

Responsive (web) design

*Créez les interfaces web une fois,
utilisez-les partout !*

CARRIÈRE

La fin du webmaster

Les métiers
et les salaires
du web

AGILITÉ

Scrum

au service du
développeur

- Découvrez le fonctionnement de la JVM
- Des interfaces dynamiques sous Android

- Redécouvrez Emacs
- Combiner Amazon EC2 et Amazon S3
- Les débordements de tampon

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

M 04319 - 162 - F: 5,95 €



Multi-Plateforme

Bureau

Créez des applications d'entreprise performantes, scalables et conçues pour le tactile sur la plateforme de votre choix.

Construisez des expériences tactiles qui s'appuient sur les standards avec HTML 5 et JQuery, pour le bureau, les tablettes et les smartphones ; ou en ciblant plusieurs de ces appareils grâce à des frameworks comme PhoneGap et MVC.



Natif pour Mobiles

Développez des applications riches, spécifiques à chaque expérience utilisateur sur iOS, sur Android, et sur Windows Phone ; aussi bien que des applications multi-plateformes avec Mono-Touch.



Prototypage d'Interfaces

Téléchargez votre version d'évaluation gratuite
infragistics.com/telechargez

Explorez des idées de design en réalisant rapidement des Wireframes centrés sur l'utilisateur, en prototypant et en faisant des évaluations avant d'écrire la moindre ligne de code.



Infragistics Ventes France  0800 667 307 • Infragistics Ventes Europe +44 (0) 800 298 9055

Copyright 1996-2013 Infragistics, Inc. All rights reserved. Infragistics and NetAdvantage are registered trademarks of Infragistics, Inc. The Infragistics logo is a trademark of Infragistics, Inc.

Natif vs HTML 5 : et le gagnant (ou le perdant) est ?

Kendo UI (éditeur spécialisé en HTML 5) a dévoilé une nouvelle étude globale sur le développement auprès de 5 000 développeurs et spécialistes de la technologie, en janvier 2013. L'Amérique du Nord pèse 31 %, l'Europe 35% et l'Asie-Pacifique 23%. La moitié des réponses vient de petites entreprises et d'indépendants.

Une étude ne fait pas le marché et ne doit pas être généralisée. Elle indique une tendance, un état des lieux (même imparfait).

> HTML 5 : au secours, il écrase tout !

Tout d'abord, le type d'applications développées avec HTML 5 concerne avant tout les logiciels de productivité au sens large (54 %), puis suivent les utilisateurs/orientés clients. Les jeux arrivent très loin derrière... Il est intéressant de constater que l'Amérique du Nord développe beaucoup avec HTML 5 (70 %), l'Europe pointe à 59 %, contre 60 % en Asie-Pacifique. Sur les technologies web, les réponses sont très disparates. Ainsi les formulaires et la validation (des données, des formulaires) sont d'une grande importance pour 32 % des développeurs interrogés, les données pour 31 %, la flexibilité des CSS pour 26 %. A l'opposé, les API liées aux terminaux, ECMAScript 6, le déconnecté et la gestion media sont d'une importance moindre pour un tiers des interrogés.

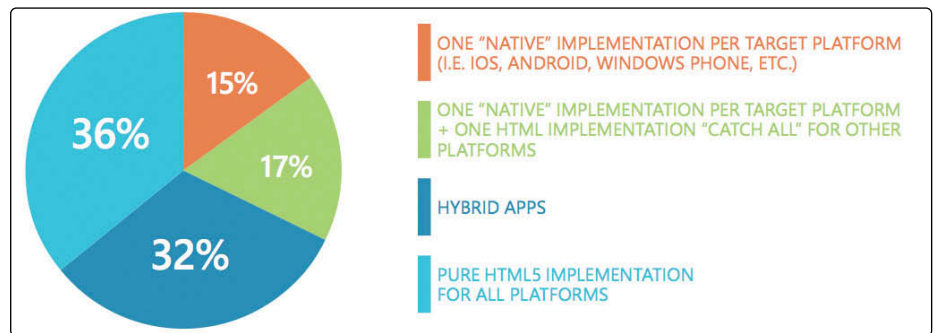
> Faut-il un débat HTML 5 vs développement natif ?

Kendo UI étant un fournisseur d'outils HTML 5 / CSS, la question ne surprend pas. Le développeur utilise-t-il des SDK natifs ou du pur HTML 5, ou encore un modèle hybride ?

Les pour du natif : performance, accès aux fonctions matérielles, optimisation.

Les pour du développement web : le coût moindre, flexibilité du modèle, portabilité, adapté pour toutes les tailles d'entreprises. Selon l'étude, le développement natif serait en moyenne à 15 % dans les entreprises, contre 36 % pour le pur HTML 5 (pour toute plateforme). En revanche, on constate que l'usage de code natif dans un projet web est fortement présent (32 % en moyenne). Pour la portabilité et le multiplateforme, et pour avoir les mêmes fonctionnalités sur l'ensemble des plateformes, HTML 5 est plébiscité, contre le modèle natif, vu comme trop contraignant. L'enquête semble être définitive sur la question :

Question : comment allez-vous relever le défi de la création d'applications pour plusieurs plateformes mobiles?



Réponse : 70 % votent HTML 5

On peut alors se demander si le phénomène n'est pas surmédiatisé ou biaisé. Il faut croire que non : HTML 5 est-il surmédiatisé ? 43 % sont totalement en désaccord avec cette affirmation. HTML 5 est fait pour créer des applications mobiles ? 78 % sont franchement d'accord. Et même pour tout type d'application, le développeur suit le mouvement. Pauvre développement natif !

> J'ai cru voir un méchant iOS !

iOS et Android écrasent tout. Les chiffres des ventes et des utilisateurs le prouvent. BlackBerry et Windows Phone prétendent à la 3e place, mais de très loin. L'enquête a voulu savoir quelle plateforme était facile à développer ou au contraire difficile.

Windows 8 et Windows Phone sont perçus comme faciles pour le développement, mais pas Android et encore moins iOS. Ces résultats sont assez étonnants, surtout pour Android. Car le modèle de développement est classique : Eclipse, SDK Java. Pour iOS, l'accès est un peu plus difficile. Il faut comprendre la logique Xcode (l'environnement de développement d'Apple) et utiliser Objective-C (dérivé du C). Un développeur C ou C++ aura plus de facilité à s'adapter qu'un développeur .Net ou Java. Est-ce que 50 % des développeurs interrogés codent pour et sur Windows ? Microsoft a toujours su créer des modèles de développement cohérents et rapides à maîtriser.

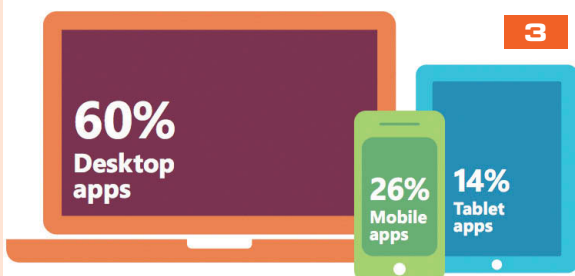
> Windows 8, plateforme émergente (la dure vie des systèmes)

« Etes-vous intéressé à développer en HTML 5 sur une des plateformes émergentes ? ».

Jusque-là, la question paraît intéressante. Mais les choix offerts nous ont fait sauter au plafond : on y trouve de tout ! Windows 8, Chrome OS, Firefox OS, Tizen et même BlackBerry 10. La présence de Windows 8 nous étonne. Oui, le système n'est sorti qu'en octobre 2012, mais est-ce pour autant une plateforme émergente ? On ne peut pas le comparer à Firefox OS.

Que dire en conclusion ? Que nos amis développeurs vont utiliser HTML 5 pour développer des applications desktop et site web classique à 87 %, un site mobile à 53 %. En revanche, le développement tablette est encore bien faiblard. Lancez un « tablette-thon » pour trouver de gentils développeurs !

Etude complète : <http://goo.gl/qXaWk>



3

BUZZWORD

Natif vs HTML 5 :
et le gagnant (ou le perdant) est ?3

ACTUS

En bref6



RESSOURCES

Les livres du mois12
Evènements14

matériel13



DÉVELOPPEUR DU MOIS

Geoffrey Parquet :
« Ayez toujours faim de technologie »18

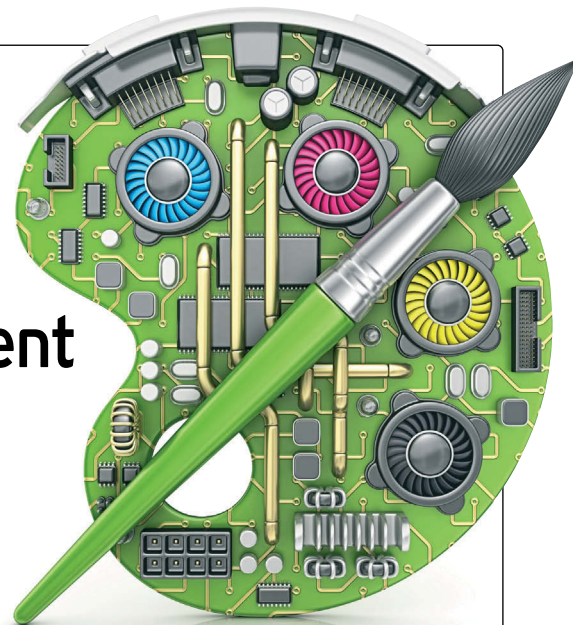
OUTILS

Faut-il passer à Zend Framework 2 ?20

VEILLE TECHNO

Responsive Design : une autre manière de penser le développement

Responsive Design, définition et approche24
Passer au « responsive web design »27
Votre site « responsive »30
Son premier site adaptatif
avec l'approche mobile-first32



A lire dans le prochain numéro

N°163-Mai 2013, à paraître le 27 avril 2013

- **Hacking : attaque et contre-mesures**
Le top 10 des failles de sécurité et comment les boucher
Concevoir un exploit avec Metasploit et avec Karmetasploit

- **Salaires 2013**
Quels salaires pour le monde IT et les développeurs ?
Quelles évolutions ? Quels profils privilégier ?



DOSSIER SCRUM

Scrum : l'agilité aide le développeur au quotidien

L'agilité et Scrum dans la vie du développeur	37
Utiliser (concrètement) Scrum	40
Du Slow Scrum à l'hyperproductivité : le rôle et les outils du ScrumMaster	42
Intégrer une équipe Scrum	45
Le Product Owner : « Le métier au cœur de la Technique ! »	47
ScrumMaster : devenir un expert Scrum	48

CARRIÈRE

La fin du webmaster Vive les métiers du web !	52
Formations et compétences	54
Brèves	55

JE DÉBUTE AVEC...

Utilisons Emacs	56
Démystifier le développement des plug-ins d'Eclipse 5e partie	60
Comprendre le fonctionnement de la machine virtuelle Java	65

CAS D'USAGE

Construire des IHM dynamiques sur Android avec les Fragments	68
Comment combiner Amazon EC2 et S3 pour un service web ?	72

PRATIQUE

Marmalade Quick : Développement rapide de jeux 2D multiplateformes	75
Débordement de tampon : comprendre le mécanisme et s'en prémunir	77

BUGTRACK

IntelliJ IDEA 12: bug d'applications Web	81
--	----



La vérité est-elle ailleurs ?

Le monde des processeurs est en train de connaître une véritable révolution. Le laboratoire First Processor a annoncé le premier processeur dit méta cognitif (ou MCP), pouvant être produit en série. Le MCP est capable d'apporter de l'intelligence artificielle à des robots, voire à des ordinateurs. Le MCP est fait pour apprendre, comprendre et agir. Pour les créateurs, il s'agissait de reprendre les bonnes idées de la bio-informatique, de l'informatique neuronale et bien entendu de l'IA. Déjà, il existe des robots capables d'agir seul et de penser (ou presque). Le MCP franchit une nouvelle étape avec des robots réellement autonomes et pensant par eux-mêmes. Faut-il alors commencer à réfléchir à des législations robotiques ou au moins à une charte éthique ? Faut-il adopter les lois fondamentales robotiques d'Asimov ?

HTML vit-il ses dernières heures sous sa forme actuelle ? Une pétition très sérieuse a été lancée par le WHATWG (groupe de travail à l'origine d'un fork de HTML5) en février dernier pour proposer un changement radical de HTML à partir de la version 6 ou 7. Remplacer purement et simplement le langage HTML actuel par un langage natif de type C ou C++ ! Idée folle ? Un peu sans doute. Pour le WHATWG, le langage natif a d'énormes avantages : performances, portabilité, normalisation (si on prend C ou C++). L'idée a germé avec la présentation du Runtime-Compiled C++, un framework permettant de modifier durant l'exécution du code C++, et ces modifications sont directement exécutées ! Mais rassurez-vous, HTML++ n'arrivera pas avant 2020 ou 2025 ! Pas sûr que le W3C voit cela d'un bon œil !

L'Open Source va-t-il devenir la norme dans les administrations françaises ? Depuis que le directeur informatique du gouvernement anglais a préconisé l'utilisation de l'Open Source dans les administrations publiques (logiciels, formats, standards), le sujet est revenu en force en France. Le ministre du redressement productif et la ministre déléguée à l'innovation et à l'économie numérique, n'ont pas caché leur intérêt pour cette démarche.

L'Open Source dans les administrations françaises n'est pas une nouveauté mais là, les bruits de couloir insistent sur l'obligation d'utiliser des standards et formats ouverts. Quid des logiciels ? Pour le moment, aucune obligation ne serait faite mais la tentation sera grande. L'économie de l'Open Source est florissante en France mais l'édition logicielle « propriétaire » ne l'est pas moins. Et surtout, la question se pose sur les standards à imposer, le calendrier, et il faudra tenir compte des développements et des migrations importantes. Nos ministres vont-ils vouloir pousser l'Open Source le plus rapidement possible ou l'imposer à partir de 2017 et refiler la patate chaude au prochain quinquennat ?

