elle provoque l'écriture des fichiers que configure.ac définit. Ces fichiers peuvent être définitifs, comme le script configure, ou intermédiaires comme les fichiers makefile.in ou aclocal.m4 :-) On le voit, tout cela est plutôt complexe et hétérogène. On peut se demander avec inquiétude où se trouve la documentation. Elle se trouve à <http://www.gnu.org/s/hello/manual/autoconf> et <http://www.gnu.org/s/hello/manual/automake>. Et ces macros m4, où sont-elles et que sont-elles finalement ? Il s'agit de

scripts que vous trouverez sur votre machine dans les répertoires /usr/share/localxxx. Elles méritent vraiment un coup d'œil. Nous avons presque fini avec notre premier exemple. Autant nous avons donné de fichiers à AC_CONFIG_FILES, autant nous devons écrire de fichiers Makefile.am (attention à la casse) qui contiennent des déclarations pour générer les makefiles. A la racine de notre distribution, notre premier Makefile.am est tout simple :

```
$ SUBDIRS = src
```

Nous comprenons qu'il va renvoyer à src/Makefile.am dont voici le contenu :

```
simplehelloworld_CPPFLAGS = -O3 -Wall
bin_PROGRAMS = simplehelloworld
simplehelloworld_SOURCES = simplehelloworld.cpp
```

Nous voyons ici que nous redéfinissons les options de compilation pour tous les sources présents dans ce répertoire. Nous donnons -O3 -Wall en remplacement des options par défaut -O2 -g. Nous définissons le nom du binaire qui sera généré, et nous énumérons tous les sources à compiler, séparés par un espace. A ce point, nous pouvons apprécier le fait que si les autotools sont complexes, ils nous dispensent ici de l'écriture manuelle des makefiles, qui, elle aussi, est souvent complexe, et toujours fastidieuse.

4 GÉNÉRER LA DISTRIBUTION

Tout étant prêt, nous pouvons lancer nos autotools, dans un terminal comme ceci : [Fig.1].

```
aclocal
autoheader
automake -a -f
autoconf
```

Dans cette séquence aclocal doit venir en premier. Son rôle est de copier les macros m4 nécessaires à la construction du projet dans le fichier aclocal.m4. L'ordre des commandes suivantes est sans importance. Les commutateurs donnés à automake font que celui-ci créera des fichiers ou répertoires manquants à votre distribution éventuellement. Dès que les outils auront fini leur travail, vous constaterez que le répertoire de votre distribution s'est considérablement enrichi [Fig.2]. L'instant magique tant attendu est maintenant arrivé. Nous pouvons lancer le script configure de façon à ce que

```
fred@mars: ~/Développement/demos-autotools/simplehelloworld
Fichier Édition Affichage Rechercher Terminal Aide
fred@mars:~/Développement/demos-autotools/simplehelloworld$ aclocal
fred@mars:~/Développement/demos-autotools/simplehelloworld$ autoheader
fred@mars:~/Développement/demos-autotools/simplehelloworld$ automake -a -f
configure.ac: installing './install-sh'
configure.ac: installing './mkinstalldirs'
configure.ac: installing './missing'
Makefile.am: installing './COPYING'
Makefile.am: installing './INSTALL'
src/Makefile.am: installing './depcomp'
fred@mars:~/Développement/demos-autotools/simplehelloworld$ autoconf
fred@mars:~/Développement/demos-autotools/simplehelloworld$
```

Fig.1

C'est parti ! Enfin nous lançons les autotools

notre application, une fois construite, soit installée sous un répertoire utilisateur pour l'essayer [Fig.3].

```
./configure --prefix=/home/fred
make
make install
```

Et, sachant que /home/fred./bin est dans notre PATH, nous constatons avec la plus grande satisfaction que cette application fonctionne comme prévu :-)

```
simplehelloworld
Hello World! du package simplehelloworld 1.0
le fichier de configuration de cette application est: /etc/simple
helloworld.conf
```

Il ne reste plus qu'à créer l'archive de votre distribution avec

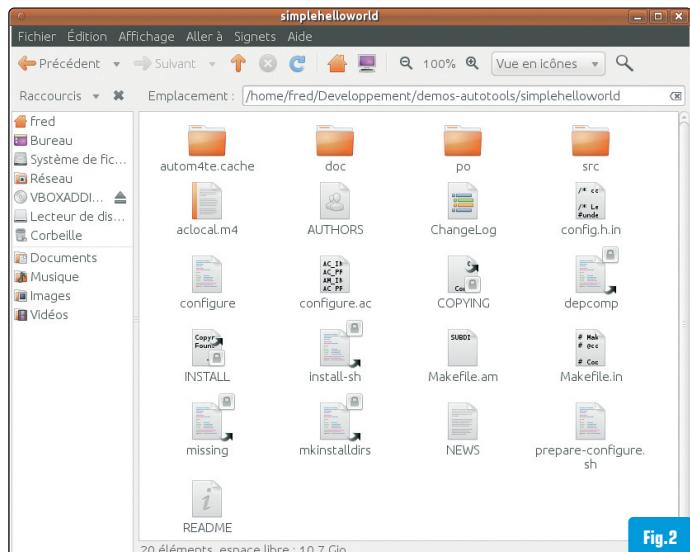
```
make dist
```

qui va créer un fichier simplehelloworld-1.0.tar.gz à la racine de votre distribution. Vous pouvez désormais partager avec le monde entier votre magnifique application :-)

5 UTILISER DES LIBRAIRIES TIERCES

Dans le monde de Linux, il est rarissime qu'une application ne soit pas liée avec des librairies tierces. Un script configure doit alors être capable de détecter la présence de ces librairies sur le système cible, et de faire échouer la construction si ce n'est pas le cas. D'autre part, les makefiles doivent être capables de réaliser l'édition de liens avec lesdites librairies. Examinons notre exemple hellogtk. Son répertoire src contient une applicationGtk+ qui n'est rien d'autre que le helloworld du tutoriel du site officiel. Le fichier config.ac est très intéressant :

```
AC_INIT([hellogtk], [1.0], [fmazue@programmez.com])
AC_PREREQ([2.59])
AM_INIT_AUTOMAKE([hellogtk], [1.0], [foreign -Wall -Werror])
AC_PROG_CC
AC_CONFIG_HEADERS([config.h])
AC_CONFIG_FILES([Makefile src/Makefile])
```



Tous les éléments constitutifs de notre distribution sont maintenant en place.

```
dnl tester la presence de glib-2.0 de cette facon
AC_CHECK_LIB([glib-2.0], [g_malloc],
[AC_MSG_ERROR(!!!! ATTENTION!!! la librairie glib2.0 est absente de votre système)])
dnl ou de cette facon
AM_PATH_GLIB_2_0([2.0.0],,, [AC_MSG_ERROR(!!!! ATTENTION!!! la librairie glib2.0 est absente de votre système)])
AM_PATH_GTK_2_0([2.0.0],,, [AC_MSG_ERROR(!!!! ATTENTION!!! la librairie glib2.0 est absente de votre système)])
AC_OUTPUT
```

Les applications Gtk+ nécessitent elles-mêmes que la librairie glib soit présente sur le système. Nous testons cette présence avec la macro AC_CHECK_LIB qui peut être utilisée pour n'importe quelle librairie. Nous réalisons le test en vérifiant que la fonction g_malloc de glib est disponible. Toutefois, dans le cas de très grosses application/librairies populaires, il arrive que celles-ci soient configurées de manière particulière et qu'une macro m4 "de base" échoue pour leur détection. Cela peut arriver avec glib, cela va se produire à coup sûr avec Gtk+. Dans de tels cas, les distributions de ces librairies/applications fournissent les macros m4 qui vont bien. Allez visiter /usr/share/aclocal sur votre Linux. Vous y trouverez à coup sûr des fichiers comme glib-2.0.m4 et gtk-2.0.m4. Examinez le contenu de ces fichiers. A leur début est donné le nom des macros qu'ils définissent, et éventuellement les variables qu'à leur tour les macros vont définir. glib-2.0.m4 et gtk-2.0.m4 définissent respectivement les macros AM_PATH_GLIB_2_0 et AM_PATH_GTK_2_0 que nous avons utilisées dans notre fichier config.ac. Il nous reste à assurer que la compilation aboutisse, c'est-à-dire que les en-têtes Gtk+ soient accessibles au compilateur, et que l'édition de liens soit faite correctement. Grâce aux variables que la macro AM_PATH_GTK_2_0 définit, à savoir GTK_CFLAGS et GTK_LIBS, ceci se fait très simplement dans src/Makefile.am :

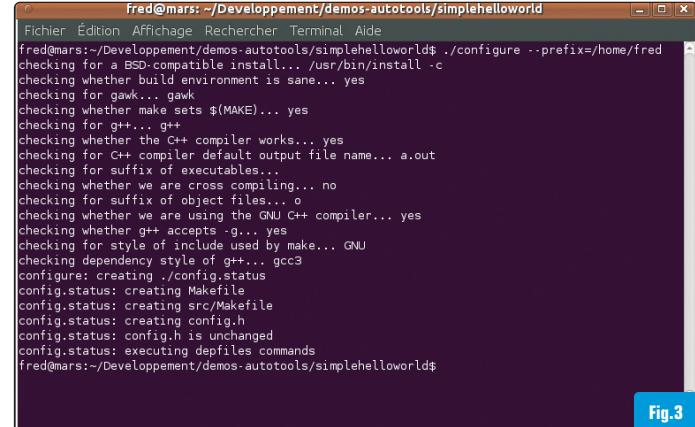
```
hellogtk_CFLAGS = -O3 -Wall $(GTK_CFLAGS)
hellogtk_LDFLAGS = $(LDFLAGS) $(GTK_LIBS)
bin_PROGRAMS = hellogtk
hellogtk_SOURCES = hellogtk.c
```

6 COMPILATIONS OPTIONNELLES

Si vous avez déjà donné la commande :

```
./configure --help dans un terminal
```

Vous savez que, très souvent, une application peut être construite



L'instant magique :-) Nous lançons le script configure que nous avons généré.

avec ou sans tout un jeu de fonctionnalités. Pour notre exemple suivant, hellopthread, nous définissons une application qui peut être monothread ou multithread. En voici le code :

```
#include <config.h>
#include <iostream>

using namespace std;

int main(int argc, char* argv[])
{
#ifdef HAVE_LIBRARY_PTHREAD
    cout << "Cette application est multithread" << endl;
#else
    cout << "Cette application est monothread" << endl;
#endif
    return 0;
}
```

L'utilisateur peut ou non décider que cette application soit multithread. Pour cela il faut d'abord qu'il soit informé de la possibilité. C'est pourquoi il faut que configne --help affiche quelque chose comme :

```
Optional Packages:
--with-PACKAGE[=ARG]      use PACKAGE [ARG=yes]
--without-PACKAGE         do not use PACKAGE (same as --with-PACKAGE=no)
--with-pthread              genere une application multithread [default=no]
```

Ainsi que nous le voyons, l'application sera monothread par défaut. Mais si l'utilisateur donne ./configne --with-pthread, alors il faut que le processus de build détecte la présence de la librairie pthread, qu'une macro, au sens C/C++ du terme soit définie dans config.h et que l'édition de liens soit faite avec la librairie pthread. Voici le script configne.ac qui assure tout cela :

```
AC_INIT([hellopthread], [1.0], [fmazue@programmez.com])
AC_PREREQ([2.59])
AM_INIT_AUTOMAKE([hellopthread], [1.0], [foreign -Wall -Werror])
AC_PROG_CXX
AC_CONFIG_HEADERS([config.h])
AC_ARG_WITH([pthread],
[AS_HELP_STRING([--with-pthread],
[genere une application multithread @<:@default=no@:>@]),
[with(pthread)=check],
[with(pthread)=no]])

LIBPTHREAD=
AS_IF([test "x$with(pthread" != xno],
[AC_CHECK_LIB([pthread], [pthread_mutex_init],
[AC_SUBST([LIBPTHREAD], ["-lpthread"])
AC_DEFINE([HAVE_LIBRARY_PTHREAD], [1],
[Macro signalant l'emploi de pthread])]],
[if test "x$with(pthread" != xcheck; then
    AC_MSG_FAILURE(
        [--with-pthread demandé, mais le test for pthread a échoué])
    Fi
)]])
AC_CONFIG_FILES([Makefile src/Makefile])
AC_OUTPUT
```

Ce code est assez ardu, il faut bien le reconnaître. Mais il vous sera facile d'adapter cet exemple à vos besoins particuliers. Le résultat du travail de ce script est exploité dans src/Makefile.am :

```
hellopthread_CPPFLAGS = -O3 -Wall
hellopthread_LDFLAGS = $(LDFLAGS) $(LIBPTHREAD)
bin_PROGRAMS = hellopthread
hellopthread_SOURCES = hellopthread.cpp
```

7 CONSTRUIRE UNE APPLICATION COMPLEXE

C'est-à-dire une application constituée d'un exécutable et d'une (au moins) librairie. Pour notre dernier exemple hellobig, libtools doit être installé sur le système. Libtools permet, entre autres, de travailler avec un format de librairies .la indépendant des plateformes cibles. Notre exemple contient les sources de l'application dans src, et ceux la librairie dans lib. Notre distribution doit comporter deux sous-répertoires supplémentaires: build-aux et m4. Notre fichier configne.ac est plutôt simple :

```
AC_INIT([hellobig], [1.0], [fmazue@programmez.com])
AC_PREREQ([2.59])
AC_CONFIG_AUX_DIR([build-aux])
AC_CONFIG_MACRO_DIR([m4])
AM_INIT_AUTOMAKE([hellobig], [1.0], [foreign -Wall -Werror])
LT_INIT
dnl Pour ne construire que des librairies partagées
dnl utiliser la ligne ci-dessous
dnl LT_INIT(disable-static)
AC_PROG_CXX
AC_CONFIG_HEADERS([config.h])
AC_CONFIG_FILES([Makefile lib/Makefile src/Makefile])
AC_OUTPUT
```

Voici les patrons de makefile :

```
Makefile.am:
SUBDIRS = lib src
ACLOCAL_AMFLAGS = -I m4

src/Makefile.am:
AM_CPPFLAGS = -I$(srcdir)/..lib
bin_PROGRAMS = hellobig
hellobig_SOURCES = hellobig.cpp
hellobig_LDADD = ../lib/libmalibrairie.la

et enfin lib/Makefile.am :
lib_LT_LIBRARIES = libmalibrairie.lib
libmalibrairie_la_SOURCES = malibrairie.cpp malibrairie.h
libmalibrairie_la_LDFLAGS = -version-info 1:0:0
```

Ne lancez pas cette fois les autotools un par un, mais donnez plutôt dans votre terminal :

```
autoreconf --install
```

ceci provoquera l'exécution des autotools bien sûr, mais aussi de l'outil libtoolize. Et voilà! Vous avez généré une vraie distribution GNU de pros :-)

```
# Frédéric Mazué - fmazue@programmez.com
```

Concevoir un design d'interaction sur Surface 2.0

C'est en 2008 que Microsoft met à disposition sur le marché une table tactile particulière, tournée résolument vers le collaboratif et bénéficiant de reconnaissances inédites. Ce produit innovant est aussi très encombrant et au jour d'aujourd'hui un peu obsolète technologiquement.



C'est donc fin Janvier 2012 que la nouvelle version de ce périphérique tactile sera disponible sur le marché. Plus fine, plus moderne avec un design réussi, la table Surface2 (appelé aussi SUR40) va indéniablement marquer son temps. Produite par Samsung, c'est un grand partenariat technologique (grâce au PixelSense) qui commence pour ce produit entre les deux géants.

> Principes de Surface

Ce support est comme sa version précédente, un outil collaboratif et multi-utilisateur ayant pour but de créer des expériences inoubliables et marquantes, on parle d'ailleurs de « l'expérience Surface ». Les interfaces des applications proposées doivent être avant tout intuitives, (interface NUI pour « Natural User Interface »). On retrouve la table Surface dans différents types de lieux et pour différents usages. Imaginons qu'un client, directeur d'une agence de voyage moyen de gamme, demande une application photos/vidéos pour la placer à l'accueil de son agence. Cette application devra être conçue pour cet usage en prenant en compte plusieurs paramètres. Il ne faut pas oublier qu'une application peut être intuitive pour certaines personnes mais ne pas l'être pour d'autres. Il est presque impossible de concevoir une interface intuitive pour tout le monde. Une conception centrée sur les utilisateurs finaux est essentielle pour obtenir un résultat simple d'utilisation. Pour cela et avant même de commencer la conception

(design) de l'application, le designer devra savoir qui est l'utilisateur, où l'application va être exécutée, quel temps moyen d'usage, va-t-il être amené à se servir de l'application seul ou avec une personne ayant la connaissance de l'outil. Pour cette première phase de l'application, il est intéressant de faire intervenir un designer d'interaction. Je suis pour ma part designer industriel et ma formation m'amène à penser avant tout pour des Hommes. Qu'un designer conçoive un objet, un packaging, une voiture ou une interface tactile, la réflexion part toujours de l'utilisateur et de l'utilisation. Une application Surface doit toujours être conçue en prenant en compte l'environnement d'utilisation car le niveau d'intuitivité est lié à ces paramètres. Dans notre cas, nos utilisateurs finaux sont les clients d'une agence préparant un voyage personnel ou professionnel. L'application sera installée à l'accueil, le temps d'utilisation moyen est d'environ 15 minutes selon le client et il sera seul ou accompagné par d'autres clients. Personne n'ayant la connaissance de l'outil, ne l'aidera lors de sa prise en main. Il ne faut pas perdre de vue que cela peut même être la première fois que notre utilisateur touche à un « écran tactile ».

C'est dans cette optique de création d'applications adaptées et centrées utilisateurs que l'équipe Surface a créé Metro.

Metro est le principe d'interface unifiée pour toutes les interfaces des produits Microsoft (Xbox, Windows Phone, Windows 8, Surface 2, etc.).

> Principes Metro

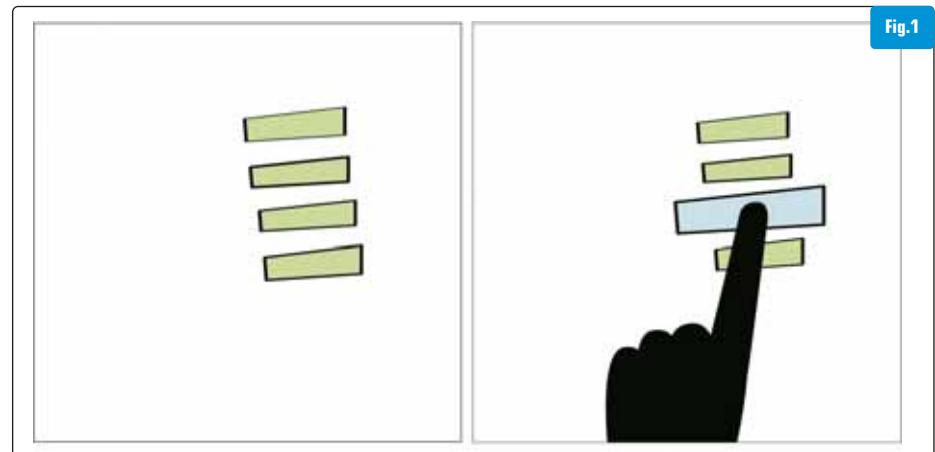
Metro n'est pas qu'une façon de mettre en page des éléments de texte dans une application mobile, il se base sur 4 grandes « inspirations » issues du 20^e siècle que les concepteurs designers et graphistes vont s'approprier pour adapter Metro au besoin du projet et des clients.

La 1^{re} inspiration est le Bauhaus, école de design industriel et architecture de 1919. Simplicité des formes, sobriété, pas de fioritures sont les principes de cette école. C'est aussi la première fois dans l'histoire que des études ergonomiques approfondies sont effectuées pour du mobilier et où l'on essaie de répondre en cohérence avec des usages précis. L'idée à retenir ici est que l'esthétisme vient après les considérations fonctionnelles. Dans notre application de photo pour agence de voyage nous veillerons donc à mettre en avant les fonctionnalités de l'application comme le fait de pouvoir naviguer dans des albums, lire des vidéos, s'envoyer les photos par mail. La 2e inspiration est le style Suisse, c'est une tendance de graphisme des années 1950 qui se base sur des principes de lisibilité, dépouillement et objectivité dans la mise en page. Ce style devient rapidement le « style international ». En ce qui concerne notre application, il faudra faire attention à ce que le texte présent dans l'application soit vraiment utile, que les graphismes présents apportent un plus indiscutable à celle-ci. Metro ne veut pas dire « ne pas mettre d'illustration et de graphisme », pour notre application de voyage il

est indispensable que le client se sente déjà en partance lors de la navigation de l'application. On pourra donc traiter le fond de l'application avec des photos, des formes et des sons évoquant le voyage. La 3^e inspiration est le « motion design », il s'agit de la mise en animation de contenu texte et graphique. Saul Bass est un des artistes dont l'équipe ayant conçu Metro s'est inspiré. Il est important de comprendre que les animations ne sont placées que lorsqu'elles sont utiles. Elles servent de retour visuel pour guider l'utilisateur dans sa navigation. Dans notre application, on pourra placer ce genre d'animation lors de l'ouverture d'un album photo et ainsi associer un son à un effet visuel (Exemple de retour visuel [Fig.1]).