François Tonic - Rédacteur en chef
ftonic@programmez.com

Pour fêter dignement le 1^{er} avril, nous vous avons concocté trois possibles « poissions d'avril » : « Mais son-çe réel-lement des canulars ? A vous de le découvrir. Réponse le mois prochain ! »



Editeur : Go-02 sarl, 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. Experts : G. Deruette, K. Alhijres, P. Laporte, F.X. Bois, V. Saluzzo, S. Hadinger, G. Gaulard, G. Rouyer, A. Villechalane, A. Amine Mezri, N. Suter, D. Tran, F. Bouloré, Y. Perio, P. Rieux

Illustration couverture : © istockphoto/pagadesign
Publicité : Régie publicitaire, K-Now sarl. Pour la publicité uniquement : Tél. : 01 41 77 16 03 - diff@programmez.com. Dépôt légal : à parution - Commission paritaire : 0717K78366 ISSN : 1627-0908. Imprimeur : S.A. Corelio Nevada Printing, 30 allée de la recherche, 1070 Bruxelles Belgique. Directeur de la publication : J-C Vaudecrane

Abonnement : Programmez, 17, Chemin des Boulangers, 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

» Le Cobol mal aimé doit être mieux enseigné !

Une étude mondiale auprès de 117 universités montre que le Cobol n'est pas ou très peu enseigné. L'étude menée par Micro Focus (éditeur bien connu d'outils Cobol) est sans appel : 73 % d'entre elles ne proposent aucune formation autour de Cobol et ce, même si elles sont nombreuses à y voir un intérêt à le faire ! Paradoxe ! D'autre part, 54 % des universités estiment que la demande en développeurs Cobol va croître, au pire restera stable, dans les 10 ans. Pour un langage que l'on dit mort depuis 15 ans, ce n'est pas mal ! Cobol demeure le langage de référence dans un nombre très important de systèmes critiques et historiques. Et leur maintenance pose de plus en plus de soucis ! Il y a aussi un problème de sensibilisation des étudiants qui voient Cobol comme archaïque et peu cool.

» OpenJDK dévoile une nouvelle spécification pour JNI.

Pour rappel JNI est la Java Native Interface. L'objectif de la nouvelle spécification (JEP 178) : améliorer le support des bibliothèques natives avec

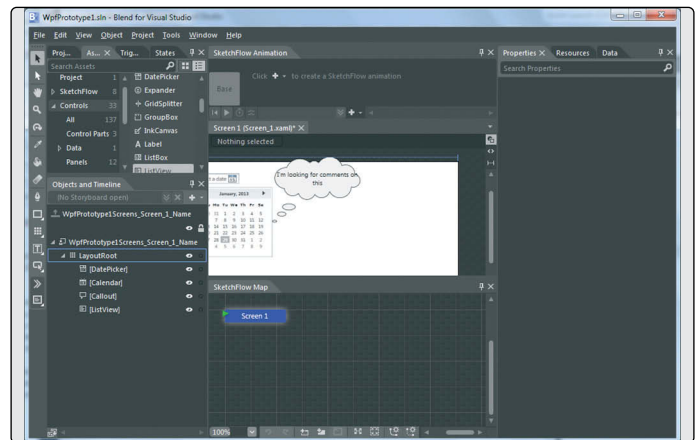
des liens statiques. Ainsi, il sera possible dans un même binaire exécutable d'avoir le code Java et le code applicatif natif, sans recourir à des bibliothèques natives partagées. Le groupe précise que le code Java existant ne change pas (hormis le fait d'utiliser une bibliothèque native statique et non dynamique). Mais il s'agit aussi de permettre à une application Java d'utiliser aussi bien des bibliothèques statiques que dynamiques (le choix étant possible). Le but n'est pas, d'après le projet, de préserver une compatibilité totale des sources natives C++ et C en passant du dynamique au statique. Le développeur devra donc faire des choix et cibler entre les deux approches. Pour en savoir plus : <http://openjdk.java.net/jeps/178>

» Visual Studio 2012 aura droit à une nouvelle mise à jour.

En avril, les développeurs pourront installer Update 2 de VS 2012. Les nouveautés concerneront l'agilité, la qualité, les développements métiers et l'expérience utilisateur. Ainsi, VS 2012 va accueillir Blend pour VS 2012 incluant le support WPF, de Silverlight et de

» Philips propose un SDK officiel pour ses ampoules « intelligentes » Hue.

Ce sont des ampoules connectées (via le sans fil) et que l'on peut contrôler grâce à une application mobile. Désormais, un SDK est disponible pour accéder plus facilement aux ampoules, à la configuration, aux réglages de la lumière, etc. Le kit s'appuie sur RESTful et le développeur l'utilise comme un simple service web. On dispose d'un portail internet sur lequel on indique l'adresse de son système Hue. La publication des applications est gratuite. Philips veut créer une communauté autour de ses ampoules. Actuellement, seul le SDK iOS est disponible mais la version Android arrivera dans les prochains mois. Une belle initiative : <http://developers.meethue.com>



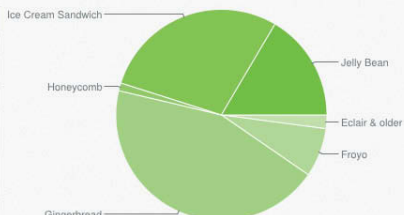
Sketchflow. On notera aussi des améliorations sur la gestion et l'intégration entre le mode debug et CodeMap. Microsoft propose aussi le thème bleu pour ceux qui veulent l'appliquer. Diverses amé-

liorations sont attendues : sur le déploiement des applications mobiles, une nouvelle connexion pour Team Explorer, une timeline pour les applications HTML/javascript.

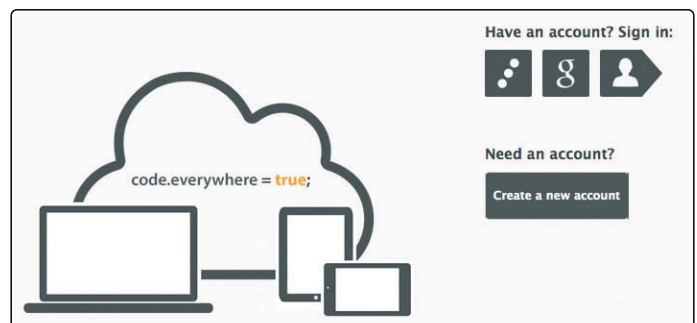
» La fragmentation Android n'est plus ce qu'elle était, ou presque.

Les dernières statistiques de Google sur les versions d'Android déployées montre la montée en puissance des 4.x et surtout de la 4.1 / 4.2 qui représente plus de 16 %. Ice Cream Sandwich pèse 28,6 % mais chute mois après mois. Les dernières versions devraient très rapidement supplanter les premières versions 4.0.x. Mais Gingerbread reste largement majoritaire (Android 2.3.x). Une bonne nouvelle pour les développeurs.

Version	Codename	API	Distribution
1.6	Donut	4	0.2%
2.1	Eclair	7	1.9%
2.2	Froyo	8	7.5%
2.3 - 2.3.2	Gingerbread	9	0.2%
2.3.3 - 2.3.7		10	43.9%
3.1	Honeycomb	12	0.3%
3.2		13	0.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	28.6%
4.1	Jelly Bean	16	14.9%
4.2		17	1.6%



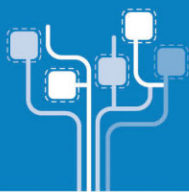
Data collected during a 14-day period ending on March 4, 2013



» La fondation Eclipse distribue Orion 2.0.

Orion est un environnement de développement accessible localement mais aussi sur un simple navigateur internet. La v2 apporte d'importantes améliorations sur les dépendances (supprimées), un JavaScript optimisé, un shell supportant mieux Node.JS interne. On dispose aussi d'un gestionnaire de projets. Pour le moment, il s'agit d'un prototype. Il se complètera d'une gestion plus fine des tâches et d'une intégration des référentiels. L'équipe travaille déjà à la version 3.0 disponible fin juin. Le focus sera mis sur la partie « projets », les composants web, des améliorations autour de Node.JS.

Nouveautés spécifiques au shell : <http://goo.gl/6Qwjg>
Portail Orion : <http://www.eclipse.org/orion/>



phpnet

Hébergement de sites Internet

www.phpnet.org



Backup cloud

Les solutions les moins chères du marché



2Go

Backup free

» Gratuit



250Go

Backup first

» 6€90HT/m



500Go

Backup jump

» 12€90HT/m



1To

Backup full

» 19€90HT/m



3To

Backup max

» 58€90HT/m



Mobilité

Vos fichiers vous suivent partout.
Votre espace comme disque dur externalisé.
Partage de données avec vos amis ou clients.



Simplicité

Outil de gestion simplifié
Création de répertoires par utilisateur.
Gestion des droits d'accès.



Compatibilité

Compatible iOS, Android, Windows, MacOS, Linux, Unix...



Sécurité

Sauvegarde en RAID 50, sur 2 sites différents.
Accès par SFTP, FTPS, SCP, RSYNC, Webdav, SAMBA.

Nos garanties

Confidentialité des données.
Surveillance 24H/24, 7j/7 de notre datacenter 100 % PHPNET.
Redondance des données dans 2 sites distincts.
Support téléphonique non surtaxé, support par mail 24h/24 et 7j/7, sans engagement !
Pour plus de liberté PHPNET n'impose pas de reconduction tacite des services.

 <http://www.phpnet.org/backup>

Code promotionnel

BACKUP1

1 mois GRATUIT
sur toutes les gammes

Offre valable jusqu'au
30 juin 2013



» **Mozilla lance Firefox OS Simulator 3.0, en préversion.** Suite à la disponibilité du système mobile Firefox OS, Mozilla veut améliorer les outils disponibles pour

le développeur. Ainsi la fondation a dévoilé mi-mars une nouvelle version de son simulateur. Les nouveautés les plus importantes sont : simulation de la rotation de l'écran, support basique des API de localisation, validateur du manifest, possibilité de mettre à jour les applications, nouveau moteur de rendu et de Gaia, et possibilité de pousser sur un terminal réel (et connecté par USB) l'application.

Lien : <https://hacks.mozilla.org/2013/03/firefox-os-simulator-previewing-version-3-0/>

Sample	C#	C++	VB	WPF	DirectX	Additional information
Audio Basics	Yes	Yes	Yes	Yes	Yes	Available in 1.6.0
Audio Capture Raw	No	Yes	No	No	No	Available in 1.6.0
Audio Explorer	No	Yes	No	No	Yes	Available in 1.6.0
Basic Interactions	Yes	No	No	No	No	Available in 1.6.0
Color Basics	Yes	Yes	Yes	Yes	Yes	Available in 1.6.0
Depth Basics	Yes	Yes	Yes	Yes	Yes	Available in 1.6.0
Depth	No	Yes	No	No	Yes	Available in 1.6.0
Depth with Color	No	Yes	No	No	Yes	Available in 1.6.0
Face Tracking	Yes	No	No	Yes	No	Available in 1.6.0, Requires Face Tracking
Face Tracking Basics	Yes	No	No	Yes	No	Available in 1.6.0, Requires Face Tracking
Face Tracking Visualization	No	Yes	No	No	Yes	Available in 1.6.0, Requires Face Tracking
Green Screen	Yes	Yes	No	Yes	Yes	Available in 1.6.0
Infrared Basics	Yes	Yes	No	Yes	Yes	Available in 1.6.0
Kinect Explorer	Yes	No	No	Yes	No	Available in 1.6.0
Shape Game	Yes	No	No	Yes	No	Available in 1.6.0
Skeletal Viewer	No	Yes	No	No	Yes	Available in 1.6.0
Skeleton Basics	Yes	Yes	Yes	Yes	Yes	Available in 1.6.0
Slideshow Gestures	Yes	No	No	Yes	No	Available in 1.6.0
Speech Basics	Yes	Yes	Yes	Yes	Yes	Available in 1.6.0
Tic Tac Toe	Yes	No	No	Yes	No	Available in 1.6.0
WPF D3D Interop	Yes	Yes	No	Yes	Yes	Available in 1.6.0
XNA Basics	Yes	No	No	Yes	No	Available in 1.6.0, Requires XNA

» **Microsoft a mis sur la plateforme CodePlex de nombreux codes liés au matériel Kinect (for Windows).**

Ces codes sont là pour aider les développeurs à comprendre le fonctionnement des capteurs, comment traiter les signaux. Le but est double : améliorer les outils et le SDK de Microsoft et aider le développeur. Plusieurs langages sont proposés (selon le code) : C#, C++, VB, WPF, DirectX. Pour les utiliser, vous devez disposer de Kinect pour Windows et du SDK.

Site : <http://kinectforwindows.codeplex.com>

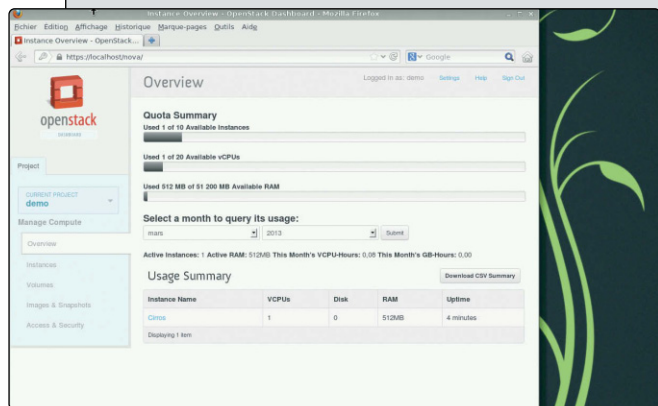


Comment aider les développeurs à développer des applications pour Windows Phone 8 et mieux les promouvoir ?

Un grand programme d'aide a été lancé conjointement par Nokia, Microsoft et l'université d'Aalto : App Campus ! L'objectif est simple : aider les développeurs à créer et à concrétiser leurs applications Windows Phone. App Campus pourra financer (dans la limite de 70 000 euros) le projet ! App Campus doit être comme un accélérateur. Les projets seront examinés et choisis sur plusieurs critères : innovation, que l'application ne soit pas déjà disponible, support des fonctions clés (géolocalisation, in-app, tuiles lives, etc.), un design étudié et le potentiel de l'application auprès des utilisateurs. App Campus, outre l'aide financière, fournit une aide technique, du coaching, le support marketing et technique. Le programme est entièrement en anglais, mais il est possible de localiser les applications dans une autre langue. A vous de jouer ! Pour vous inscrire et soumettre votre projet : <http://www.appcampus.fi>

» **openSUSE fait le plein de nouveautés avec la version 12.3.** Cette édition est disponible depuis le 13 mars dernier. Tout d'abord, la distribution

supporte ARM (v7) et une première version 64 bits sera disponible très rapidement. C'est une annonce importante, surtout avec les incertitudes de support ARM par Microsoft. Cette 12.9 met à jour de nombreux packages : PostgreSQL 9.0 (avec le support de JSON), migration vers MariaDB (MySQL n'est plus un composant par défaut), et



surtout, arrivée d'OpenStack (version Folsom), pour gérer et déployer les infrastructures ! Dans la foulée, VirtualBox, KVM, Qemu ont été remis à niveau. Côté développeur, on dispose des nouvelles versions de Anjuta, QtCreator, Glade, Kate, Momo.

Toutes les nouveautés : <http://en.opensuse.org/Features>

» **Google renforce BigQuery.**

BigQuery est un outil d'analyse pour les données volumineuses. L'éditeur y a inclus plusieurs nouveautés : Big Join (utilise des requêtes de type SQL pour les grands ensembles de données), Big Group Aggregation (permet de regrouper des grands ensembles de données distinctes) et Timestamp (support des données de type Timestamp). BigQuery prend son sens avec des données de plusieurs To. Clairement, BigQuery se rattrape des concepts du framework Hadoop et de MapReduce. Un moyen pour Google de détourner les développeurs d'Hadoop ?

Blog technique : <http://goo.gl/BMPUm>

» **Unity3D, un des moteurs graphiques les plus utilisés, est disponible en version 4.1.**

Cette version supporte la technologie AirPlay d'Apple, un meilleur profiling de mémoire

et diverses améliorations sur les shaders et pour les terminaux mobiles. La partie animation (mecanim) a été améliorée sur les animations 2D. AirPlay permettra de faire du multi écran iOS. Sur la partie OS X, l'outil supporte le centre de jeux d'Apple. Tous les changements : <http://unity3d.com/unity/whats-new/unity-4.1>

» **Vous avez toujours rêvé de modifier du code natif en cours d'exécution ?**

Runtime-Compiled C++ se propose de le faire ! Via l'utilisation du framework, il est possible de modifier en temps réel le code C++, même si le code s'exécute, et les changements sont alors immédiatement répercutés en exécution. Le projet est disponible sur GitHub.

A voir comment évoluera le projet et s'il rencontre un intérêt auprès des développeurs et des éditeurs d'outils. Site : <http://runtime-compiledplusplus.blogspot.fr>

» Windows Azure / Windows Azure Mobile Services subit une importante mise à jour.

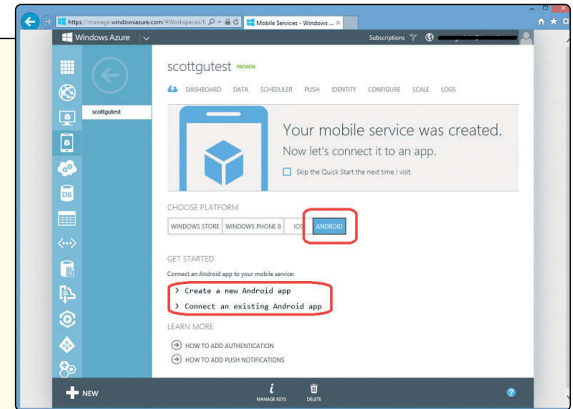
Tout est concerné ou presque : support d'Android, SQL Reporting Services, Active Directory, les services Cloud, Media Service, Service Bus, etc. Mobile Services, qui permet de connecter des applications (mobiles) à des services Windows Azure supporte maintenant

officiellement Android. Un SDK est disponible sur GitHub. Pour intégrer et connecter Mobile Services à Android :

<http://goo.gl/k5nsN>

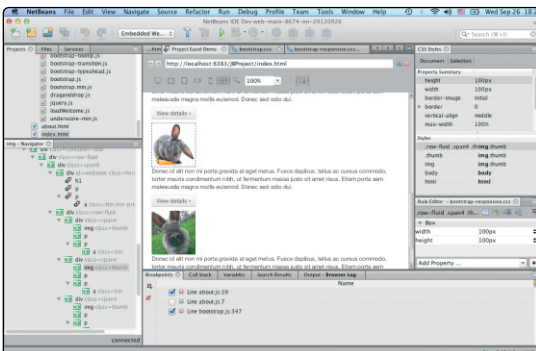
Le SDK iOS s'enrichit lui aussi avec de nouveaux exemples, du code, de la documentation. Le service de monitoring d'Azure est disponible pour les services Cloud, les sites web, les Mobile Services et les machines virtuelles. Tous les détails :

<http://goo.gl/4FzZ0>



» NetBeans 8.0 est annoncé pour fin 2013.

La dernière version importante de la famille 7.x est sortie fin février avec notamment une sérieuse amélioration du développement web (avec HTML 5). Pour le moment, aucune préversion n'est officiellement disponible et le planning de développement et des fonctionnalités n'est pas totalement arrêté. Une des principales annonces sera le support de Java 8 et des principales nouveautés (nous ne savons pas si tout Java 8 sera ou non supporté). Le support de Java FX sera modifié et adapté à Java 8. L'équipe NetBeans veut aussi fixer de



nombreux bugs sur l'ensemble des langages et continuer à améliorer les performances (travail sans fin). Cependant, reste à savoir si cela suffit à changer de numéro... Réponse dans quelques mois...

» Qt Creator 2.7 est disponible en version finale.

L'outil corrige des bugs et améliore notamment le support d'Android et du NDK. Creator 2.7 doit aussi mieux gérer les versions de Qt et la configuration des terminaux. Site : <http://releases.qt-project.org/qtcreator/2.7.0-rc/>. D'autre part, Digia, éditeur de Qt, annonce aussi la prochaine disponibilité de Qt 5.0.2. Au menu : Qt Creator 2.7 par défaut, nouveaux installateurs pour Windows 8, pas de support WebKit 2 dans les packages OpenGL. Des problèmes de performances ont été remontés sur MinGW (chaîne de compilation) mais il faudra sans doute attendre Qt 5.1.



ACCÉLÈRE TON SUCCÈS
reçoit de **20 000 à 70 000€**

➡ une seule adresse : **www.appcampus.fi**

Et pour suivre l'actualité des développeurs Nokia en France :

www.facebook.com/NokiaDevFrance

www.nokiadevblog.fr

@NokiaDev_France

WINDOWS OU LINUX

2 SYSTÈMES D'EXPLOITATION, 1 SEUL PRIX : CHOISISSEZ !



DOMAINES | EMAIL | HÉBERGEMENT | E-COMMERCE | SERVEURS

* Offre 6 mois gratuits : à l'issue des 6 premiers mois, le pack 1&1 Illimité est à son prix habituel de 6,99 € HT/mois (8,36 € TTC/mois). Frais de mise en service : 9,99 € HT (11,95 € TTC). Offre soumise à un engagement de 12 mois. Offres sans durée minimum d'engagement également disponibles. Conditions détaillées sur 1and1.fr. Windows et le logo Windows sont des marques déposées de Microsoft aux Etats-Unis et dans les autres pays. Linux est une marque déposée de Linus Torvalds aux Etats-Unis et dans les autres pays.



HÉBERGEMENT WINDOWS OU LINUX

Chez 1&1, vous êtes libre de choisir entre un hébergement Linux et Windows.

Avec la **dernière version d'ASP.NET 4 et 4.5**, vous bénéficiez d'une palette d'outils innovants, pour des sites Web de qualité professionnelle dans un environnement Windows.

Pour les adeptes de Linux, les packs 1&1 offrent un environnement de développement idéal avec le **nouveau PHP 5.4** et un grand choix d'applications Click & Build : WordPress, Joomla!, Drupal et bien plus encore !



1&1 Illimité Windows



1&1 Illimité Linux

Trafic illimité et bande passante de 100 Mbps

Espace Web **illimité**

500 comptes email (2 Go) et 500 accès FTP

3 noms de domaine inclus pendant toute la durée de votre pack

1&1 Sécurité : géo-redondance, 1&1 SiteLock (en option)

Service expert 6j/7 via hotline non surtaxée et email

et bien plus encore...

NOUVEAU : ASP.NET/.NET Framework **4, 4.5** 
(compatibles avec les versions précédentes)

NOUVEAU : **PHP 5.4**,
Zend Framework, Perl, Python

NOUVEAU : ASP.NET MVC 3 et 4

Protection de vos données :
Restauration de l'espace Web

NOUVEAU : **10** bases de données MS SQL (**1 Go**) 

100 bases de données MySQL 5 (1 Go)

NOUVEAU : Applications en pools isolés

Accès à 65 applications Click & Build
au choix : WordPress, Joomla!, Drupal...

6 MOIS
GRATUITS

puis 6,99 € HT/mois (8,36 € TTC/mois)*

6 MOIS
GRATUITS

puis 6,99 € HT/mois (8,36 € TTC/mois)*

Découvrez toutes nos autres offres sur notre site Web.



SÉCURITÉ MAXIMALE

Vos données sont hébergées en parallèle dans 2 centres de données 1&1 haute performance situés à différents endroits. Le « plus » : sauvegardes automatiques quotidiennes.



SERVICE EXPERT 6j/7

En tant que client 1&1, vous bénéficiez d'une assistance non surtaxée (hotline et email), assurée par des professionnels de l'hébergement.



Les centres de données 1&1 sont alimentés en énergie verte : réduction des émissions de CO₂ de 30 000 tonnes par an.



0970 808 911
(appel non surtaxé)

1and1.fr

Notre sélection de livres

SPÉCIAL IOS/OS X

Les éditions Pearson proposent trois livres de programmation sur iOS, Cocoa et Objective-C, dans la collection « Big Nerd Ranch »

Programmation iOS

Dès la première page, on plonge les mains dans la plomberie avec Xcode, le premier projet, puis on poursuit sur les bases d'Objective-C, le langage de référence d'Apple depuis 15 ans. Ensuite, les chapitres abordent point par point les différents frameworks d'iOS (Core Location, MapKit, Core Data, etc.). Les auteurs n'oublient pas non plus l'architecture des applications iOS, les fonctions de base, les classes.

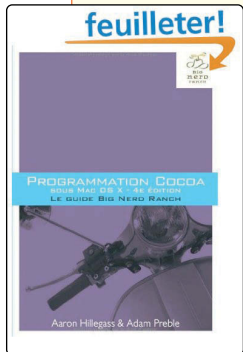
Tous les codes sont compatibles avec iOS 6, Xcode 4.5 ; OS X 10.8. Seul défaut, le prix : 45 €.

Programmation Cocoa sous OS X 4e édition

Cocoa est l'ensemble des technologies, frameworks et API des systèmes Apple (iOS et OS X). Ce livre se concentre sur OS X. Vous verrez toute la richesse des frameworks (même si tout n'est pas abordé). Les auteurs n'oublient pas la programmation concurrente, ni OpenGL et le sandboxing, désormais obligatoire dans les applications OS X. 39 €

Programmation Objective-C

ce complément est conseillé aux deux autres ouvrages. Qu'est-ce que Objective-C hormis d'être un héritier direct du langage C ? L'auteur, expert Next et Apple, propose une solide base pour comprendre et apprendre Objective-C avec les variables, classes, types, les fonctions. 29 €

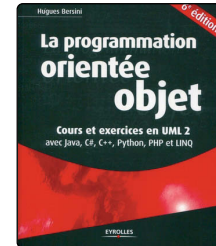


interfaces. Vous y trouverez les bonnes pratiques, des conseils pour choisir le graphisme, l'apparence, créer la navigation, etc. L'ouvrage est illustré par des cas pratiques de conception d'interface pour lesquels les auteurs ont appliqué la démarche ergonomique. Ils témoignent concrètement du processus mis en œuvre pour réaliser une interface intuitive.

HUGUES BERSINI

La programmation orientée objet 6e édition

Eyrolles



Une des références des livres de programmation revient dans une 6e édition ! L'auteur décortique l'ensemble des mécanismes de la programmation objet

(classes et objets, interactions entre classes, envois de messages, encapsulation, héritage, polymorphisme, interface, multithreading, sauvegarde des objets, programmation distribuée, modélisation...). Il les illustre d'exemples empruntés aux technologies les plus populaires : Java et C#, C++, Python, PHP 5, UML 2, LINQ, mais aussi les services web, Corba, les bases de données objet, différentes manières de résoudre la mise en correspondance relationnel/objet, dont le langage innovant de requête objet Linq et enfin les design patterns.

BEN FRAIN

Responsive Web Design avec HTML 5 et CSS3

Pearson



Ce mois-ci, le Responsive Design est un des thèmes abordés par Programmez !. Cet ouvrage vous permettra d'aller encore plus loin. Pourquoi et comment faire du Responsive interface /

design ? Qu'est-ce que cela signifie réellement quand on travaille en HTML et CSS3 ? L'auteur aborde particulièrement les fonctions avancées pour traiter la typographie, les animations, les ombres, etc. Et surtout, être le plus universel possible dans son approche.

LUC VAN LANCKER jQuery mobile

éditions ENI

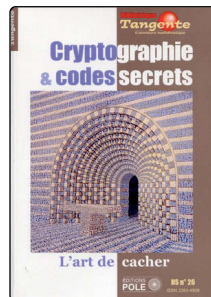


jQuery est un des frameworks les plus utilisés. Il possède une déclinaison spécifique pour les terminaux mobiles : jQuery Mobile. Les premiers chapitres parlent des caractéristiques du Web mobile, puis l'auteur dresse le panorama des éléments incontournables du HTML5, des CSS3 et des API JavaScript spécifiques aux applications mobiles. Bien entendu, l'utilisation de jQuery Mobile n'est pas oubliée : pages, navigations, fenêtres, formulaires, objets d'interface, etc.