La signalétique urbaine internationale est la dernière grande inspiration. Metro étant amené à être une interface utilisée sur tous les supports, il devait être compris par tout type de population de tout pays, c'est pour cela que l'utilisation des codes visuels de la signalétique urbaine (Métro, aéroports, gare...) est pertinente. Cette signalétique a été mise en place dans les années 80 avec des codes communs sur tous les continents et cela pour suivre l'intensification du phénomène de mondialisation. Dans notre application, les codes visuels compréhensibles sont moins complexes à établir car il s'agit d'une application française, pour une cible restreinte de personnes. Il faut cependant se poser la question des langues parlées par les clients pour que si nécessaire, les textes de l'interface soient adaptés.

D'après le *Microsoft Surface 2.0 Design and Interaction Guide* les interfaces de la table devront suivre quelques grands principes : Le premier principe est la simplicité des visuels tout en respectant la charte graphique et l'univers de la marque. Le deuxiè-



me principe est l'organisation hiérarchique des éléments de l'interface selon leur importance à un instant T. Et le troisième est l'utilisation de références au monde réel seulement lorsque c'est nécessaire. Dans le cas de notre application, nous utiliserons la charte graphique de l'agence de voyage. Les éléments principaux mis en avant seront les catégories des albums et en second plan, le contenu média (photo et vidéos). Le contenu est prioritaire, les menus et actions sont secondaires. L'important, ce sont les photos et leur manipulation (zoomer, pivoter, déplacer) il est par exemple secondaire de pouvoir, dupliquer la photo au travers d'un menu. Le dernier principe mis en avant par ce guide est le fait d'amener le contenu « à la vie » grâce aux transitions et aux retours visuels afin de guider l'utilisateur, même principe que Metro avec le « Motion Design »

> La table dans les grandes lignes

Le multi-utilisateur est une évidence. Il supporte jusqu'à 50 contacts simultanés. Un scénario comme le croquis ci-dessous [Fig.2] est donc à bannir, cela entraînerait une mauvaise expérience utilisateurs pour la plupart d'entre eux.

Une application Surface doit être au moins multi-utilisateurs, au plus collaboratif. Créer une application mono-utilisateur sur un support multi-utilisateurs n'a pas de sens. Chacun doit pouvoir se servir de l'application sans gêner les autres. Dans notre application, cela se traduit par la possibilité que plusieurs clients puissent manipuler les photos, naviguer dans les albums simultanément.

Comme pour la V1, la V2 peut voir ce qui la touche, mais cette fois-ci

c'est grâce à la technologie du PixelSense, c'est-à-dire que sous chaque pixel de l'écran se trouve une micro caméra infrarouge, chacune étant reliée aux autres comme une matrice, cela permet d'avoir une image instantanée de la Surface. En plus de reconnaître un doigt d'un autre élément et son orientation, la table sait lire les tags qu'on va lui poser dessus, c'est principalement grâce à ces éléments que la table Surface se différencie des autres supports tactiles du marché. En effet, les dalles tactiles à infrarouge, souvent moins chères et pouvant aller au-delà de 55" ne permettent pas cela et les écrans capacitifs sont très peu fiables au-delà de 24". La table est équipée de Bluetooth et Wifi, on retrouve aussi un port HDMI un lecteur de carte SD et des ports USB. Fonctionnalités que nous pourrions utiliser dans notre application pour transférer de nouvelles photos dans les albums. Cette nouvelle version de la table permet diverses orientations qui vont amener différents usages avec chacun leurs avantages et inconvénients.

Une position horizontale permettra un usage long car le poignet sera bien droit, contrairement à une position verticale qui fera forcer sur le poignet. En termes d'accessibilité aux personnes en fauteuil, les positions horizontales et inclinées (environ 30° à 45°) pourront être adaptées aussi bien aux personnes avec ou sans handicap, contrairement à la position verticale.

Même en position horizontale, une orientation peut être donnée lors d'un certain type d'usage tel que l'aide à la vente en côte à côte, souvent la solution demandée par les établissements de conseil travaillant sur la qualité de la relation entre les conseillers et les clients. Si l'on oriente une partie de l'interface, la table détectant la position et l'orientation des doigts, nous pouvons imagi-

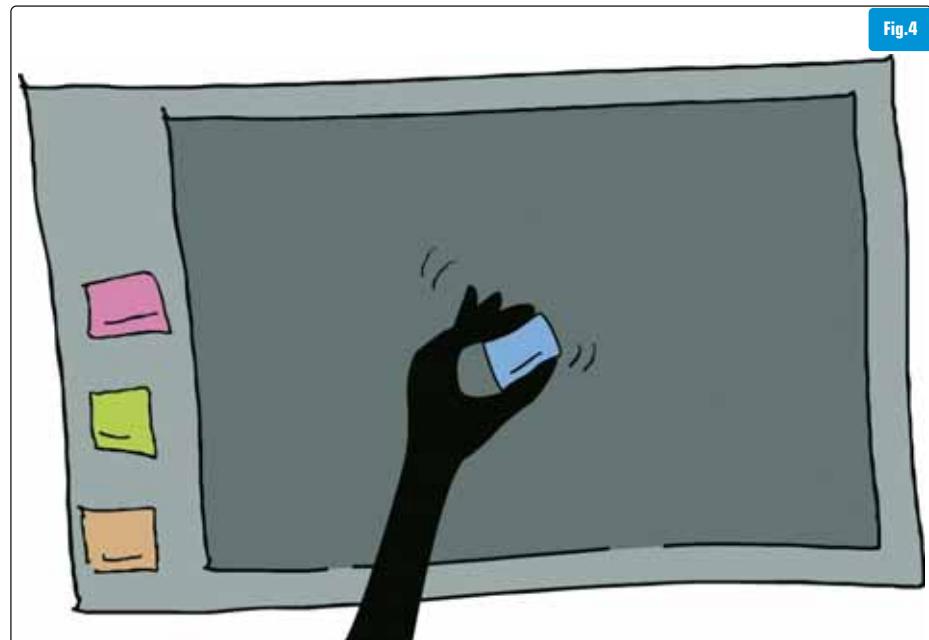
Fig.2



ner que des éléments « mobiles » de l'interface s'orientent par rapport à l'orientation du doigt qui va se poser sur lui. Pour ces expériences à 360° les développeurs pourront utiliser le Scatterview fourni avec le SDK Surface 2. Le ScatterView est un composant qui se comporte comme une ListBox ou ComboBox, mais dont l'affichage et la manipulation des éléments est tactile. Ce composant va afficher les éléments sous forme de petite zone tactile ayant le rendu de l'élément souhaité et pouvant être manipulé à sa convenance (rotation, translation, etc.) sur toute la zone tactile. Cependant pour que le ScatterView oriente son contenu en fonction de l'utilisateur automatiquement, c'est au développeur au moment du contact si c'est un doigt orienté, de remettre l'élément visuel dans le bon sens. Dans le cas de notre application photo, l'interface ne sera pas orientée et nous choisirons de positionner la table Surface à l'horizontale pour que plusieurs clients ou une famille puissent l'utiliser simultanément.

Bien que chacun puisse s'aménager sa zone de manipulation, forcer les utilisateurs à rester dans une partie de l'écran entraîne une expérience utilisateurs non-collaborative, c'est pour cela qu'il n'est pas conseillé de diviser l'écran en plusieurs parties.

Au niveau de l'interface des applications, elle sera obligatoirement « orientée » en position verticale et inclinée, à l'horizontale l'interface n'a pas d'orientation pour permettre aux utilisateurs de se positionner tout autour de la table. Le choix de l'utilisation à 45° (techniquement nous pouvons incliner la table de 0° à 90° mais ergonomiquement 30° à 45° est la meilleure orientation pour l'inclinaison du poignet) ou à la verticale, implique donc forcément la réalisation d'une application orientée. Attention à ce que l'utilisateur ne puisse pas réorienter l'interface à l'envers pendant son utilisation même s'il pose son doigt à l'envers. Ces orientations permettant deux

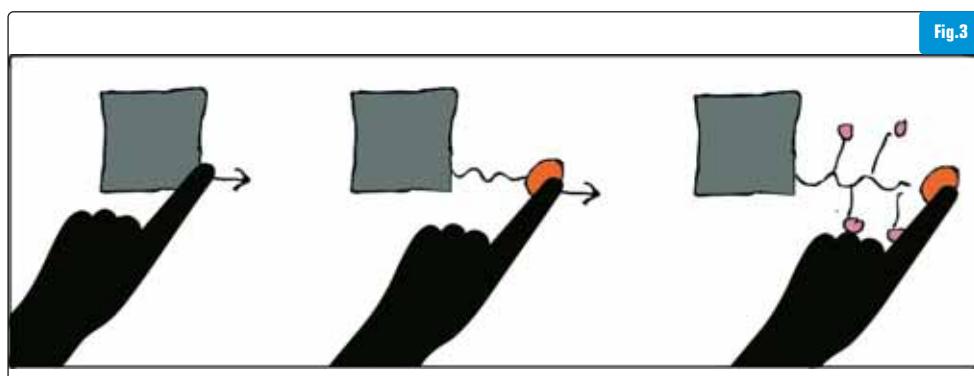


utilisateurs simultanés, les applications doivent donc fonctionner en multiutilisateur. L'erreur souvent effectuée est, par exemple, lors de la réalisation d'une application de dessin, de ne mettre qu'une palette de couleur avec une seule gomme. A chaque fois qu'un des utilisateurs va changer de couleur ou de forme de pinceau ou décide d'utiliser la gomme cela changera pour tous les utilisateurs simultanément. On peut aisément imaginer la mauvaise expérience que cela va entraîner pour celui dont le pinceau rouge se transforme en gomme.

> Les gestuelles

La table ne reconnaît que la présence d'un contact, peut le suivre durant toute sa durée de vie sur la surface et enfin elle peut tracer son inclinaison (rotation). C'est ensuite aux API (notamment WPF 4 et le SDK Surface 2) de déterminer les gestuelles classiques comme le zoom, la rotation, la translation etc. Il existe d'ailleurs une multitude de gestuelles pour les utilisateurs.

- Le tap : appuyer et relever rapidement sur une zone.
- Hold : appuyer et garder le doigt posé.
- Slide : appuyer et déplacer le doigt.
- Slide and move [Fig.3] : poser le doigt sur un élément graphique et le déplacer
- Flick : poser le doigt et le déplacer très rapidement (gestuelle utilisée pour naviguer dans un panorama Windows Phone par exemple).
- Flick to move : poser le doigt sur un élément graphique et le déplacer en bougeant le doigt très rapidement, comme pour le lancer.
- Touch and turn : poser le doigt sur un élément et le faire pivoter (avec un doigt).
- Spin : faire pivoter un élément avec deux doigts (gestuelle de rotation rapide).
- Stretch : poser deux doigts sur un élément et les éloigner l'un de l'autre (agrandissement d'un élément).
- Shrink : poser deux doigts sur un élément et les rapprocher l'un de l'autre (rétrécissement d'un élément).
- Pinch : poser deux doigts d'une même main et rapprocher les doigts l'un de l'autre (rétrécissement, dezoom d'une photo par exemple).
- Squeeze : poser plusieurs doigts d'une même main sur des éléments et les rapprocher les uns des autres.
- Spread : poser plusieurs doigts d'une même main sur des éléments et les éloigner les uns des autres.
- Pin turn : appuyer sur un élément avec un doigt et tourner avec deux doigts d'une autre main.



Vous pouvez en inventer d'autres si elles sont intuitives. Le plus important c'est d'utiliser la bonne gestuelle pour la bonne action. Retrouvez des visuels et des exemples de toutes ces gestuelles dans le *Surface2.0 Design and Interaction Guide*.

> Cas pratique

Repartons sur l'idée de l'application photos/vidéos pour une agence de voyage. Dans un projet tel que celui-ci les équipes pluridisciplinaires sont les plus pertinentes. L'équipe avec laquelle je travaille est au moins constituée de développeurs, graphistes et designers industriels spécialisés dans l'interaction.

Le processus est le suivant : 1^{re} étape, le designer conçoit l'application à l'aide de maquettes en papier à la bonne échelle pour tester l'encombrement et les interactions, il réalise ensuite les storyboards au crayon. La 2^e étape est la validation par l'équipe technique des storyboards du designer.

C'est cette étape sur laquelle nous allons revenir. Dans le storyboard apparaît la demande suivante : Il faudrait que nous puissions avoir des objets physiques qui correspondent aux catégories des différents albums photo. Lorsque nous les posons sur l'écran de la table, ils font apparaître des photos liées aux thèmes. Le contenu doit apparaître dans l'interface comme accroché à l'objet [Fig.4].

Réponse de l'équipe de développement :
Nous allons associer chaque thème à un Tag Microsoft Surface, à chaque valeur des Tags nous allons associer une librairie d'images. Ensuite nous allons créer un custom control pour l'affichage en bataille des images et les interactions qui vont avec.

Nous ferons la liaison entre le tag et le custom control via un TagVisualizer (composant du SDK Surface 2, permettant d'associer un visuel à un Tag).

La deuxième demande du designer est qu'il faudrait que les photos soient regroupées sous l'objet et que nous puissions en décrocher certaines [Fig.5].

Réponse de l'équipe de développement :
Nous allons repartir du custom control précédent et lui ajouter des gestuelles et interactions avec le ScatterView principal, le tout sera encapsulé dans un custom control pour regrouper les interactions.

Il faudrait que les utilisateurs puissent zoomer et tourner les photos et vidéos. Plusieurs utilisateurs peuvent le faire simultanément [Fig.6].

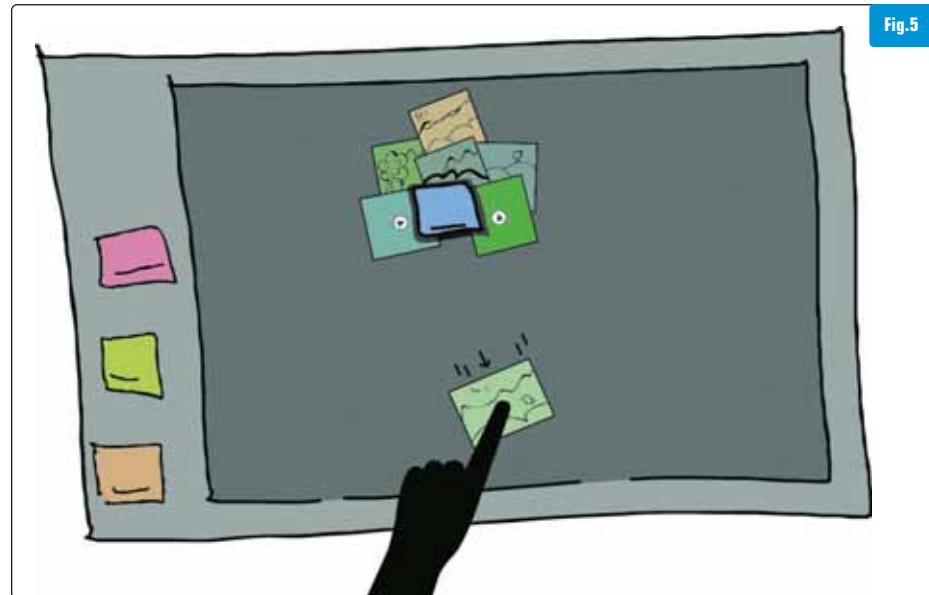


Fig.5

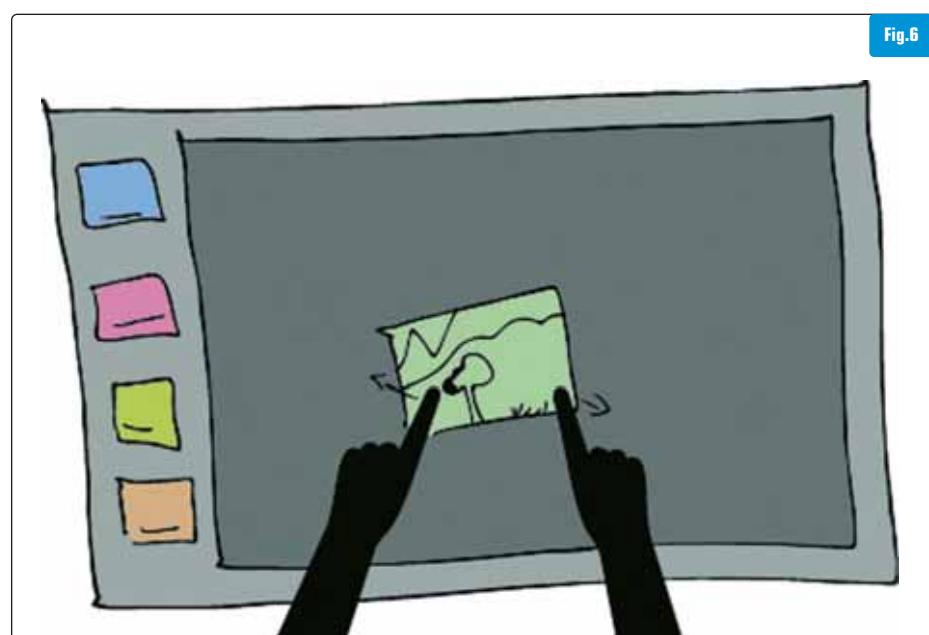


Fig.6

Réponse de l'équipe de développement :
Nous allons utiliser un ScatterView.

Le designer demande qu'on puisse lire des vidéos, les redimensionner et déplacer en même temps que la vidéo se lise.

Réponse de l'équipe de développement :
Pas de soucis, un MediaElement WPF fera l'affaire pour l'affichage des vidéos, et le ScatterView fera le reste pour le redimensionnement et la manipulation.

Pour finir, dans son storyboard, le designer explique qu'au travers d'une application mobile nous pouvons ajouter des photos de sa propre collection dans une catégorie de l'application Surface. Les nouvelles photos apparaîtront instantanément sur la table si l'objet lié au thème de notre photo est déjà posé sur l'écran.

Réponse de l'équipe de développement :
Oulalala... Dans ce cas, il faut créer un service WCF qui fera la connexion entre la table Surface et le Windows Phone. Ce service peut être hébergé sur la table elle-même (aussi dans l'application elle-même). Une fois les photos ajoutées dans le Windows Phone, on appelle ce service pour lui pousser l'information. Du côté Surface, le service WCF hébergé au sein de l'application va récupérer l'information et l'afficher sur la table.

Pour plus d'information sur le design et les interactions sur Surface : <http://www.microsoft.com/download/en/details.aspx?id=26713>

Johanna Rowe

Designer industriel et interactif
chez Winwise, MVP Microsoft Surface

Comment traquer (efficacement) les bugs de son site Web ?

Il paraît normal de contrôler sur un serveur Web que la dernière modification apportée à son site a bien été prise en compte. Par contre, parcourir à nouveau l'ensemble des pages pour vérifier qu'aucun bug n'a été généré par cette dernière mise à jour se révèle particulièrement contraignant. Pour minimiser cette charge de travail, il existe des outils d'automatisation de tests, dont nous allons présenter le mode de fonctionnement et les grands principes.

Tout site ou application Web subit des évolutions au fil du temps. Si la faible volumétrie du contenu facilite considérablement les tests dans un premier temps, les mises à jour successives nécessitent des efforts de plus en plus importants lorsque le site grossit, à la fois en termes de nombre de pages et de richesse du contenu de chacune de ces pages. La phase la plus fastidieuse consiste alors à vérifier à nouveau et sans cesse que les ajouts et modifications apportés n'ont pas engendré des régressions dans les autres parties du site. Par exemple, le changement de destination en cliquant sur un lien hypertexte, pour accéder à une nouvelle page, peut se traduire par l'impossibilité d'accès à l'ancienne page. Pour effectuer un tel contrôle manuellement, il faut partir de la page d'accueil et recenser toutes les pages affichées à partir d'une navigation exhaustive sur le site. De plus, les possibilités d'accès d'une page A vers une page B peuvent être multiples, complexifiant considérablement le maillage de l'arborescence du site (accès depuis le menu principal, image dans le site, résultat de recherche, bouton de validation, accès depuis le pied de page...). Cette tâche, particulièrement lourde, peut se révéler insoluble quand on souhaite par exemple valider le bon fonctionnement de son site sur la majorité des navigateurs Web encore opérationnels sur le marché.

UNE PREMIÈRE RÉPONSE AUX TESTS MANUELS, LES OUTILS D'AUTOMATISATION

Pour ces tâches répétitives, il existe plusieurs solutions, dont l'emploi d'outils d'automatisation de tests fonctionnels. Ceux-ci ne

permettent pas de valider le bon fonctionnement d'un site ou d'une application de manière aussi poussée qu'un testeur humain, mais permettent la détection de nombreuses anomalies. Le principe de fonctionnement de ces offres est quasiment toujours le même. Le testeur démarre son outil, et lance le module d'enregistrement. À partir de ce moment, il suffit de naviguer avec la souris sur l'interface de l'application à tester, comme lors d'une navigation classique, et de parcourir l'ensemble des pages que l'on souhaite contrôler. Durant cette phase, l'outil de test enregistre toutes les opérations effectuées par le testeur, à savoir les mouvements de souris, les clics, les saisies clavier, etc. mais également les informations liées à l'application testée. Une fois la navigation terminée, il suffit d'arrêter l'enregistreur, via généralement un bouton d'arrêt proposé en permanence en fenêtre popup au dessus du navigateur. Le scénario est alors enregistré et peut à volonté être enrichi et rejoué. Le rejeu se traduit par la réexécution du scénario enregistré lors de la création des scripts, le testeur revoyant défiler très rapidement à l'écran l'ensemble des opérations qu'il a pu effectuer initialement sans avoir à intervenir. Cette opération de rejeu ressemble à un « screencam », si ce n'est que l'outil de tests opère sur les objets de l'interface et non sur de simples enchaînements d'images statiques. Ceci permet à l'outil de notamment contrôler que les résultats obtenus sur l'interface testée sont les mêmes que ceux attendus. L'intérêt majeur de ce rejeu est de s'affranchir de la présence du testeur, celui-ci pouvant effectuer une pause pendant l'exécution ou bien

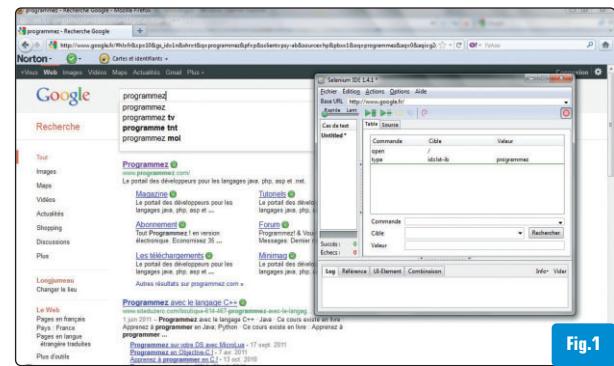


Fig.1

lancer le test le soir pour analyser les résultats obtenus le lendemain matin.