COLLECTIF, Cryptographie et codes secrets

Pole

Depuis longtemps, les Hommes utilisent toute sorte de techniques pour cacher les messages. Dans ce livre original, les auteurs



évoquent une histoire de la cryptographie et du chiffrement, avec de nombreux exemples jusqu'aux techniques les plus récentes.

Aujourd'hui, la cryptologie est partout et elle évolue sans cesse pour contrer les attaques et tenter de les devancer. A lire !

COLLECTIF Ergonomie des interfaces

Dunod

Une interface, sa réussite ou son échec, vient en grande partie de son ergonomie.



Cette 5e édition intègre les interfaces mobiles ainsi que les recommandations pour le e-commerce et fait la synthèse des différentes règles à suivre en matière d'ergonomie des

» Nvidia annonce de nouvelles cartes graphiques basées sur l'architecture Kepler.

Ces cartes de la gamme Quadro se dédient aux stations de travail et aux utilisateurs très exigeants sur les performances. Cette gamme propose désormais :

- NVIDIA Quadro K4000 : Dotée de 3 Go de mémoire intégrée, du support multi-moniteurs et de la fonction stéréo dans une configuration mono-slot.
- NVIDIA Quadro K2000 : une carte milieu

de gamme. Equipée de 2 Go de mémoire intégrée pour exploiter de grands modèles et du support multi-moniteurs pour améliorer la productivité.

- NVIDIA Quadro K2000D : une variante de la Quadro K2000 dotée d'un support natif pour deux connecteurs pour écrans Dual Link DVI destinés à un interfaçage avec des dispositifs d'affichage d'imagerie médicale très haute résolution.
- NVIDIA Quadro K600 : une carte d'entrée de gamme. Dotée d'1 Go de mémoire

intégrée, conçue pour un faible encombrement et un maximum de flexibilité d'utilisation.

Côté tarif, l'entrée se situe

à 169 € et monte jusqu'à 959 €

(modèle K4000).

Attention : ces cartes nécessitent des pilotes et des applications compatibles et certifiés.



» XPS 13 Developer Edition : Dell propose un ultrabook pour développeur (?)

Dell met en vente son XPS 13 Developer Edition fonctionnant sous Ubuntu. Le projet était connu sous le nom « Spoutnik ». La démarche officielle est simple : savoir ce qu'attendent les développeurs dans un ordinateur portable. Le constructeur met en avant les performances, la mobilité (heureusement pour un portable) et les fonctions Cloud. Il dispose d'une distribution Ubuntu 12.04 LTS, d'un outil pour créer un micro-Cloud local (Cloud launcher) et de divers outils de profiling. Cette suite d'outils est disponible sur la page du projet sur github. On peut rapidement mettre en place des environnements de développements Ruby, Android.

La machine possède un Core i7, 8 Go de mémoire vive, une GPU Intel HD 4000, un écran haute définition 1080p, un disque SSD. Les performances semblent très bonnes mais est-ce que cela suffit à lui donner le nom « édition développeur ». Il faudrait des services étendus dédiés aux développeurs, une panoplie d'outils de développement pour chaque type de programmation, etc. Ce qui n'est pas le cas aujourd'hui. Pourquoi ne pas imaginer un XPS 13 Visual Studio edition par exemple avec un programme développeur MSDN inclus ? La machine est belle et puissante, mais reste à véritablement étoffer la partie développeur. Elle est vendue un peu moins de 1200 €.

Github :

<https://github.com/sputnik>



LE NUMERIQUE ON EN PARLE BEAUCOUP NOUS, ON L'INVENTE

EPITECH.
L'ÉCOLE DE L'INNOVATION ET DE
L'EXPERTISE INFORMATIQUE

Iconoclaste ou surprenante, d'un dynamisme insoupçonné quand il faut lancer un projet ou une entreprise, telle est la génération numérique et son école, Epitech. Depuis 15 ans, par une pédagogie qui assume sa différence, par une confiance dans nos étudiants et dans leurs projets audacieux, nous apportons aux entreprises ce nouveau vent de passion indispensable. La génération numérique transforme chaque jour le monde. Son école a un nom, Epitech, l'école de l'innovation informatique.

www.epitech.eu

Établissement d'enseignement supérieur privé



En avril

Dev King 2013 : 12 – 14 avril, la finale !

Devenez le roi des développeurs en codant une application Windows 8 – Windows



Phone 8 / HTML 5 et en utilisant des données ouvertes. Le thème central est l'application citoyenne. La grande finale aura lieu du 12 au 14 avril 2013 au Under de l'école Epitech. Le concours est doté de 15 000 € dont 8 000 € pour l'équipe gagnante, avec en prime la mise en avant de l'application dans le Windows Store. D'autres prix seront distribués : Nokia, PopWings, HTML 5. Information : <http://bemyapp.com/devkings2013/>

agenda

► AVRIL

- Du 02 avril au 06 avril 2013, Espace Cap 15, au pied de la Tour Eiffel, **Symfony Live Paris 2013** (voir détails ci-dessus) <http://paris2013.live.symfony.com/>
- Du 03 avril au 04 avril 2013, CNIT Paris la Défense, **Big Data Paris 2013**. 1500 participants, 60 exposants et 100 speakers sont attendus pour deux journées de conférences, d'exposition et retours d'expérience BigData. <http://www.bigdataparis.com>
- 11 avril 2013, IBM Client Center, Bois-Colombes, **Scrum Day 2013**, troisième édition annuelle du Scrum Day. 500 personnes sont attendues à cet événement incontournable pour la communauté Agile et pourront assister aux conférences et ateliers animés par des experts reconnus. <http://www.scrumday.fr/>
- Du 28 au 29 mai 2013, CNIT de Paris La Défense **Salon Solutions Linux, Libres et Open Source 2013**. La 15e édition du salon Solutions Linux fait peau neuve. Renommé « Solutions Linux, Libres & Open Source ». Avec près de 6 000 visiteurs, la dynamique du salon fait figure de référence pour rassembler l'écosystème et les acteurs des secteurs publics, privés et associatifs. <http://www.solutionslinux.fr>

SymfonyLive 2013 : du 2 au 6 avril

Événement incontournable du monde web, SensioLabs organise sa grande conférence technique et stratégique : SymfonyLabs. Deux journées de formations seront proposées les 2 et 3 avril, avec une grande journée hacking, le samedi 6 avril. SymfonyLive est l'occasion de découvrir les dernières nouveautés du framework et des outils, rencontrer les développeurs, les partenaires et voir surtout les entreprises utilisant concrètement Symfony. Les conférences se dérouleront les 4 et 5 avril. Si vous ne connaissez pas encore, SensioLabs possède sa propre chaîne YouTube avec de nombreuses vidéos des sessions, lien : <http://www.youtube.com/sensiolabs> Pour en savoir plus : <http://paris2013.live.symfony.com>

Communautés

Code d'Armor : les développeurs bretons !

Une association dans la région de Lannion s'est créée en 2012 pour construire et animer une communauté de développeurs de la région. L'animation se fait autour d'événements techniques comme les soirées Google



Cloud, Git. Ces soirées sont mensuelles et animées par un conférencier expert. Site : <http://codedarmor.fr>



Fier d'être développeur : l'association est lancée

« Fier d'être dev » avait été lancée en 2012 mais il a fallu un an pour formaliser l'association, et les premières initiatives. Les objectifs de ce mouvement sont les suivants :

1. Promouvoir le métier de développeur de logiciels,
2. Expliquer la valeur de ce métier alliant rigueur scientifique et force de créativité,

3. Communiquer la noblesse du choix de faire carrière en tant que développeur,
4. Valoriser l'impact de l'expérience sur l'équation économique des développements logiciels,

Encourager le respect mutuel entre les développeurs, indépendamment des plateformes et technologies utilisées.

Nous encourageons cette initiative pour mieux faire connaître et reconnaître le métier de développeur en France !

Site : <http://fierdetredeveloppeur.org>

Une communauté française autour de Unity

Vous développez avec Unity, vous cherchez des groupes d'utilisateurs, des informations, une aide ? Unity France est là pour ça. Le portail officiel fournit de nombreuses informations autour d'Unity mais aussi et surtout toute l'actualité en France : réunions et conférences des groupes utilisateurs, des tutoriaux, des forums. Vous trouverez aussi de nombreuses vidéos, dont celle du Montpellier Unity User Group.

Site : <http://www.unity3d-france.com>



DANS LA JUNGLE DU CLOUD, MIEUX VAUT CHOISIR LE BON PARTENAIRE.



Aruba Cloud, les solutions IaaS qui répondent à chacun de vos besoins.

CLOUD COMPUTING

- Créez, activez et gérez vos VM.
- Choisissez parmi nos 3 hyperviseurs.
- Maîtrisez et planifiez vos ressources CPU, RAM et espace disque.
- Uptime 99,95% garanti par SLA.

CLOUD OBJECT STORAGE

- Créez vos espaces et stockez vos données en toute sécurité.
- Une solution qui s'adapte à vos besoins : Pay as you Go, ou formule prête à l'emploi.
- Bande passante et requêtes illimitées.

LE CLOUD PAR ARUBA

- Ubiquité : choisissez votre pays et datacenter.
- Interopérabilité : API et connecteurs.
- Agnosticisme : choisissez votre hyperviseur.
- Scalabilité : étendez votre infrastructure à l'infini.
- Transparence : pas de coûts d'activation, ni coût caché.
- Pay as you Go : ne payez que ce que vous consommez.

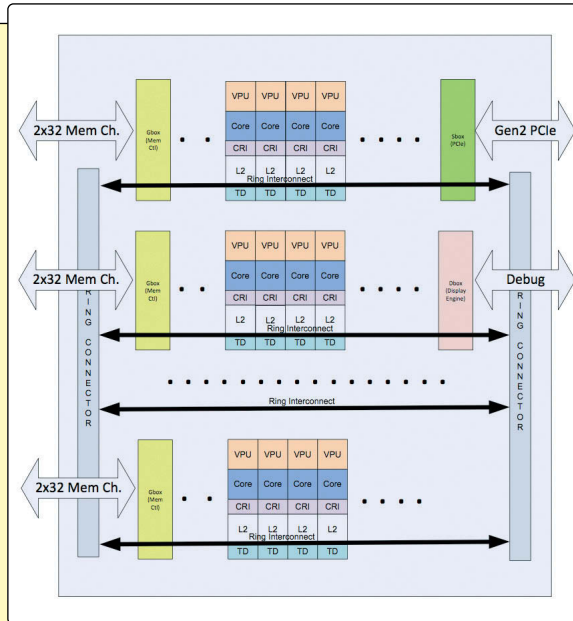
Aruba, le bon partenaire pour bénéficier de la puissance d'un acteur majeur qui considère que chaque client, dans chaque pays, est unique. **MY COUNTRY. MY CLOUD.**



arubacloud.fr | TÉL : 0810 710 300
(COÛT D'UN APPEL LOCAL)

» Intel mise sur les développeurs pour les processeurs Xeon Phi.

Phi a mis deux ans à être disponible. Cette gamme de processeurs vise le marché des HPC, les machines et clusters à très haute performance de calculs. Phi repose sur une architecture x86 classique mais dispose de 60 cœurs (minimum), en exécution 64 bits ! Et l'ambition autour de ce processeur nouvelle génération est grande : arrivée au exaflop (2018). Dans leurs architectures, ils ressemblent aux puces Tesla de Nvidia, mêlant cœur processeur et unité vectorielle. On ne retrouvera pas un processeur Phi dans un PC normal, peut être dans une station de travail très haut de gamme. Mais pour pouvoir exploiter ce type d'architecture, il faut des applications optimisées et massivement parallèles. C'est pour cela qu'Intel et Microsigma ont organisé un séminaire tech-



nique le 19 mars dernier à Paris. Le modèle de développement repose sur le Many Integrated Core (ou Intel MIC), une architecture de programmation. Il faut disposer de Composer XE 2013 ou d'un outil équivalent. Pour en savoir plus (guide du développeur Phi) : <http://goo.gl/iOy3b>

» Les processeurs ARM 64 bits vont pouvoir virtualiser sans difficulté.

En 2012, un premier portable de l'hyperviseur KVM avait été réalisé pour les ARM 32 bits. Désormais, c'est au tour des processeurs 64 bits de pouvoir utiliser KVM comme hyperviseur. C'est le fondement qui a modifié et adapté l'outil. Comme le précise ARM, le code est très similaire entre les versions 32 et 64, selon les développeurs du projet (XEN dit l'exact contraire). Il est prévu que dans les prochains mois, les modifications seraient disponibles par défaut dans le noyau Linux 3.10. Et pour le moment les ARM 64 sont peu supportés par les systèmes serveurs. Xen avait aussi annoncé la disponibilité de son hyperviseur sur ARM v5 - v7 (32 bits). Initialement, ce portage était dédié aux terminaux mobiles. Xen sur les serveurs Cortex A15 est un projet en cours ainsi que le portage sur les ARM v8 (64 bits).

» Raspberry Pi peut servir à tout et n'importe quoi. Un des usages les plus extrêmes est de vouloir le transformer en Macberry Pi pour le transformer en serveur Time Machine, le système de sauvegarde automatique d'OS X. Le but est de pouvoir avoir un environnement de sauvegarde par-

tagé visible par tous les Mac. Il faut configurer le système et bien entendu y connecter un solide disque dur et connecter la machine en Ethernet. Avec un peu de patience, vous disposez d'une « Time Capsule like »...

Lien : <http://goo.gl/zNFRN>

» Windows RT ne décolle pas chez les constructeurs. C'est le constat que l'on peut faire après l'annonce radicale de Samsung de ne plus vendre de tablettes

(modèle Ativ Tab) utilisant Windows RT dans plusieurs pays européens. Les ventes de ces tablettes semblent atones et très limitées. Samsung avait déjà annoncé l'arrêt aux Etats-Unis. Pour Windows RT, c'est le signe qu'il y a un problème sérieux sur le positionnement du système et des matériels face à Windows 8. Microsoft doit rapidement recadrer son approche ARM. HP a même lancé une tablette, bas de gamme, sur Android et non Windows...

» Une imprimante 3D avec un scanner intégré chez soi ?

C'est tout le défi de MakerBot qui va proposer dans quelques mois Digital Desktop 3D Scanner. Il s'agit d'un scanner 3D pour capturer un objet en volume, en produire un modèle numérique le plus précis possible puis de l'imprimer en « 3D », donc en volu-



me. MakeBot propose déjà une imprimante 3D, la Replicator 2 à 2 799 \$ (sans le filament et un délai de 2 mois). On pourra donc, sans modélisation personnelle, reproduire des objets...

Site : <http://store.makerbot.com/digitizer.html>

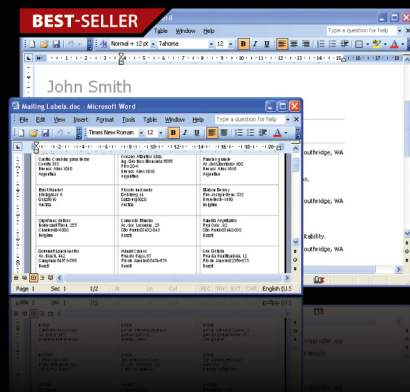
» Microsoft sort un clavier et un souris (gamme Wedge) dédiés aux tablettes Windows.

Le clavier se veut minimaliste par sa taille mais aussi par sa composition, sans pavé numérique ou touches de fonctions séparées. Mais, et c'est une bonne nouvelle, Wedge est compatible avec les iPad et les tablettes Android... Le tout est sans fil (bluetooth). Le couvercle peut servir de support à la tablette. Ce qui rend tout de même l'ensemble un peu encombrant.

La souris ressemble à un pavé tactile et non à une souris, un peu comme le propose Apple avec son Magic Trackpad.

Site : <http://www.microsoft.com/hardware/fr-fr/mice>



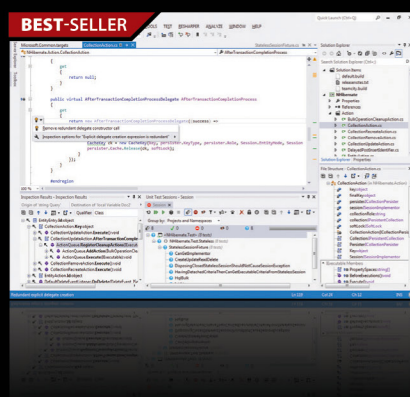


Aspose.Words for .NET à partir de € 783



Lisez, modifiez et écrivez des documents Word sans Microsoft Word.

- Création de documents, manipulation du contenu/formatage, puissante capacité de fusion de courrier et exportation en DOC/HTML
- Accès détaillé à tous les éléments d'un document par programmation
- Support les formats de fichiers: DOC, DOCX, WordprocessingML, RTF, HTML, OOXML, OpenDocument, PDF, XPS, EMF et EPUB

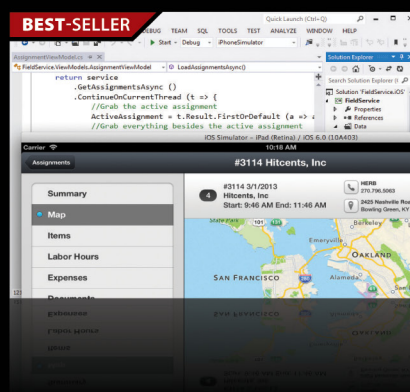


ReSharper à partir de € 308



Outil de développement Microsoft Visual Studio 2005-2012 pour améliorer la productivité.

- Analyse de qualité du code à la volée dans C#, VB.NET, XAML, ASP.NET, MVC, JS, CSS, HTML, XML
- Accès ultrarapide à tout fichier, type ou membre dans votre base de code
- Signale les problèmes de code et propose des solutions rapides de correction
- Assistants de codage incluant IntelliSense et transformations instantanées du code
- Refactorisations de la solution pour changer la base de code en toute sécurité



Xamarin.iOS et Xamarin.Android à partir de € 783



Créez des applis iOS et Android 100 % natives avec Visual Studio.

- Écrivez 100 % de votre code en C#
- Partagez jusqu'à 90 % de votre code entre iOS, Android et Windows Phone
- Inclut le superbe et nouveau IDE Xamarin Studio
- Supporte App Store et Enterprise Distribution
- Choisissez des composants d'appli pré-construits pour accélérer le développement



GdPicture.NET à partir de € 3 136



Outils complets d'imagerie documentaire et de gestion pour les développeurs.

- Capturez, traitez, créez, affichez, modifiez, annotez, composez, fractionnez, fusionnez et imprimez des documents depuis vos applications Windows et Web
- Lisez, écrivez et convertissez les images vectorielles et raster en plus de 90 formats, dont PDF, PDF/A, TIFF, GIF, JPEG, PNG, JBIG2, WMF, BMP, WBMP, ICO, PCX, PNM, XPM, JPEG 2000, HDR, PSD, TGA, PICT, EXR, DDS, PPM, SGI, PBM, PGM, PFM, XBM, IFF et le format graphique RAW

© 1996-2013 ComponentSource. Tous droits réservés. Tous les prix sont corrects au moment de la presse. Prix en ligne mai différentes de celles décrites en raison de fluctuations quotidiennes et remises en ligne.

Siège social en Europe
ComponentSource
30 Greyfriars Road
Reading
Berkshire
RG1 1PE
Royaume-Uni

Siège social aux États-Unis
ComponentSource
650 Claremore Prof Way
Suite 100
Woodstock
GA 30188-5188
États-Unis

Siège social au Japon
ComponentSource
3F Kojimachi Square Bldg
3-3 Kojimachi Chiyoda-ku
Tokyo
Japan
102-0083

Numero vert:
0800 90 92 62
www.componentsource.com

Nous acceptons les bons de commande. Contactez-nous pour demander un compte de crédit.



« Ayez toujours faim de technologie ! »

Geoffrey Parquet est notre développeur du mois. A 24 ans, il termine sa 5^e année de cycle supérieur à l'Exia.Cesi d'Arras. Passionné d'informatique depuis l'âge de 7 ans, il réalise maintenant son rêve : travailler dans les nouvelles technologies et développer des logiciels. Malgré un parcours scolaire atypique, sa volonté a fini par payer !
Portrait.



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

Je n'ai que 24 ans et pourtant l'informatique est pour moi ce que la potion magique est à Obélix ! Je suis tombé dedans lorsque j'avais 7 ans, avec mon premier Windows 95. J'ai un cursus atypique sur le plan des études : BEP électronique suivi d'un BAC Professionnel système électronique numérique et je ne savais toujours pas ce que je voulais faire à l'âge de 16 ans. Cependant j'étais obsédé par la conception de sites internet et de petites applications en Java que je m'obstinais à créer. J'ai eu une illumination en deuxième années de bac pro et je me suis

dit que je devais aller en BTS informatique de gestion, option développeur d'application, pour continuer ce que j'aimais faire : la création et l'informatique ! Bricolant quelques sites et aimant l'analyse et la conception, je me suis dit, malgré mes résultats catastrophiques, que je me devais de continuer. Voilà comment je suis entré dans le monde merveilleux du développement ! Je ne voulais pas m'arrêter en si bon chemin ! J'ai donc intégré un cycle supérieur d'ingénierie informatique à l'Exia.Cesi d'Arras où je termine ma 5^e année. J'ai participé à de nombreux stages dans des univers complètement différents, Web (JavaScript, PHP, AJAX), Mobile (Android et Windows Phone), client/serveur (.net C# et Java), WebServices (WCF)..... En bref, je suis vraiment un touche-à-tout ! Je suis maintenant en stage de longue durée chez OCEANE Consulting Nord pour une période de 6 mois avec normalement un CDI à la clé. Je dois actuellement réaliser deux missions très importantes au sein d'Océane. La première est l'interconnexion de leurs portails SharePoint à leur site internet, et la seconde, la création

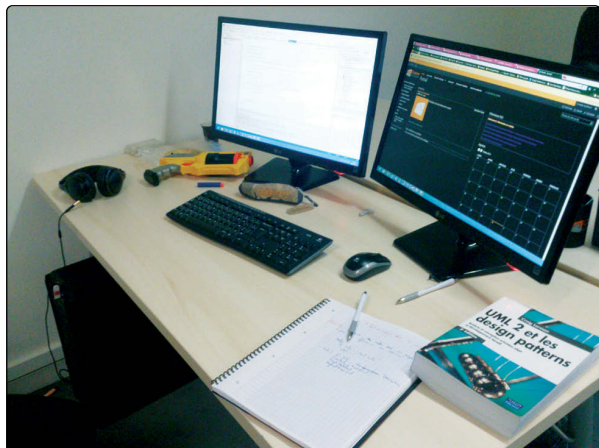
d'une application Windows phone 8 qui servira de réseau social en interne. Voici mon poste de travail chez OCEANE Consulting Nord, je peux dire que j'ai été gâté ;)

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

Je ne vais pas reprendre ce que tout le monde dit, mais lorsque l'on est développeur et que l'on aime créer, la technique ne se perd jamais, tous les développeurs ont soif de connaissance et de reconnaissance ! En mettant en avant leurs compétences techniques au service des autres. C'est vrai que la veille techno et la passion de l'informatique y sont pour beaucoup ;) ! En tous cas, je n'ai jamais perdu cet engouement pour la technique car il y a de nouvelles choses à découvrir chaque jour. Je suis réellement fier d'être développeur ! Je ne pense pas un jour quitter ce monde de découverte et de technique auquel je tiens tant. Ma plus grande crainte est de ne pas en voir autant que je le souhaiterais à cause de l'évolution rapide de ce domaine.

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

Je suis depuis très longtemps un grand



Geek dans l'âme, toujours à l'affût du gadget inutile donc indispensable, ce qui fait toujours bien rire mon entourage d'ailleurs ! Mon futur achat sera un pistolet nerf conseillé par mon futur tuteur de stage afin de prendre de bonnes pauses dignes de ce nom ! C'est pour vous dire à quel point je tiens à mon côté Geek ;)

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

A l'heure actuelle où nous souffrons « d'info-bésité », il est très difficile de suivre l'actualité de tous les langages que nous utilisons. C'est pour moi la partie la plus difficile du métier : toujours rester à jour. Cela en devient même frustrant ! J'ai toujours l'impression de ne pas pouvoir explorer entièrement toutes les facettes et les possibilités qui me sont offertes lors de la découverte d'une nouvelle technologie. Je pense que le travail du développeur change de plus en

plus, il doit être à la fois multifonction et touche-à-tout. Mais il doit avant tout avoir une vision globale du SI, un sens inné de la rédaction et de la pédagogie afin de pouvoir retransmettre son savoir. C'est vrai que l'on ressent parfois une certaine pression dans ce métier, car on veut toujours bien faire et montrer que l'on est capable de faire vite et bien, malheureusement il faut savoir aussi souffler, se poser pour analyser mieux les choses et prendre un maximum de recul pour être efficace, c'est ça aussi être un bon développeur !

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

Je suis un grand cinéophile, j'adore aller au cinéma ! Ce que j'aime par-dessus tout, c'est

“ Inutile... donc totalement indispensable ”

la lecture : ça me permet d'avoir du temps rien qu'à moi, pour m'évader, m'isoler et respirer. En revanche, j'ai énormément de mal à m'arrêter de travailler sur les différents projets en cours, ce qui ne fait pas toujours bon ménage avec la vie de couple.

J'ai heureusement une compagne et des parents qui me soutiennent dans tout ce que je fais et je les en remercie !

Peux-tu nous présenter ton quotidien en quelques mots ?

Très peu de sommeil et beaucoup de travail... si l'on peut appeler cela du travail étant donné que c'est une passion ;)

Plus sérieusement, actuellement je travaille sur plusieurs projets pour l'école en l'occurrence, un approfondissement technique sur la mobilité et le monde d'Android, et un autre projet sur une application Windows phone qui va permettre aux étudiants de recevoir leurs notes en temps réel par le biais de notifications push.

Comment vois-tu ton job évoluer ?

Etant dans ma dernière année d'étude et un grand passionné des technologies Java et .NET, je me donne 2 ans pour être expert technique sur une de ces plateformes, tout en ayant quand même une préférence pour le .NET, je souhaite malgré tout garder les deux casquettes.

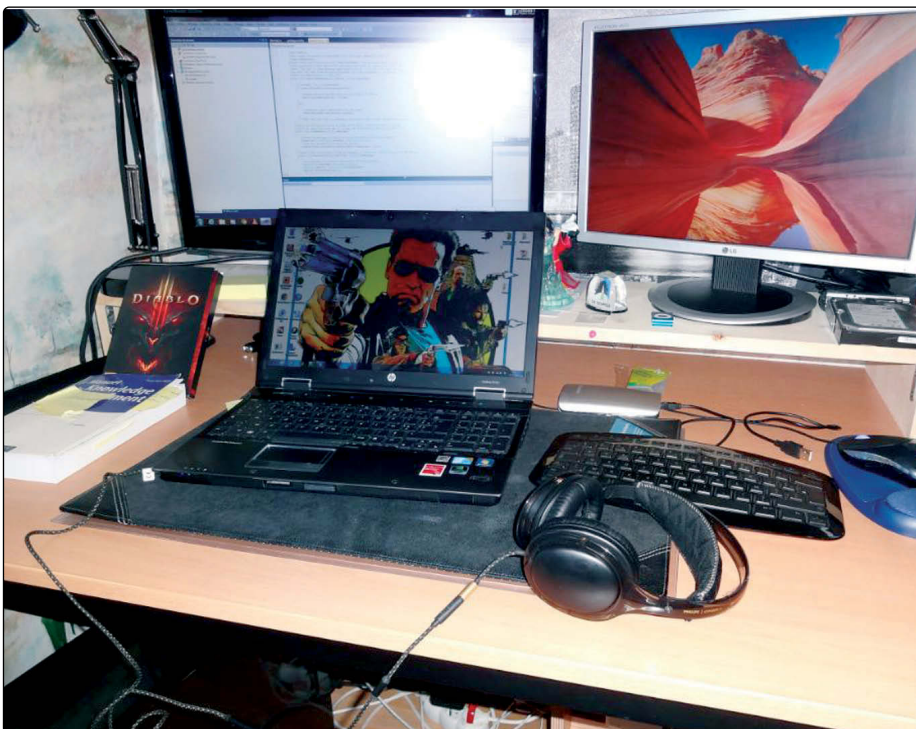
En effet, à l'heure actuelle ce sont pour moi les deux technologies les plus demandées et les plus utilisées. La consécration ultime sera d'être, dans une vision très lointaine, architecte logiciel, ce qui me permettrait de me pencher encore plus dans l'analyse, ce que je préfère !