LES OUTILS D'AUTOMATISATION DE TESTS POUR QUI ?

Les testeurs sans aucune connaissance de la programmation peuvent utiliser certains outils d'automatisation qui proposent des assistants complets. À partir de l'enregistreur, un premier niveau de contrôle est offert, permettant notamment aux équipes utilisatrices et/ou maîtrise d'ouvrage dans les entreprises d'effectuer une première validation avant tests plus poussés. Ces derniers s'adressent plutôt aux Webmasters et intégrateurs HTML qui possèdent quelques notions de programmation Web, avec des connaissances du HTML, du Javascript, et quelques bases en PHP et en programmation SQL. Ces utilisateurs, souvent propriétaires de leur propre site Web, peuvent aller beaucoup plus loin dans l'utilisation des outils de tests fonctionnels. En effet, ceux-ci, par exemple pour les tests menés sur les applications Web, reposent sur le DOM (Document Object Model) pour les actions à effectuer sur les objets de l'interface ainsi que pour l'analyse des résultats obtenus. Enfin, seuls les informaticiens pourront pleinement exploiter le potentiel de ces outils d'automatisation de tests, notamment via la nécessité de s'interfacer avec les méthodes

ou fonctions de l'application testée. Il n'y a néanmoins pas de catégorie de testeur plus ciblée qu'une autre dans l'utilisation d'outils d'automatisation de tests. Les caractéristiques de l'application ou du site testé, l'objectif recherché avec l'emploi d'un outil d'automatisation ou encore le rythme et la forme des évolutions apportées à l'application sont autant de paramètres à prendre en considération avant de se lancer dans ce type de solutions.

PREMIERS PAS AVEC SELENIUM

Selenium est un outil d'automatisation de tests fonctionnels libre et gratuit. Celui-ci comprend plusieurs composants, tels Selenium IDE pour l'atelier de développement et Selenium Server pour le moteur d'exécution. Pour effectuer son premier test, le plus simple consiste à télécharger sur le site <http://seleniumhq.org> le plug-in Selenium IDE pour Firefox. Une fois le composant récupéré et la mise à jour de Firefox effectuée, il suffit de lancer Selenium IDE depuis la barre d'outils de Firefox. A partir de cet instant, l'enregistrement démarre automatiquement. On peut naviguer sur internet, en allant par exemple sur la page d'accueil de Google, saisir « programmez » [Fig.1], rechercher, puis cliquer sur le premier lien de résultat qui se traduit par l'ouverture de la page d'accueil du site Programmez. L'appui sur le bouton d'arrêt d'enregistrement marque alors la création de son premier scénario de tests [Fig.2]. Il est possible de rejouer celui-ci, plus ou moins rapidement. On s'aperçoit ainsi que le test rejoué lentement fonctionne parfaitement [Fig.3], alors que celui rejoué à vitesse maximale génère une erreur, Selenium ne pouvant trouver le résultat de la recherche Google dans le temps d'attente alloué par l'outil [Fig.4]. Le résultat d'échec

sera identique, même à vitesse de rejet la plus lente, si les sites Google ou Programmez deviennent inaccessibles.

DES TESTS PLUS POUSSÉS, AVEC DES FONCTIONNALITÉS ÉTENDUES

Pour aller plus loin avec Selenium, il faut passer par le composant serveur qui permet de coder directement l'ensemble de ses tests. S'il est possible d'exporter les tests conçus automatiquement depuis Selenium IDE pour par exemple les retravailler en Java sous Eclipse puis les exécuter à partir de Selenium Server, il est néanmoins regrettable que l'on ne puisse ensuite enrichir ses scripts de cette manière, la régénération depuis Selenium IDE écrasant alors les éventuels tests plus poussés qui auraient pu être ajoutés en Java. Ceci se traduit in fine, pour des tests poussés, par la mise à l'écart de Selenium IDE et l'impossibilité de profiter d'assistants pour préparer et mettre à jour plus simplement ses propres tests. S'il est ainsi possible d'utiliser le navigateur Web de son choix pour les tests, cette approche confine l'utilisation de Selenium pour des tests élaborés à une population d'informatiens. Ces derniers pourront profiter d'un environnement unique, Eclipse par exemple, dans lequel ils pourront à la fois apporter les évolutions à leur application tout en faisant évoluer les scenarii de tests [Fig.5].

Pour aller plus loin tout en profitant d'un environnement intégré, il est possible de s'orienter vers une offre telle que TestComplete, de l'éditeur AutomatedQA. Non représenté en France, ce qui se traduit notamment par un support exclusivement en anglais, cet outil reste simple d'approche, proposant de nombreux assistants et une notion de fenêtre unique permettant d'aller beaucoup plus loin dans les tests à partir du

même outil. Pour effectuer ce test plus approfondi, prenons l'exemple d'un intranet, application transactionnelle PHP-Symfony-MySQL-JQuery au sein de laquelle la validation du contenu des pages est requise. Sur la page de pose des jours de congés, le test va consister à valider qu'à l'ouverture de la page, le bout de code initialisant à la date du jour la période de congés à poser fonctionne bien [Fig.6]. Pour effectuer ce test avec TestComplete, il suffit de lancer l'enregistrement du scénario et de naviguer sur la page de pose des jours de congés. A partir de là, l'outil voit les objets composant la page, et va ainsi permettre la récupération de la date du jour. Une comparaison avec une variable contenant le jour, le mois et l'année permet alors de poser un point de contrôle validant le bon fonctionnement de l'application sur ce point lors du rejet [Fig.7].

Ces tests élémentaires sont assez faciles à implémenter, les évolutions continues de l'application Web rendant cependant la maintenance et la viabilité des scénarii compliqué. Par exemple, lorsqu'on lance un enregistrement et que l'on sélectionne le premier lien hypertexte dans une liste, il faut bien s'assurer lors du rejet que le lien que l'automate va actionner correspond bien à celui que l'on avait initialement sélectionné. Ainsi, dans le cadre d'un blog ou d'une liste triée, il faudra peut-être aller plus loin que de simplement sélectionner l'élément en tête de liste. Pour ce faire, une solution consiste à invoquer une URL construite par programmation, notamment si le lien recherché n'apparaît plus sur le premier bloc de la liste paginée. Autre exemple, sur de la comparaison d'images, la possibilité de définir un taux de comparaison pour estimer si une image résultat est considérée comme identique à l'image d'origine est très intéressante. Sur une photo au format JPG, un taux de res-

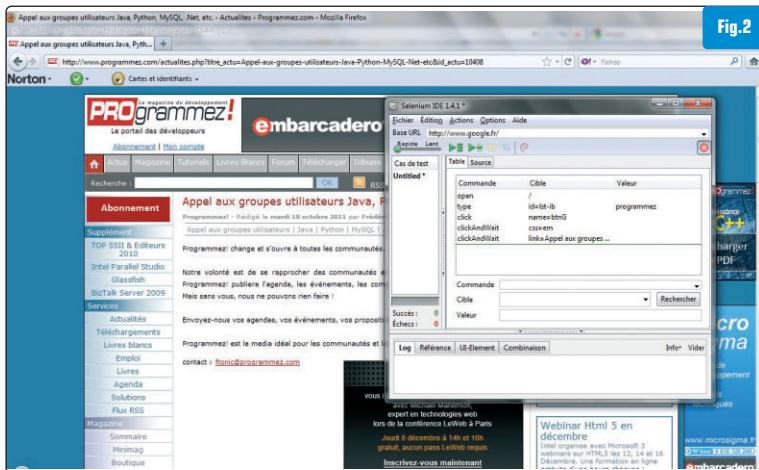


Fig.2

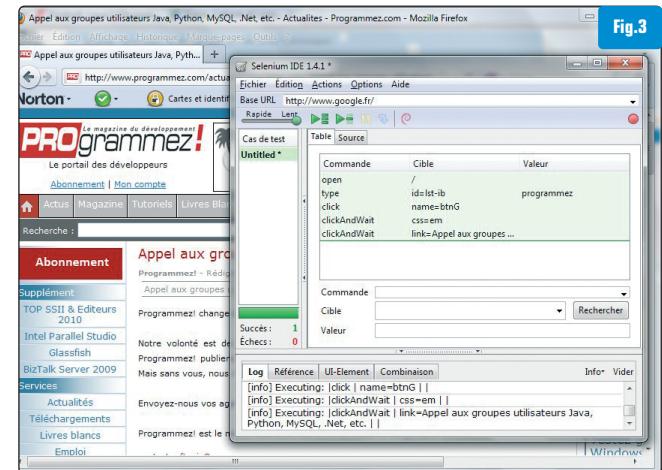


Fig.3

semblance supérieur à 80% sera souvent suffisant, alors que sur un schéma GIF un taux inférieur à 95% signifiera que les deux images comparées sont différentes. Néanmoins, la problématique principale consiste à bien localiser la position de l'image avant d'effectuer la comparaison d'images, étant donné qu'il n'est pas possible de se baser exclusivement sur les coordonnées en pixels pour être sûr que l'on compare la même chose.

LES GRANDES FONCTIONNALITÉS ATTENDUES D'UN OUTIL D'AUTOMATISATION DE TESTS FONCTIONNELS

Un outil d'automatisation de tests fonctionnels doit donc avant tout disposer d'un enregistreur, directement à partir d'une navigation sur le site ou l'application, et d'une fonctionnalité de rejet. Ceci permet à un non-informaticien d'enregistrer puis de rejouer autant de fois que souhaité son scénario sans avoir recours à de la programmation. Néanmoins, de nombreuses autres fonctionnalités sont requises pour mener des tests plus poussés, fonctionnalités plus ou moins abouties selon les outils présents sur le marché.