Des conseils aux étudiants et développeurs qui nous lisent ?

Gardez votre faim ! Ayez toujours envie de voir plus haut, n'ayez pas peur de vous lancer dans des études si votre motivation est plus forte que tout, vous y arriverez !

Ne vous découragez surtout pas ! Restez humble, ayez toujours une envie d'apprendre en partageant votre savoir-faire avec d'autres personnes sans vouloir imposer votre choix et votre vision des choses.

#



Mon bureau

Mon environnement de travail se compose d'un PC portable HP EliteBook i5, environnement Multi Boot Windows 7 et Ubuntu 12.04. Les IDE que j'utilise le plus sont : Visual Studio 2010 Ultimate, NetBeans 7.1 ainsi que Eclipse « Juno » 4.2 pour le développement Android. Je travaille toujours en musique, il est très rare que j'arrive à travailler sans ! Ça m'aide énormément à réfléchir. Vous pouvez remarquer mon acteur préféré en fond d'écran sur le PC portable, je vous laisse deviner ;)

Faut-il passer à Zend Framework 2 ?

La réponse à la problématique que nous nous posons, à savoir «Faut-il passer à Zend Framework 2 ?» pourrait être évidente : Zend Framework 1 ne sera plus supporté fin 2013 (ou début 2014 selon que l'on voit le verre à moitié vide ou à moitié plein). Mais avant toute chose, pourquoi une nouvelle version majeure ?

La raison principale tient au fait que Zend Framework 1 n'évolue plus. Sa dernière version, la 1.12, ne corrige que quelques bugs mineurs. Zend Framework 1.12 étant devenue extrêmement stable, c'est l'occasion de développer une nouvelle version diront les mêmes optimistes voyant ce fameux verre à moitié plein. Cela ne fait aucun doute.

Développer une nouvelle version a certainement une deuxième raison propre à beaucoup de projets informatiques.

À force d'ajouter des fonctionnalités permettant d'enrichir le produit, sa clarté s'est doucement mais sûrement détériorée, à tel point qu'ouvrir le capot de cette technologie bricolée fait frémir les plus déterminés.

Zend Framework était sans doute arrivé au point où il fallait choisir entre ajouter des rustines à chaque amélioration désirée (et sans doute mourir à petit feu) et revoir ses bases en profondeur afin d'être à la pointe des dernières technologies, comme lors des tous premiers jours.

En particulier, Zend Framework 2 intègre pleinement les nouveautés de PHP 5.3 contrairement à la 1.11 qui n'en exploitait pas tout le potentiel. Zend Framework 2 est par ailleurs compatible avec PHP 5.4.

Zend a donc fait ce choix de refonte, lourd de conséquences, avec notamment la perte de compatibilité ascendante avec la version précédente.

Le principal contributeur de Zend Framework 2, Matthew Weier O'Phinney (presque un tiers des commits lui sont dus), annonce la couleur : « *L'objectif premier de Zend Framework 2.0 est de faire un produit plus cohérent, bien documenté, permettant l'amélioration de la productivité des développeurs et des performances* ». Avec ses 13 000 commits pour environ 600 contributeurs, découvrons donc le dernier protégé de la société fondée par les créateurs de PHP 3, ce dernier ayant alors révolutionné le monde PHP.

> Quelques atouts de la nouvelle version

Fini la magie !

En PHP, les méthodes magiques sont ces méthodes commençant par « `__` » qui sont lancées alors qu'aucun appel explicite ne leur est fait. Par exemple :

```
$monObjet->attribut = 42;
```

Ceci appelle la méthode `__set()` dans laquelle on peut avoir mis ce que l'on veut. Ces pratiques font parfois gagner du temps. Mais cela complique énormément la compréhension et les corrections.

Zend a considéré qu'utiliser des méthodes magiques était une mauvaise pratique, et en a drastiquement limité leur utilisation dans son Framework.

L'autoloading

L'auto-chargement des classes est la fonctionnalité de PHP 5 qui permet de charger une classe sans que le développeur ait besoin d'écrire une longue liste de `required_once()` définissant les classes dont dépend la classe courante. Cette astuce était déjà utilisée dans la première version de Zend Framework, grâce à sa classe `Zend_Loader_Autoloader`. Mais cette fois, Zend Framework 2 intègre entièrement cette nouveauté et rend son utilisation systématique. De plus, l'autoloader est construit de manière à respecter la norme PSR-0, la dernière en vigueur.

Ainsi, les espaces de nom (ou namespaces) ne sont plus séparés par des « `_` » mais par des `DIRECTORY_SEPARATOR` (des « `/` »). Par exemple, le code ci-dessous représente le début du fichier du contrôleur du module application :

```
namespace Application\Controller;

use Zend\Mvc\Controller\AbstractActionController;
use Zend\View\Model\ViewModel;

class IndexController extends AbstractActionController
{
```

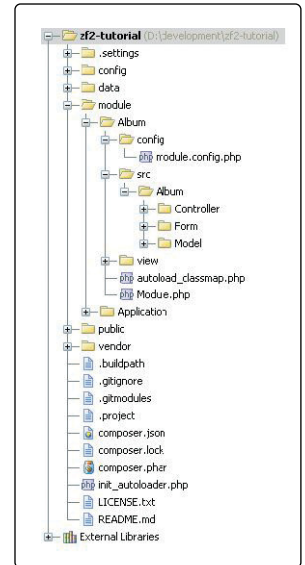
Nouvelle gestion des modules

L'architecture de Zend Framework 2 est très similaire à celle de Zend Framework 1 (organisation MVC : Modèle - Vue - Contrôleur). Mais elle a évolué de façon à permettre la réutilisation d'un module moyennant quelques configurations toutes regroupées dans le dossier « `config` ».

Ceci permet de récupérer des modules supportés par la communauté Zend Framework afin de ne pas réinventer la roue, principal intérêt d'un framework, tout en étant pleinement dans l'esprit de Zend Framework 1. Par rapport aux autres frameworks, celui de Zend est construit comme un ensemble de composants (briques) que le développeur utilise à sa guise.

D'autres nouveautés sont moins spectaculaires mais bien présentes :

- Les suffixes « `interfaces` » ont été retirés
- Les méthodes et attributs privés ne possèdent plus de « `_` » les précédant.



Architecture du squelette d'une application Zend Framework 2

- Les dépendances codées en dur ont été supprimées
- Plusieurs parties du framework, dont on s'est aperçu qu'elles étaient mal pensées lors des remontées de Zend Framework 1, ont été retravaillées.

> Les principales critiques

Le reproche le plus récurrent fait à Zend Framework 2 concerne sa complexité. Les blogueurs sont nombreux à crier leur désarroi face à cette « inutile » complexité, alors que la simplicité était le principal argument de « vente » de Zend Framework 1 (et de PHP).

Par exemple, le « getting started » de la documentation officielle (documentation bien fournie, pour le moment uniquement en anglais), pourtant une des priorités de Zend Framework 2, fait peur à voir. Tout ce code (45 pages de tutoriel tout de même) pour si peu ? C'est ce que beaucoup d'adeptes de ce framework décrivent. Après avoir suivi ce tutoriel, on ne peut qu'espérer que cette complexité disparaisse à l'usage et qu'elle constitue un réel gain de temps pour la suite, notamment pour les réalisations de grande envergure. Cette apparente complexification s'explique, entre autres, par le choix des contributeurs de réduire les traitements masqués (la magie, ou la complexité cachée évoquée plus haut). Avec les avantages de ses défauts...

La seconde critique, émanant elle-aussi des blogueurs anglo-saxons, pointe du doigt la succession de versions bêta qui sont sorties (peut-être un peu pour alimenter le buzz), avec entre elles des différences si importantes que le code écrit sous l'une de ces versions devait être entièrement retravaillé pour être fonctionnel sous une version bêta postérieure. Un logiciel en version bêta est pourtant censé connaître une certaine stabilité.

Un point qui se révèle à la fois positif et négatif : l'autoloader étant devenu systématique, il est désormais impossible d'utiliser une des briques de Zend Framework 2 sans passer par lui. Ceci met un terme aux briques totalement indépendantes qui apparaissent Zend Framework à une librairie plus qu'à un véritable framework.

Dernière critique mais pas des moindres : une application codée sous Zend Framework 1 est incompatible avec Zend Framework 2.

Zend propose bien des outils permettant d'automatiser le passage vers sa nouvelle version, mais cette migration automatisée ne recouvre qu'une partie des développements.

Ces critiques, toutes venues de courageux pionniers se lançant, pleins de bonne volonté, dans la nouvelle technologie de Zend, sont toutefois à modérer. On trouve en effet beaucoup de ces développeurs certainement expérimentés et de bonne foi, pour dire qu'ils n'utiliseront pas cette technologie qui a perdu l'esprit qu'elle avait à ses débuts. Mais quand on a la curiosité d'aller chercher les critiques des développeurs découvrant Zend Framework 1, en 2005, on est étonné de la virulence et de la similarité des arguments critiquant cette première release (complexité, documentation inextricable, « quick start » inexistant, influence de Java...). Oui, malgré une extrême simplicité souvent vantée, on échappe rarement à une période d'apprentissage.

> Conclusion

On l'aura bien compris, comme le répètent les auteurs de Zend Framework 2, en particulier Rob Allen (contributeur et auteur du livre « ZEN Framework in Action »), il est indispensable de penser « évolution » plutôt que « révolution », et donc d'accepter un petit investissement afin de profiter du patient mais inéluctable progrès des technologies. Bien que l'avalanche de nouveaux frameworks PHP sortant ces dernières années nous incite à la méfiance, il faut regarder les indicateurs nous permettant de se lancer avec confiance : l'importance de la communauté soutenant la technologie. Les développeurs contribuant à cette dernière utilisent GitHub (outil de gestion de versions décentralisé contrairement au célèbre SVN), le dernier chouchou de l'open source, qui met d'ailleurs à disposition une série de statistiques très intéressantes sur la fréquence des commits, l'importance des contributeurs...



La confiance que l'on peut mettre dans la société Zend qui a déjà fait ses preuves

Gautier Deruette, consultant Osaxis (www.osaxis.fr)

L'INFO permanente

- L'actu : le fil d'info quotidien de la rédaction
- La newsletter hebdo : abonnez-vous, comme 38 000 professionnels. C'est gratuit !

C'est PRATIQUE !

- Le forum : modéré par la rédaction et les auteurs de Programmez!, rejoignez les forums techniques de programmez.com
- Les tutoriels : une solution en quelques clics !
- Le téléchargement : récupérez les nouveautés.

www.programmez.com



Responsive Design : une autre manière de penser le

Souvent, on réduit le concept au Responsive Web Design. En réalité, il faut parler de Responsive Design ou encore de Responsive Interface. L'objectif est cependant identique, peu ou prou : comment concevoir une interface, une ergonomie (in fine, une application web, un site web) qui s'adaptent au contexte d'utilisation (smartphone, desktop, box, tablette, etc.). Il y a quelques années on parlait beaucoup de langage descriptif pour l'interface pour permettre une grande souplesse. Finalement, le Responsive Design est un lointain héritier de cela.

L'une des difficultés actuelle est d'avoir une application la plus « universelle » possible, notamment pour les applications web et les sites web. Car chaque plateforme mobile, tablette, possède des spécifications différentes. Au lieu d'adapter pour chaque terminal, le Responsive Design facilite la vie du développeur.

Avantages. Avec la multiplication des terminaux, plateformes et usages, impossible de concevoir une interface dédiée. Cela prendrait trop de temps, trop d'argent et trop de branches projets à maintenir. Le développeur se concentre plus sur le cœur du code, les fonctionnalités. Le designer se concentre en quelque sorte sur une matrice. Et le futur du web sera l'adaptatif. La promesse est simple : un site pour tous les terminaux.

Cependant, on peut opposer plusieurs contre-arguments : les ressources et le temps de développement pour faire du responsive (et le temps de tests), prudence sur la compatibilité entre les navigateurs, le temps de chargement (le responsive se fait à la volée), les diffé-



développement

rents terminaux impliquent du contenu différent et/ou adapté, le fait que tout contenu n'est pas forcément adapté aux mobiles...

A vous de jouer maintenant ! (pensez éventuellement à l'aspirine).

François Tonic

10 CONSEILS !

Splio a récemment sorti une liste de 10 bonnes pratiques.

- 1** Faire des couches et des calques simples ! Ayez un code HTML, le plus simple et le plus lisible possible.
- 2** Penser média, penser requête média. Ayez tout de suite les médias dans les différentes résolutions...
- 3** Viser les principales cibles. Bref ne vous dispersez pas.
- 4** Avoir des couches et calques flexibles et non pas statiques.
- 5** Que les images soient prêtes à être utilisées et visibles partout à la bonne taille et résolution...
- 6** Ne pas oublier le minimum et le maximum.
- 7** Penser « relatif ». Indispensable pour avoir une interface et les éléments flexibles
- 8** Se dire que l'on peut avoir une interface sur une colonne et non 3 !
- 9** Afficher uniquement le contenu essentiel et indispensable (à chaque terminal). Qu'est-ce qui est essentiel ?
- 10** Éviter de faire de l'émulation desktop sur un terminal mobile. Faites du vrai mobile !



© istockphoto/MFFphotography

Responsive Web Design : définitions et approches

Qu'est-ce qui se cache derrière le « Responsive Design » ? C'est en partant des fondations de cette nouvelle approche que l'on peut comprendre pourquoi et comment cela change complètement la manière de designer pour le Web. L'engouement autour de cette méthode est lié au changement du marché des périphériques pour consulter le Web, ce qui en fait un sujet incontournable pour les designers. Pour les intégrateurs, les solutions existent et sont simples à appréhender.