Un des aspects les plus importants de ce type d'outil concerne la capacité de gestion et de reconnaissance des objets de l'interface. En effet, il ne suffit pas lors du rejet de se repositionner en coordonnées graphiques pour effectuer une action, encore faut-il pouvoir localiser un objet afin de pouvoir mener l'action appropriée. Par exemple, si lors de l'enregistrement du script de test un bouton « Enregistrer » se trouvait en bas à gauche de son navigateur, il ne faut pas qu'une redéfinition de la charte

du site le positionnant désormais en bas à droite génère une erreur lors du rejet. Ainsi, il est demandé à l'outil d'automatisation des tests de reconnaître les objets lors de l'enregistrement afin de pouvoir effectuer l'action souhaitée lors du rejet, quel que soit leur positionnement à l'écran. Pour aller plus loin, certains outils gèrent même un référentiel d'objets, voire permettent d'effectuer un mapping d'objets pour l'activation d'événements non prévus sur un objet non reconnu lors de l'enregistrement. Ceci est important pour des enregistrements effectués sur des interfaces particulières comprenant des objets non standard, ou plus communément sur des utilisations dérivées d'objets graphiques. Par exemple, pouvoir mapper une image cliquable en bouton permet de gérer cette image avec toutes les propriétés propres à un bouton, comme le clic pour validation d'un formulaire.

Un autre aspect important concerne la mise en place de points de contrôle, et les paramétrages offerts pour ceux-ci. Exécuter un scénario et le rejouer est certes important, mais insérer des points de contrôle pour valider la cohérence du résultat obtenu par rapport à celui attendu l'est encore plus. Ainsi, certains outils, via la présence d'assistants, permettent a posteriori de positionner des points de contrôle à certains endroits du script. Ceux-ci ont pour objectif lors de la réexécution du scénario de valider les propriétés de la page et des objets, permettant d'interrompre ou pas l'exécution du scénario si des différences sont constatées par rapport à l'enregistrement initial.

D'un point de vue plus général, certains outils sont destinés aux tests de non régression sur des applications Web exclusivement, alors que d'autres vont supporter

d'autres types d'interfaces telles les interfaces graphiques Windows, les interfaces Java, les modules Flash, etc. Les capacités des outils à ce niveau diffèrent fortement, certains pouvant même concilier au sein d'un même test des objets issus d'interfaces hétérogènes. Par exemple, il est parfois utile de pouvoir piloter le navigateur Web depuis sa barre de menu, sur un test d'interface graphique, et continuer le test en validant le contenu HTML de la page, le test basculant alors dans un contexte Web. A l'inverse, une application Web peut ouvrir une feuille de calcul Excel ou Calc, et il peut être intéressant au sein des tests menés sur l'application Web de valider le contenu d'une cellule du tableau ouvert depuis son site Web. A noter à ce niveau que les outils permettant de forcer l'exhaustivité d'un test sur le positionnement en coordonnées graphiques du pointeur de la souris peuvent plus facilement s'affranchir de l'hétérogénéité des interfaces. Ces tests sont néanmoins fortement limités car contraints à une constance de la charte graphique des applications testées pour pouvoir répondre aux attentes des testeurs. Par exemple, dans cette situation, le simple fait de changer la taille d'un menu rend l'ensemble des tests caducs lors du rejet.

Un objectif prioritaire de ces tests consiste également à analyser les résultats obtenus lors de l'exécution d'un scénario. Ce dernier pouvant prendre plusieurs dizaines de

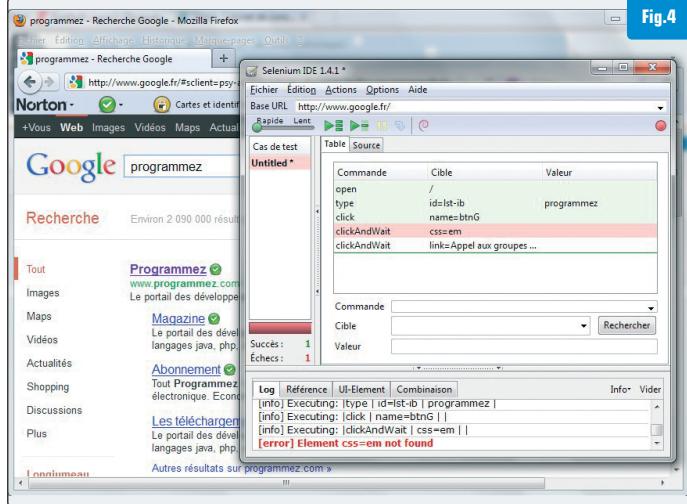


Fig.4

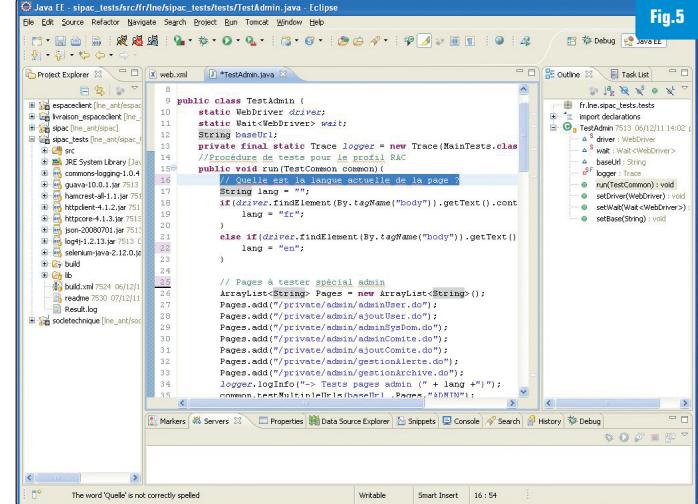


Fig.5

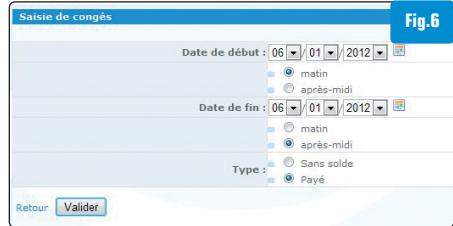


Fig.6

minutes, il est important de pouvoir rapidement obtenir une synthèse des résultats obtenus à partir des fichiers de résultats de tests créés par l'outil. Sur ce point, la lisibilité des journaux, hiérarchisant par exemple les tests et mettant en évidence les erreurs relevées, les points de contrôle en échec ou encore les avertissements ou doutes sur validation d'un aspect se révèlent à l'usage essentiels. Il en va de même pour la capacité offerte par l'outil pour paramétriser les journaux résultats, plus ou moins verbeux selon la volumétrie du test, ou encore pour un export de ces logs aux formats PDF, pour une exploitation de qualité, ou bien aux formats Calc ou Excel pour effectuer des analyses statistiques.

Pendant le déroulement du rejet d'un scénario, un événement extérieur à l'application peut intervenir, bloquant le bon déroulement de celui-ci. Par exemple, lors du test de son site, un popup d'avertissement « mémoire disponible insuffisante » généré par le système d'exploitation peut survenir, prenant le focus sur l'application testée. Dans cette situation, l'outil va donc rechercher dans cette fenêtre popup l'objet concerné par le test, et ne le trouvera bien entendu pas. Il est toujours possible, pour les outils permettant de faire cohabiter dans un même test plusieurs applications de type différent, de demander à ce moment-là de cliquer sur le bouton « annuler » et de poursuivre le test. Ceci se traduit néanmoins par l'obligation d'effectuer ce test à tous les niveaux du code du script, chose impossible pour un scénario même relativement petit. Certains outils gèrent heureusement la gestion d'événement sur l'ensemble d'un scénario, permettant d'effectuer un bout de script particulier lorsqu'arrive un événement extérieur particulier. La quintessence de la couverture de cette fonctionnalité consiste à proposer un

assistant pour traiter ces événements et la gestion d'erreur associée, mais très peu d'outils vont aussi loin à ce niveau.

Dans un contexte plus orienté développement d'entreprise, les outils d'automatisation de tests peuvent être utilisés pour valider le bon état de marche d'une application sur une couverture fonctionnelle plus large que celle qui pourrait être effectuée manuellement. Dans un tel contexte, ces outils vont alors bien au-delà du test de non régression. C'est ainsi le cas quand on souhaite valider l'application sur un jeu de données dynamique. L'enregistrement du scénario se fait alors à partir d'un scénario basé sur un jeu de données unique, et le scénario est rejoué en boucle autant de fois que souhaité sur un jeu de données le plus exhaustif possible. Par exemple, un scénario de test est censé valider la modification d'un formulaire proposant les informations d'un fournisseur. Lors de l'enregistrement, le testeur va ouvrir le formulaire d'un fournisseur en le sélectionnant dans la liste, modifier son prénom, appuyer sur le bouton enregistrer puis demander la vérification de la page de confirmation de la modification affichant le nouveau prénom. Lors du test, l'utilisation d'un jeu de données dynamique va permettre de rejouer ce scénario pour chacun des fournisseurs sans avoir à effectuer une opération supplémentaire, et ainsi valider le bon fonctionnement de l'application pour tous les fournisseurs. Selon les capacités de l'outil de test, ce jeu de données pourra être parcouru séquentiellement ou aléatoirement, à partir d'un fichier plat, d'un tableau ou même directement depuis des tables stockées dans une base de données.

Enfin, la richesse du langage de développement et de ses API constitue un des aspects essentiels dans la puissance proposée par un outil d'automatisation de tests fonction-

nels. En effet, pour des tests très poussés, il est essentiel de pouvoir s'appuyer sur un nombre de fonctions/méthodes le plus étendu possible. En ce sens, un outil reposant sur un serveur d'applications JEE dispose de capacités bien plus étendues qu'une solution articulée sur son script propriétaire. Par exemple, si l'outil de tests exploite pleinement JEE ou .Net, il est aisément d'automatiser lors de la détection d'une anomalie l'envoi d'un message électronique au testeur en s'interfaisant avec un serveur de messagerie. Ce service est impossible à implémenter avec un outil de tests proposant un langage de script propriétaire n'intégrant pas nativement cette fonctionnalité.

Selon la nature de tests à mener, de nombreuses autres fonctionnalités peuvent être utiles. Parmi celles-ci, on peut citer la comparaison de fichiers et d'images [Fig.8], la reconnaissance de caractères, très utile par exemple pour déterminer le contenu d'un fichier PDF, le paramétrage de l'exécution des scripts, la possibilité de debugger, l'interfaçage avec le back-office, etc. Un outil doté de la plupart de ces fonctionnalités offre ainsi une probabilité plus forte d'arriver à ses fins, les solutions de contournement délivrées pour effectuer un test étant ainsi d'autant plus nombreuses quand un point de blocage apparaît.

DANS QUELLES SITUATIONS RETENIR UN OUTIL D'AUTOMATISATION DE TESTS ?

Si l'intérêt d'automatiser les tests paraît évident, le retour sur investissement n'est malheureusement pas toujours au rendez-vous. En effet, la réalisation d'un scénario de tests complet est beaucoup plus coûteuse en temps que la réalisation manuelle des tests. A cela, il faut ajouter également une charge

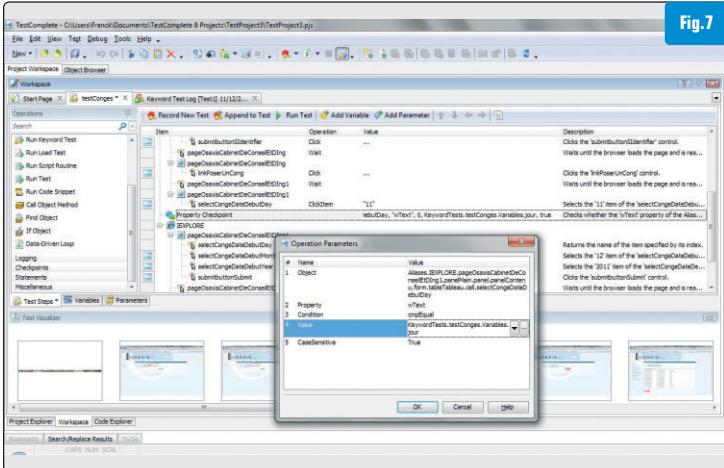


Fig.7

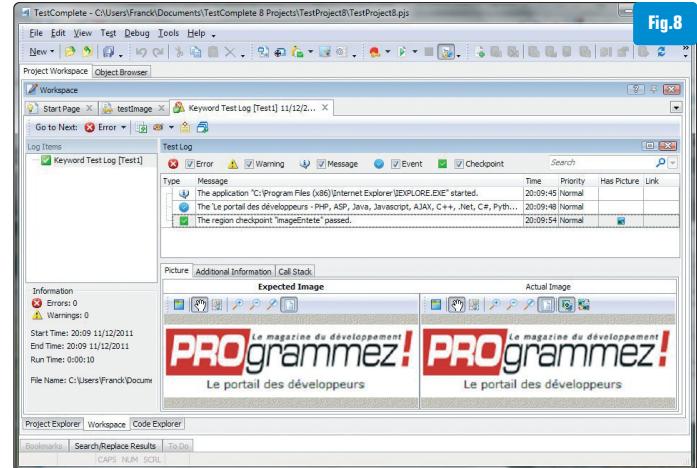


Fig.8

supplémentaire de maintenance lors de chaque rejet, plus ou moins importante selon le type d'évolutions apportées à l'application ou au site Web testé. Enfin, n'oublions pas le coût de la ou des licences de développement de l'outil qui peut s'ajouter au temps passé à réaliser et maintenir les tests.

D'un point de vue général, l'utilisation d'outils d'automatisation de tests fonctionnels devient très avantageuse à partir du moment où on arrive à maîtriser le périmètre des tests à couvrir. L'effort consenti par exemple pour couvrir 20% des tests sera assez faible, alors que la couverture de 30% des tests doublera la charge de travail requise, et il faudra même 10 fois plus de temps pour développer et couvrir 40% des tests à partir de son outil d'automatisation. L'autre facteur prépondérant concerne le type d'application testée.

Ainsi, un site Web censé fonctionner sur plusieurs navigateurs différents, sous les 2 ou 3 dernières versions de chaque navigateur, qui plus est dans un contexte multilingue, sera d'autant plus adapté à l'automatisation. Dans ce contexte, le script réalisé pourra ainsi avec très peu de modifications tourner sous toutes les versions de navigateur dans toutes les langues requises, multipliant ainsi l'utilisation de l'automatisation sans effort supplémentaire.

Un autre cas bien adapté concerne les phases de tests et de recette pour les applications manipulant une volumétrie importante de données. A partir d'un scénario écrit pour un jeu de données unique, il est possible de dérouler celui-ci sur un jeu de données bien plus exhaustif sans avoir à effectuer les tests manuels au cas par cas pour chacun des enregistrements de sa base de données. Enfin, dans le milieu de l'entreprise, les tests sur les applications

requérant d'incessantes évolutions et donc des tests répétitifs, notamment de non-régression, gagnent à être automatisés. C'est notamment le cas des éditeurs de logiciel particulièrement consommateurs en tests et donc en outils d'automatisation.

PANORAMA DES OFFRES

Au-delà de Selenium et de TestComplete présentés succinctement dans le présent article, une trentaine d'offres cohabitent sur le marché de l'automatisation des tests fonctionnels. Si certains ont pour objectif d'adresser la plupart des problématiques, certains se spécialisent dans des domaines d'expertise très ciblés afin de se démarquer de la concurrence. Voici un rapide aperçu de ces solutions.

En acquérant Borland et les outils de tests de Compuware, Microfocus s'est doté de deux solutions d'automatisation de tests fonctionnels. Tout d'abord SilkTest, produit initialement lancé par Segue Software avant un rachat par Borland, s'adresse avant tout aux maîtrises d'œuvre avec intégration des tests dans le cycle de vie du développement. De l'autre côté, TestPartner, provenant de l'ancienne offre de Compuware, est plutôt destiné aux maîtrises d'ouvrage. SilkTest, assez difficile à appréhender, se distingue par la couverture de nombreux types d'interface graphique, y compris au sein d'un même test.

TestPartner, qui permet d'effectuer des tests assez poussés à partir de son atelier de développement abouti, atteint néanmoins ses limites dès qu'il faut entrer plus précisément dans le code pour des besoins non couverts nativement par le produit.

IBM propose également deux outils à son catalogue, Robot et RFT, issus tous deux de la société Rational achetée en 2003. Plus

complet, Rational Functional Tester (RFT) est depuis quelques années clairement positionné comme l'offre phare d'IBM pour ce type de besoins. S'intégrant dans Visual Studio et Eclipse pour respectivement des développements .Net et JEE, RFT respecte ainsi la notion de fenêtre unique en proposant aux équipes de développement un outil de tests fonctionnel directement intégré dans l'atelier de développement. Au-delà du prix, le principal inconvénient de l'outil demeure sa lourdeur, une machine performante étant requise pour réaliser confortablement ses scripts de tests.

QA Wizard, de la société Seapine Software, ne manque pas d'intérêt malgré une présence encore marginale en France. Peu onéreux, QA Wizard est très simple d'utilisation. Le passage d'un mode « tableau » pour les débutants à un mode « code » pour les testeurs aguerris permet d'apprehender progressivement et en douceur la complexité des tests fonctionnels. Le support en France reste un des points faibles du produit.

Enfin, QuickTest Pro, de HP, représente ce jour l'offre phare dans le monde professionnel. WinRunner, autre produit d'automatisation de tests fonctionnels de l'éditeur, a été mis de côté au profit du très complet QuickTest Pro. Le référentiel d'objets est particulièrement bien conçu, celui-ci pouvant notamment être généré lors de l'enregistrement, enrichi manuellement ou encore importé à partir d'un autre scénario. Paradoxalement, le produit souffre principalement de sa complétude, difficile à prendre en main pour un débutant. De plus, le coût de la licence est assez élevé par rapport aux autres offres du marché.

Franck Gonzales

Gérant d'Osaxis (<http://www.osaxis.fr>)

L'information permanente



The screenshot shows the homepage of PROgrammez.com. At the top, there's a banner with the text "Le magazine du développement" and "Le portail des développeurs". Below it, a navigation bar includes links for "Accueil", "Magazines", "Tutoriels", "Livres blancs", "Forum", "Téléchargement", "Tribune", "Video", "Livres", "Avis d'experts", "Inserz", "Abonnez", and "Email". A search bar is also present. The main content area features several news items:

- Actu**: "Angoissé par vos restaurations?"
- Supplément**: "Windows 8 et Windows Server 8 en téléchargement !"
- Build conférence : Nous avons repensé Windows !**
- Actualités**: "Windows 8 et Windows Server 8 en téléchargement !"
- Technologies**: "Windows 8 et Windows Server 8 en téléchargement !"
- Services**: "Windows 8 et Windows Server 8 en téléchargement !"
- Abonnements**: "Windows 8 et Windows Server 8 en téléchargement !"
- Livres blancs**: "Windows 8 et Windows Server 8 en téléchargement !"
- Emploi**: "Windows 8 et Windows Server 8 en téléchargement !"
- Livres**: "Windows 8 et Windows Server 8 en téléchargement !"
- Agenda**: "Windows 8 et Windows Server 8 en téléchargement !"
- Solutions**: "Windows 8 et Windows Server 8 en téléchargement !"
- Plus RSS**: "Windows 8 et Windows Server 8 en téléchargement !"
- Magazines**: "Windows 8 et Windows Server 8 en téléchargement !"
- Actualités**: "Windows 8 et Windows Server 8 en téléchargement !"
- Tribune**: "Windows 8 : débat le 16 juillet avec Linux ?"
- Revue**: "Bing est un gouffre financier"
- Abonnement**: "Windows 8 et Windows Server 8 en téléchargement !"
- Archives**: "Windows 8 et Windows Server 8 en téléchargement !"
- Interactive**: "Windows 8 et Windows Server 8 en téléchargement !"
- Enquête lecteurs**: "Windows 8 et Windows Server 8 en téléchargement !"
- Tribune libre**: "Windows 8 et Windows Server 8 en téléchargement !"

On the right side, there are several advertisements:

- Arkeia Network Backup - la solution**: "• +200 plateformes supportées • sauvegardes et restaurations rapides • gestion intuitive par interface web".
- Découvrez HAPEE**: "Une version packagée et envoiée par la poste, logiciel open source, facile à installer et à configurer, avec un accès à des services de support, mis à jour et d'assistance."
- Les rencontres d'Open source**: "Sorinie Meet'08 le 20 octobre : développement PHP, le rôle de l'open source dans l'industrie et l'innovation".
- Prés de 4000 offres d'emploi**: "Postes immédiats ou à pourvoir dans les prochaines semaines".
- FLEX SERVER**: "Le 1^{er} serveur dédié entièrement configurables".
- Programmez e-Commerce**: "Programmez e-Commerce est disponible en version

● **L'actu de Programmez.com :**
le fil d'info quotidien

● **La newsletter hebdo :**
la synthèse des informations indispensables.
Abonnez-vous, c'est gratuit !

www.programmez.com

LE MOIS DES CADEAUX !

OFFRE
VALABLE
JUSQU'AU 29 FÉVRIER

Abonnez-vous **-50%**

jusqu'à

50%

Exceptionnellement, en février,
vous pouvez bénéficier, pour 1 euro de plus,
d'un des "livres numériques" au choix,
sélectionné par la rédaction de Programmez.

(accès html au contenu intégral de l'ouvrage
sur la Librairie numérique des Editions ENI,
d'une valeur de 21 à 31€ selon les titres)

Pour 1€ de +,
Un "livre numérique" des Editions Eni



Valeur des 11 numéros du magazine : 65,50€ + Valeur du livre : 21 à 31€
= Valeur totale : 86€ à 96,70€

Offre : 50€ seulement (ou 60€ avec accès illimité aux archives).

Code,
gestion de projets,
développement web, mobile.