Elles remontent à un article de John Allsop écrit en 2000 dans lequel il présente sa vision du Web de demain <http://alistapart.com/article/dao>. Mais il faut attendre dix ans pour qu'une définition claire, code à l'appui, soit donnée. C'est ce qu'a fait Ethan Marcotte en mai 2010 dans son article « Responsive Web Design » (<http://alistapart.com/article/responsive-web-design>), dans lequel, comme il le déclarera lui-même plus tard (<http://youtu.be/EtOEutOOeeE>), il concrétise les notions abordées par John Allsop en 2000.

Flexibilité

Pour John Allsop, la flexibilité du format Web doit être vue comme un avantage et non plus comme une contrainte. Pour se faire comprendre, il utilise un rapprochement très intéressant entre la radio parente de la télévision et l'imprimerie parente du Web. Les premières émissions de télévision étaient simplement des émissions radio que l'on filmait. Partant de ce constat, la perspective concernant le média Web change : on pourrait croire que celui-ci est aujourd'hui mature, mais lorsque l'on observe le nombre de conventions qu'il emprunte encore à l'imprimerie sans recul, on revoit sa position sur le sujet. En effet, pour produire des interfaces facilement consultables par les utilisateurs, les designers se sont appuyés sur leurs compétences sur format papier. Or, celui-ci est par nature très strict : la taille des pages est définie et comme il est impossible d'interagir avec son contenu, celui-ci reste figé dans la page. Les designers ont donc pour habitude de travailler dans un cadre fixe : il s'agit de contrôler, pour chaque page, tout ce que perçoit et consulte l'utilisateur. Le designer attend alors des intégrateurs que ceux-ci reprennent exactement le modèle qu'il leur a donné. Il en est de même pour les clients qui ne comprennent pas toujours pourquoi le

rendu final de leur application n'est pas en tous points identique à ce que leur avait présenté le designer. Or, un des fantastiques avantages du Web, c'est le dynamisme possible du contenu de ses pages. Avantage que l'approche conventionnelle de l'expérience utilisateur ne permet pas de bien exploiter.

Adaptabilité

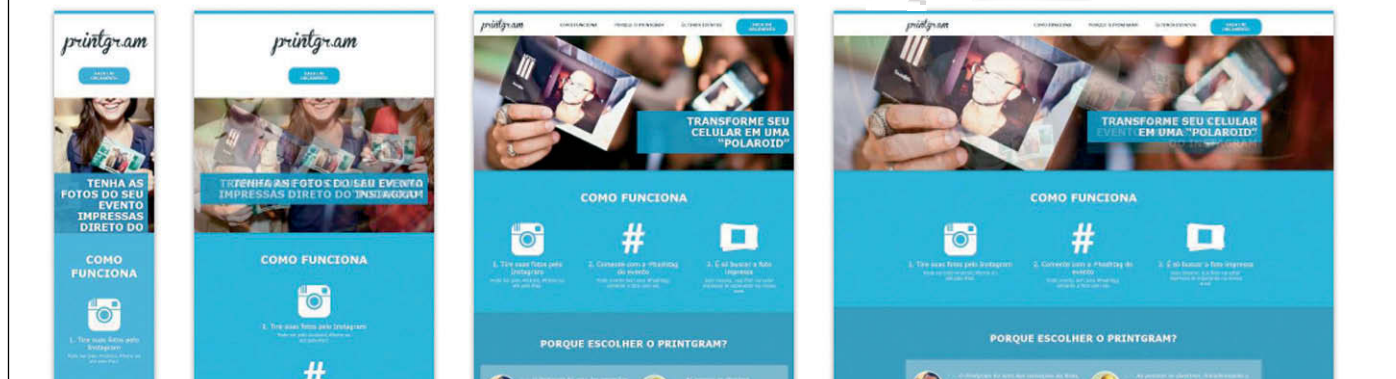
Ethan Marcotte reprend l'idée que présente John Allsop et lui donne une application concrète. Certaines contraintes techniques de 2000 ne sont plus aussi présentes, ce qui ouvre de nouvelles possibilités. Les navigateurs implémentent les nouveaux standards de présentation (CSS) avec toujours plus de soin. La dernière version de la norme CSS introduit les « media queries » qui permettent de spécifier le style à appliquer en fonction de certains critères d'affichage de la page Web : taille de la fenêtre, densité de pixel, etc. Ethan Marcotte en fait un des éléments principaux du « Responsive Web Design » permettant ainsi, enfin, la mise en place de l'adaptabilité des pages Web. Pourtant, en JavaScript, il est possible de récupérer ces critères depuis longtemps puis de changer l'affichage dynamiquement. L'utilisation des « media queries » est bien plus simple et propre mais n'offre pas de nouvelles possibilités. Pourquoi, alors, le Responsive Design et la réalisation d'interface de manière flexible et adaptative est-elle d'actualité ?

> LE CHOC SMARTPHONE

Afin de pouvoir travailler dans un cadre fixe, comme pour le format papier, il fallait convenir d'une taille cible puis de s'y contraindre. Cette démarche s'accompagnait généralement d'une étude pour connaître la taille des écrans les plus utilisés. Les utilisateurs minori-



Printgram



taires qui naviguaient sur un écran plus petit ou plus grand étaient alors tous simplement ignorés.

Une version spécifique

Avec l'arrivée des smartphones, le besoin exprimé a été dans un premier temps de réaliser des versions dédiées des applications et sites Web. Il s'agissait de la suite logique du « Specific Design ». La version « iPhone », les smartphones « Android » n'étant pas encore très présents sur le marché, devait suivre un design spécifique reprenant les normes et conventions du périphérique, un véritable casse-tête pour la mutualisation du contenu. De plus, les architectures en place proposaient rarement des services Web pour permettre à plusieurs applications d'interagir avec le système. L'arrivée des smartphones a donc introduit les prémices de la mise en place d'une architecture orientée services (SOA).

Et de deux

Le marché évoluant, la part de marché des utilisateurs ayant un smartphone sous Android ne pouvait plus être ignorée. Il fallait trouver une solution pour ces utilisateurs car les designs Web en place suivaient les préconisations d'Apple concernant la réalisation d'applications natives. Au-delà du style, la compatibilité surtout n'était pas optimale. Il y a donc eu une phase d'adaptation pour faire que la version mobile supporte à la fois les environnements « iOS » et « Android ». La charge de travail pouvait être assez conséquente et la plus-value perçue très minime. La stratégie utilisée pour aborder les smartphones commençait à être remise en question.

Et de trois ... vous êtes sûrs ?

La stratégie du « Specific Design » vola définitivement en éclats avec l'arrivée des tablettes et de Windows Phone. Il n'était plus question de produire encore une énième version du site ou de l'application. Si on additionne les coûts de développement des nouvelles solutions aux coûts de maintenance des anciennes, cette stratégie devient encore plus absurde. Car même pour des très grands noms qui n'ont pas de soucis de retour sur investissement, la complexité de gestion devient bien trop importante. Une solution à toutes ces problématiques est donc bien de suivre la nature de ce média qu'est le Web et de proposer des designs flexibles et adaptables comme l'ont suggéré John Allsop et Ethan Marcotte. Oui mais comment ?

> L'OPTIQUE MOBILE FIRST

Avant la phase d'intégration du design, qui utilisera alors nos fameuses « media queries », il faut réaliser une ergonomie et un design pour notre site. L'optique « Mobile First » consiste à se focaliser sur le format mobile en premier. C'est seulement ensuite que l'on fait

évoluer l'ensemble pour enrichir l'expérience utilisateur lorsque la surface d'affichage devient plus importante. Physiquement, les surfaces des mobiles sont très réduites. Le cadre que définit le mobile est alors très contraignant, il n'est pas possible d'afficher beaucoup d'informations simultanément. Le seul moyen de réussir à proposer une expérience utilisateur simple et efficace est de revenir à l'essentiel du besoin des utilisateurs. L'optique « Mobile First » permet donc à l'ensemble des acteurs autour de l'application, designers, ergonomes, commanditaires, de rester centrés sur les besoins premiers des utilisateurs et de bien les distinguer du superflu.

Ergonomie

L'ergonomie, ou expérience utilisateur, consiste à définir la manière dont les utilisateurs vont interagir avec le contenu du site. La séparation claire du design et de l'ergonomie est un facteur clé du succès de l'interface utilisateur. Cela se fait en réalisant des maquettes sans aucun esthétisme où les composants d'interface sont le plus abstrait possible. Le mieux est de procéder de manière itérative en réalisant des tests utilisateurs. Le livre « Don't Make Me Think! » de Steve Krug encourage les tests utilisateurs et est une excellente approche de l'ergonomie. Néanmoins, la flexibilité de l'ergonomie dépend de l'agencement des différents éléments de l'application entre eux, ce qui peut parfois devenir très complexe à mesure que l'on s'adresse à des écrans plus grands. En « Mobile First » la démarche est simplifiée car, dans un premier temps, on va surtout s'interroger sur ce qui doit vraiment être présent sur chaque page de site ou écran d'application. Les briques de base que sont les logo, titre, menu, article, produit, commentaire, etc., sont alors réalisées de manière très réduite et sont donc plus facilement identifiables, leur agencement devenant alors beaucoup plus aisé.

Pour la version mobile, les briques de base de la page seront généralement les unes au-dessus des autres. L'agencement des briques va ensuite être une affaire de colonage pour s'adapter à des tailles de page plus grandes. Les rubriques d'un menu qui étaient les unes au-dessus des autres se retrouveront les unes à côté des autres pour un affichage large. Le nombre de colonnes où les articles sont présentés passera d'une à trois selon la taille d'affichage.

Design

Encore une fois, lors de la réalisation du design, il est important de toujours bien séparer sa présentation avec celle de l'ergonomie auprès du client pour que des éléments d'esthétisme ne perturbent pas l'étude ergonomique. La phase de design doit elle aussi cibler le mobile. Comme pour l'ergonomie, le cadre minimaliste du mobile va contraindre le design à être simple et dépourvu d'éléments superflus. La phase de réflexion pour rendre le design flexible est plus déli-



cate. La définition des tailles des éléments va être très importante et souvent en rupture avec ce qui était proposé jusque-là. Les sites en « Responsive Design » vont avoir naturellement tendance à proposer des titres, textes et images plus gros et ayant moins de contenu. Le design est alors beaucoup plus focalisé sur la hiérarchisation de l'information pour faciliter son analyse que sur l'agencement d'une quantité trop importante d'information.

Graceful degradation

Pour réaliser des interfaces en Responsive Design, il existe une autre approche : le « Graceful degradation ». A l'inverse du « Mobile First », il s'agit de partir de l'interface la plus riche d'information (desktop) puis de supprimer des éléments pour arriver à la version la plus essentielle (mobile). Cette approche peut être intéressante dans le cas d'une refonte d'un existant. Il s'agit alors de vérifier si les éléments identifiés comme non essentiels sur la version minimale ont réellement une plus-value sur la version complète. La démarche est donc tout de même plus complexe qu'avec celle du « Mobile First » car il faut pouvoir être critique sur un travail déjà réalisé.

> EN PRATIQUE

Fluidité

Avant de cibler différents types d'écrans à l'aide des media queries, il faut réaliser un agencement fluide des différents éléments. Pour cela, la taille de ces différents éléments doit être définie le plus possible avec des pourcentages. Les images, les rubriques de navigation, etc., seront alors toujours présentées de manière optimale selon la taille de l'affichage. C'est une pratique en rupture avec les designs statiques car le rendu des pages ne semble alors plus contrôlé. La crainte de voir le design défini pour la page devenir inutilisable dans certains cas extrêmes d'affichage, pourrait être légitime si le « Responsive Design » s'arrêtait là. C'est alors qu'interviennent les media queries.

Media queries

Les medias queries sont dans les recommandations du W3C pour la norme HTML5/CSS3 depuis 2012. Le principe consiste à pouvoir appliquer des styles suivant un contexte d'affichage. Celui-ci va être ciblé en évaluant un ensemble de propriétés. Le conditionnement peut se faire lors de la déclaration de la feuille de style :

```
<link rel="stylesheet" media="screen and ("ma condition)" rel="stylesheet" href="exemple.css" />
```

Ou encore directement dans les styles :

```
@media screen and "mes conditions" { "mes css à appliquer" }
```

Le « et » logique entre plusieurs conditions se définit avec le terme « and ». Le « ou » logique se fait avec la virgule « , ». Attention toutefois, il n'est pas possible de mixer les deux en spécifiant des parenthèses. En d'autres termes : le « et » est prioritaire sur le « ou ». Il faut donc réaliser les expressions toujours avec les formes suivantes :

```
@media screen and ("ma condition") and ("ma condition") { "mes css à appliquer" }
@media screen and ("ma condition"), screen and ("ma condition") { "mes css à appliquer" }
```

Dans les exemples, le terme « screen » en début de requête permet d'adresser uniquement les périphériques de type « écran ». Un autre type communément utilisé est « print » pour les styles à appliquer

lors de l'« impression ». Il existe d'autres termes, mais ils ne sont pas actuellement implémentés de manière cohérente.

Les conditions

Elles se structurent par un couple de propriété/valeur séparées par « : ». Les propriétés sont constituées par des termes « width, height, etc. » qui peuvent se préfixer par « min » et « max ». Les préfixes servent à combler l'impossibilité d'utiliser les signes inférieurs et supérieurs « < », « > » car ils rentrent en conflit avec les balises HTML. Il est important de noter qu'il faut entourer les conditions avec des parenthèses.

```
@media screen and (min-width: 400px) and (max-width: 700px) { "mes css à appliquer" }
```

Les styles seront appliqués ici uniquement si la taille de la page est comprise entre 400 et 700 pixels. La liste complète des propriétés utilisables est disponible sur le site du W3C à l'adresse suivante : <http://www.w3.org/TR/css3-mediaqueries/>. Parmi les 13 propriétés de conditionnement, la propriété « width » sort du lot car elle permet de réaliser des conditions concernant la largeur de la page Web. Il s'agit de la propriété clé du « Responsive Design ». Cette propriété requiert de la subtilité car c'est elle qui permet d'adresser de manière optimisée les tablettes et smartphones. Le terme utilisé pour définir les paliers est « point de rupture ». Ce terme met bien en évidence le fait qu'ils servent à intervenir lorsque le design se « brise », c'est-à-dire lorsque le rendu n'est plus agréable. Il existe des points de ruptures standards, mais il est très important de garder en tête qu'ils doivent être redéfinis suivant le design choisi :

- Moins de 600px : smartphone
- 600 – 1024px : tablette
- Plus de 1200px : ordinateur de bureau

> FRAMEWORKS ET OUTILS

L'utilisation des frameworks de colonage permet de simplifier l'agencement lors des passages en mode mobile, tablette et desktop.