Programmez ! est à la fois :
votre outil pratique :
articles de code par les meilleurs experts,
et votre veille technologique

TOUTES LES OFFRES EN LIGNE : www.programmez.com

Abonnez-vous à partir de 3,80€ seulement par mois

(à retourner, avec votre règlement à :
Groupe GLI, 17 chemin des Boulangers 78926 Yvelines cedex 9

- Abonnement 1 an au magazine + 1 « Livre Numérique ENI » : 50 € (au lieu de 65,45 €, prix au numéro + valeur du « livre numérique ») Abonnement seul : 49 €
 Abonnement Intégral : 1 an au magazine + Archives Internet et PDF + 1 « Livre Numérique ENI » : 60 € Abonnement Intégral seul : 59 €.
 Abonnement 2 ans au magazine + 1 « Livre Numérique ENI » : 79 € (au lieu de 130,9 €, prix au numéro + valeur du « livre numérique ») Abonnement 2 ans seul : 78 €
 Abonnement intégral 2 ans au magazine + Archives Internet et PDF + 1 « Livre Numérique ENI » : 89 € Abonnement intégral 2 ans seul : 88 €

Livres à Choisir : Html 5 CSS 3 Google Analytics Développez une application Android

Détails sur www.programmez.com/abonnement.php

Tarifs France métropolitaine

M. Mme Mlle Entreprise : _____ Fonction : _____

Prénom : _____ Nom : _____

Adresse : _____

Code postal : _____ Ville : _____

Tél : _____

(Attention, e-mail indispensable pour le livre numérique et les archives sur internet)

E-mail : _____ @ _____

Je joins mon règlement par chèque à l'ordre de Programmez ! Je souhaite régler à réception de facture



Repérer une valeur NULL

La valeur NULL d'une variable peut devenir rapidement une cause de perturbation dans certaines applications métiers. Bien sûr, un site web traite des données qui lui sont propres, mais votre site peut devenir une application à part entière, c'est pourquoi l'approche de conception et de réalisation reste identique, à la seule différence que des contraintes supplémentaires sont à prendre en compte.

LA PROBLÉMATIQUE DE DÉPART

Un site internet possède ses propres formulaires pour enregistrer ses données, qu'il traite en direct. Bien entendu, ce site peut proposer aussi des services par l'intermédiaire de web services, mais la provenance des données reste souvent limitée. Cependant, lorsque votre projet devient orienté métier, accessible par internet, il est nécessaire d'être plus vigilant sur le développement et par conséquent, cela demande une mise sous surveillance supplémentaire. Le point qui nous intéresse aujourd'hui concerne la provenance de données, c'est-à-dire les informations venant des variables GET ou POST, générées par différentes applications codées en différents langages (VB, C, C++, Java, etc.), tournant sur différents systèmes d'exploitation (Windows, Mac, Linux...) et permettant d'afficher le tout en PHP à travers votre navigateur. Même si on peut considérer qu'il s'agit d'envoyer des données avec des milliers de machines qui simuleront des milliers d'utilisateurs remplissant vos formulaires automatiquement. A ce niveau, vous êtes confrontés à un risque de bug, par l'intermédiaire de ces logiciels qui peuvent vous envoyer une ou plusieurs valeurs NULL ou erronées sans le vouloir, et par conséquent des effets de bord non souhaités peuvent apparaître dans votre projet Web.

COMMENT LE COMPRENDRE

Tout d'abord, la détection d'une erreur peut se repérer facilement si vous possédez une seule source de données comme un site internet dont vous êtes le seul à gérer la saisie, le contenu et l'affichage. Mais lorsque vous ne maîtrisez pas le nombre d'ordinateurs, ni la configuration et par conséquent des versions différentes de logiciels et de sources de données, cela devient rapidement un casse-tête pour repérer la machine qui peut provoquer des erreurs et pour couronner le tout, vous pourrez avoir ce problème aléatoirement, ce qui rajoute une difficulté supplémentaire. En résumé, des milliers de données et de contenus, avec une même structure codifiée selon votre projet.

DÉTECTER LE PROBLÈME

Pour mieux comprendre ce phénomène de données aléatoires, l'alerte provient souvent des utilisateurs ou du service qualité qui ne sont pas capables de reproduire ce phénomène systématiquement, mais qui vont vous expliquer que le résultat affiché est faux. Bien sûr, si vous ne possédez pas de service de contrôle dans votre entreprise, la définition d'absence de données devient problématique car cela joue au final sur la qualité de votre produit 'métier' et de sa distribution. Cependant quelques solutions existent sans qu'il soit nécessaire de mettre en place de grandes méthodes et principalement de sensibiliser les développeurs à la nécessité d'ajouter des indicateurs

d'alertes dans leurs fonctions, les classes et les routines de traitement. Ces indicateurs d'alertes peuvent être générés en PHP, en envoyant dans une fonction spécifique les messages d'erreurs :

```
<?php  
try {  
    // Votre code  
} catch (Exception $e) {  
    sauvegarde_les_erreurs($e->getMessage());  
}  
?>
```

SOLUTION

Il existe différentes solutions pour mieux résoudre et comprendre ce genre de problème de pertes de données car vous ne devez pas oublier que même si vous maîtrisez la structure des données, vous ne maîtrisez pas son contenu. Tout d'abord, vous pouvez vous appuyer sur les logs venant du serveur Apache ou de la base de données (MySQL par exemple), ce qui peut être long à traiter. Mais, surtout vous devez pouvoir accéder à ces sources d'informations, ce qui n'est pas toujours le cas. L'autre solution qui est la plus facile à mettre en place, consiste à enregistrer la provenance des données, les formats et le retour d'informations en enregistrant le tout dans un système de logs. Le système de logs est une solution intéressante car avec un même fichier, vous pouvez concevoir une hiérarchie se décomposant comme ceci :

Identification du type de données / Date / Nom de la machine

pour obtenir un résultat sous cette forme :

- bug-20120102-pc1.log
- bug-20120105-pc30.log
- ok-20120102-pc1.log
- ok-20120102-pc2.log
- ok-20120102-pc4.log

A l'intérieur de ces fichiers, vous allez enregistrer toutes les informations traitées, votre fonction PHP et suivant le cas, ces données seront enregistrées dans son fichier associé. L'avantage de cette pratique consiste à résoudre rapidement une anomalie. Lorsque vous avez défini le problème et les différents risques, vous devrez mettre en place dans votre projet un correctif pour éviter que celui-ci ne réapparaisse.

Ce genre de problème vient principalement au début du déploiement et de sa mise en production. Mais il peut réapparaître à tout moment, suite à la mise à jour d'une DLL, d'un driver matériel ou du système d'exploitation.



Christophe Villeneuve

Consultant pour Alter Way solutions, auteur du livre « PHP & MySQL-Mysqli-PDO, Construisez votre application », aux Éditions ENI. Rédacteur pour nexen.net, membre des Teams DrupalFR, AFUP, LeMug.fr, PHP7V.

Les outils des Décideurs Informatiques

*Vous avez besoin d'info
sur des sujets
d'administration,
de sécurité, de progiciel,
de projets ?
Accédez directement
à l'information ciblée.*



Actus | Evénements | Newsletter | Vidéos

LE MAGAZINE DES DÉCIDEURS INFORMATIQUES

SOLUTIONS & LOGICIELS

N°26 DÉCEMBRE 2011 JANVIER 2012

www.solutions-logiciels.com

IBM IBM a 100 ans et innove pour la collectivité

Dématérialisation du courrier De l'enveloppe à l'écran

Enquête exclusive 2012

Les projets des entreprises.

Les tendances

Mobilité, externalisation, consolidation, sécurité.

Le CLOUD relève l'industrie du STOCKAGE

Sécurité Dans les secrets des labos de SYMANTEC

CRM CRM social, cloud et SaaS, e-marketing, etc.

M 09551 - 26 - F: 6,00 € - RD

DÉCEMBRE 2011 - JANVIER 2012

www.solutions-logiciels.com

OUI, je m'abonne (écrire en lettres capitales)

Envoyer par la poste à : Solutions Logiciels, service Diffusion, GLIE - 17 chemin des Boulanger 78926 Yvelines cedex 9 - ou par fax : 01 55 56 70 20

1 an : 50€ au lieu de 60€, prix au numéro (Tarif France métropolitaine) - Autres destinations : CEE et Suisse : 60€ - Algérie, Maroc, Tunisie : 65€ , Canada : 80€ - Dom : 75€ Tom : 100€

10 numéros par an.

M. Mme Mlle Société

Titre : Fonction : Directeur informatique Responsable informatique Chef de projet Admin Autre

NOM Prénom

N° rue

Complément

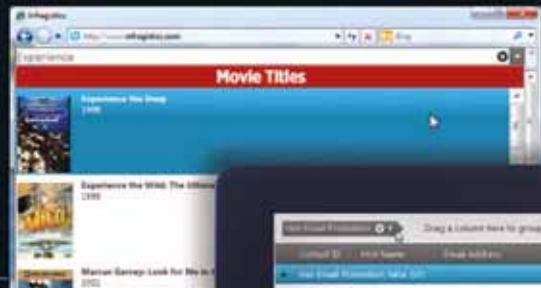
Code postal : Ville

Adresse mail

Je joins mon règlement par chèque à l'ordre de SOLUTIONS LOGICIELS Je souhaite régler à réception de facture

Contrôlez Votre Expérience Multi-Plateforme

check out infragistics.com/jquery

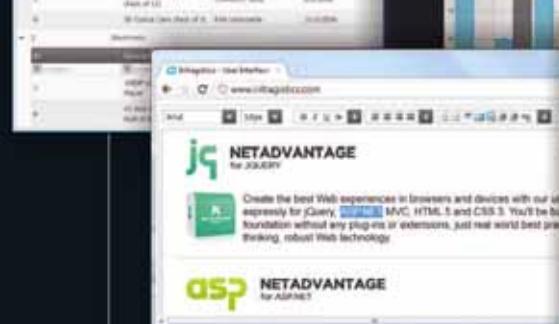


COMBO

Ce pack d'applications riche en fonctionnalités offre par défaut la suggestion automatique et intuitive ainsi que le filtrage et la complétion automatique des données.

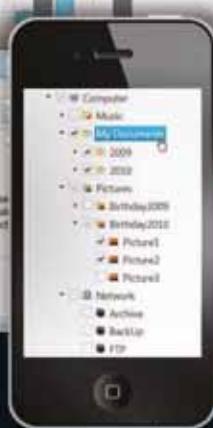
GRILLE HIERARCHIQUE

Une grille de données extensible offrant plusieurs relations parent-enfant est la clé de votre de toutes vos applications de bases de données.



EDITEUR HTML

Donnez à vos utilisateurs une expérience d'édition HTML de haute qualité en incorporant une interface JQuery à votre outil d'édition WYSIWYG.



TREE
Simplifiez l'apparence de vos données hiérarchiques en offrant à vos utilisateurs un design, des fonctionnalités et une expérience unique qu'ils ont toujours voulu avoir.

BUSINESS CHARTING

Combinez facilement les grilles de style d'Outlook avec des graphiques riches et interactifs pour offrir une solution fonctionnelle, complète et portative.