<http://foundation.zurb.com/>, <http://lessframework.com/>, <http://adapt.960.gs/>

Émulateurs

Au lieu de redimensionner la fenêtre de votre navigateur préféré, les émulateurs permettent de voir plusieurs formats en même temps ou de visualiser le rendu sur des périphériques cibles. <http://screenqueri.es/>, <http://quirktools.com/screenfly/>, <http://www.responsinator.com>

Mockup

Pour réaliser la phase d'ergonomie en faisant glisser les composants (textes, champs de saisie, etc.) dans un style abstrait.

<http://www.balsamiq.com/>

Exemples

Beaucoup d'exemples de sites en « Responsive Design » avec une présentation qui permet de visualiser le rendu selon 4 tailles d'affichage. <http://mediaqueri.es/>

Compatibilité

Certains scripts pallient l'absence des media queries lorsque les navigateurs n'implémentent pas les CSS3.

<https://github.com/scottjehl/Respond>,

<https://code.google.com/p/css3-mediaqueries-js/>

Guillaume Gaulard

Spécialiste Front-End : Ergonomie, Design et Développement, chez Zenika.





Passer au « responsive web design »

Cette approche présente l'inconvénient d'obliger les possesseurs de petit écran à scroller pour visualiser l'ensemble du contenu, ainsi que de laisser une impression de vide à ceux, plus chanceux, qui naviguent sur un écran de grande dimension, par exemple un 27 pouces. À l'inverse, l'apparence d'un site dont le design est responsive s'adapte en fonction de l'écran sur lequel il est affiché. Cela est possible de manière extrêmement simple, grâce à une nouvelle fonctionnalité du CSS 3 : les media queries. Grâce à elles, il suffit d'ajouter quelques lignes de code en en-tête de nos pages :

```
<!-- Feuille de style réservée aux fenêtres de moins de 1024 pixels de largeur -->
<link rel="stylesheet" media="(max-width: 1023px)" href="css/handled/global.css" type="text/css" />

<!-- Feuille de style réservée aux fenêtres de 1024 pixels ou plus -->
<link rel="stylesheet" media="(min-width: 1024px)" href="css/global.css" type="text/css" />

<!-- On note que les parenthèses encadrant min-width et max-width sont importantes -->
```

Dans cet exemple, la valeur seuil est fixée à 1024 pixels : en deçà, on bascule sur la feuille de style réservée aux petits écrans.

Les media queries peuvent également être directement insérées dans les feuilles CSS :

```
@media (min-width:1024px) {
    body {
        background: #e9f4fb no-repeat;
        width: 100%;
        margin: 0;
        padding: 0;
    }
}

/* ... */

}

@media (max-width:1023px) {

    /* ... */

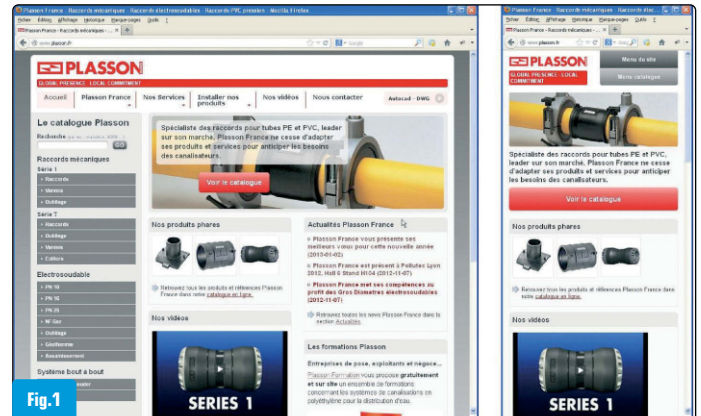
}
```

Les deux manières de procéder présentées ci-dessus sont tout à fait équivalentes [Fig.1]. Il est également possible de définir plusieurs valeurs seuils et autant de styles correspondants. On peut ainsi par exemple définir trois types de présentation pour les smartphones, les tablettes et les écrans d'ordinateur :

```
<link rel="stylesheet" media="(max-width: 479px)" href="css/smartphone.css" type="text/css" />

<link rel="stylesheet" media="(min-width: 480px) and (max-width: 767px)" href="css/handled/tablette.css" type="text/css" />

<link rel="stylesheet" media="(min-width: 768px)" href="css/
```



Exemple d'un site où le design change en fonction de la largeur de la fenêtre (responsive web design appliqué efficacement sous Drupal).

```
pc.css" type="text/css" />
```

```
<!-- Attention aux chevauchements de feuilles de style ! -->
```

La propriété media servait historiquement à distinguer des types de média. Par exemple, quand media prend la valeur « print », elle appelle la feuille de style qui donne les éléments de style nécessaires à une impression. Les media queries sont supportées par tous les navigateurs récents. Côté Microsoft par exemple, Internet Explorer 9 les supporte mais pas Internet Explorer 8.

Sur les navigateurs modernes donc, en réduisant la fenêtre, on voit qu'à chaque fois que le seuil est dépassé, le style change. Seulement, dès que l'on teste sur mobile, bien que doté d'un petit écran, le site semble considérer que celui-ci est bien plus grand. C'est là qu'entre en jeu la notion de viewport. Le viewport est la place qui est attribuée par le navigateur au site pour s'afficher. C'est par rapport au viewport que le site gère son affichage. Par exemple, considérons que le viewport a une largeur (width) de 1000 pixels et que l'on a défini dans notre style la largeur de la balise <html> :

```
html { width: 95% }
```

Dans un tel cas, la largeur de la balise <html> est de 950 pixels. Classiquement, les dimensions du viewport correspondent à l'intérieur de la fenêtre du navigateur. Seulement, sur les mobiles, le viewport est bien plus grand que l'écran lui-même. Il dépasse en effet de l'écran et le site est affiché sur une surface plus grande. Il faut alors forcer la largeur du viewport pour que celle-ci corresponde à la taille réelle de l'écran (device-width). Ceci se fait par une balise méta :

```
<meta name="viewport" content="width=device-width" />
```

Les media queries utilisent alors cette nouvelle largeur du viewport pour faire leur test. On aurait pu se référer directement à la largeur de l'écran en utilisant dans l'attribut media l'expression device-width pour former des balises de ce type :

```
<link rel="stylesheet" media="(min-device-width: 479px)" href="css/grand_ecran.css" type="text/css" />
```

Cela serait une solution en théorie. Mais, outre le fait qu'il serait dommage de ne pas pouvoir tester les réactions du site aux redimension-

nements à partir de son navigateur de bureau, il se trouve qu'en pratique, sans rentrer dans les détails, la largeur de l'écran détectée par les media queries ne correspond pas toujours à la largeur réelle de l'écran. Que chacun se rassure, cette notion de viewport est de loin la plus complexe dans tout ce qui a trait au responsive web design. En plus de définir des « points de ruptures » au-delà desquels le style est changé, la méthode du responsive web design préconise d'introduire de la souplesse dans le design. En ce sens, il convient de faire en sorte que toutes les mesures soient relatives et non statiques, en pourcentage et non en pixel. Ainsi, il n'est pas nécessaire de multiplier les points de ruptures qui, sinon, risquent d'être aussi nombreux qu'il existe de tailles d'écran ! Une fois cela mis en place, à savoir simplement rajouter trois lignes de balises HTML, le gros du travail se passe uniquement dans les feuilles de style CSS. Ceci constitue bien évidemment un énorme avantage. Le comportement du site côté serveur (PHP, .Net ou Java par exemple) reste en effet identique. Rendre un site « responsive », c'est donc travailler son style. Par exemple, tout site mis en place depuis une solution de gestion de contenu web (CMS) permettant de personnaliser son style, peut devenir responsive via les méthodes précédemment expliquées.

> Est-ce vraiment utile ?

Avoir un site disponible et ergonomique pour un smartphone n'est donc plus une option réservée aux plus gros, mais bien un investissement inévitable pour les années à venir. Une des solutions permettant d'adapter un site aux terminaux modernes est de les détecter grâce au « user agent ». Le user agent est envoyé au serveur lors de la requête HTTP et contient un certain nombre d'informations sur le client, notamment le type de terminal qu'il utilise. Avec une condition adéquate, on peut adapter le site en fonction du terminal du client. Par exemple :

```
if( strpos($_SERVER['HTTP_USER_AGENT'],'iPhone') ||
    strpos($_SERVER['HTTP_USER_AGENT'],'Android') ) {
    /* Code spécifique aux iPhones et Androids */
}
```

Ce genre de pratique présente cependant de nombreux inconvénients. En effet, la condition présentée dans l'exemple précédent est très incomplète. Il faut y ajouter tous les terminaux susceptibles d'avoir besoin d'un design adapté. La liste est en constante évolution : iPod, tablettes, Windows Phone... et les concurrents en tous genres qui se multiplient. Autant le dire, ceci représente un inconvénient majeur, notamment par rapport à une technique telle que le responsive web design.

> Les astuces à connaître

HTML

Si le site a été bien pensé en amont, la tâche en sera plus aisée. Quelques rappels de bonnes pratiques utiles pour un bon design responsive :

- les tableaux ne sont pas utilisés pour faire de la mise en page, mais pour structurer des données. Leur position statique empêche de faire du web design. Il faut donc les employer selon l'usage pour lequel ils ont été conçus, à savoir afficher et structurer des données,
- il est important que le HTML soit simple. Il faut éviter un maximum les <div> imbriqués, le style directement inclus dans le HTML (!!), les éléments flottants... La règle étant, cela paraît évident, que l'arbre HTML soit le moins complexe possible.

CSS

Il a été présenté en introduction comment utiliser les media queries. Cependant, tout le style concernant la mise en forme (background, couleurs, police...), avec la méthode présentée en début d'article, serait redondant puisque présent dans plusieurs feuilles destinées à divers écrans. Une astuce consiste à appeler sous condition la feuille de style réservée aux petits écrans après les appels sans condition des feuilles de style du site classique. Les propriétés CSS s'appliquant de manière effective étant les dernières appelées, on peut créer une forme d'héritage, et ne retravailler que le code de mise en page :

```
<!-- Style général du site -->
<link rel="stylesheet" href="css/global.css" type="text/css" />

<!-- Style particulier aux fenêtres d'au plus 1024 pixels écrasant
les propriétés à adapter (width,...) -->
<link rel="stylesheet" media="(max-width: 1024px)" href="css/
mobile/global.css" type="text/css" />
```

[Fig.5]

On a ainsi moins de code redondant, donc moins de maintenance. Cependant, procéder ainsi dégrade les performances du site puisque des informations inutilisées sont chargées. Mais alors, pourquoi ne pas construire un design pour mobile et le surcharger quand on se trouve sur un ordinateur avec a priori une meilleure connexion ? Tout simplement parce que, chronologiquement parlant, il est rare que l'on construise le site mobile avant de vouloir l'adapter aux écrans de taille plus importante.

Dimensionnement et positionnement

Pour des raisons évidentes, il est impératif de transformer toutes les positions absolues en positions relatives. Ceci signifie qu'il ne doit plus y avoir de « position : absolute » dans le code. Cette modification a de quoi donner la migraine, surtout quand du JavaScript vient modifier largeurs et positions. Heureusement, grâce aux fonctions comme celle de l'onglet « apparence » de Firebug, on s'en sort moyennant quelques cachets d'aspirine ! Pour pouvoir être responsive, le style doit comporter le minimum de dimensions en pixel. Les pourcentages qui changent dynamiquement en fonction de la taille de leur conteneur doivent être préférés. Les « em » ou « ex » qui donnent la taille des caractères en fonction de la taille courante de la police représentent également une bonne pratique mais on peut leur préférer les pourcentages afin d'uniformiser les unités. Les trois lignes suivantes sont équivalentes :

```
font-size: 50%;
font-size: 0.5em;
font-size: 0.5ex;
```

À ce propos, faut-il préférer « width:100% » ou « width:auto » ? Ces valeurs n'induisent pas le même comportement. Avec la valeur 100 %, l'objet prend toute la place qui lui est attribuée, faisant dépasser à l'extérieur de son conteneur sa bordure et sa marge [Fig.6]. « auto », qui tient compte de la bordure et de la marge, est donc à préférer.

Toutes les dimensions ne sont pas forcément à changer : une marge de 3 pixels ne choque pas plus sur un iPhone que sur un 19 pouces. Les hauteurs également peuvent être conservées. Pour rendre le site agréable sur un petit écran, les blocs doivent être passés les uns en dessous des autres afin de n'avoir qu'à monter ou descendre pour naviguer sur nos pages. Le menu, le contenu, la sidebar seront

sur une même colonne. Cette étape est généralement appelée la linéarisation d'un site. Certaines hauteurs seront bien sûr à adapter. Ce sera par exemple le cas pour les hauteurs des blocs contenant du texte qui, une fois leur largeur réduite, tiennent sur un plus grand nombre de lignes. Dans ce cas, utiliser « height: auto; » est tout simplement magique. On supprime également les blocs qui ne sont pas indispensables. Une sélection doit ainsi être faite sur les informations à afficher prioritairement pour les mobiles. Cette étape est appelée la « dégradation ». Pour cela rien de compliqué : [Fig.7].

```
#sidebar1 {
  display: none;
}
```

Les images

Pour adapter la taille des images, il existe une astuce géniale permettant en deux lignes d'avoir des images redimensionnées à une taille adéquate :

```
img {
  max-width: 100%;
  height: auto;
}
```

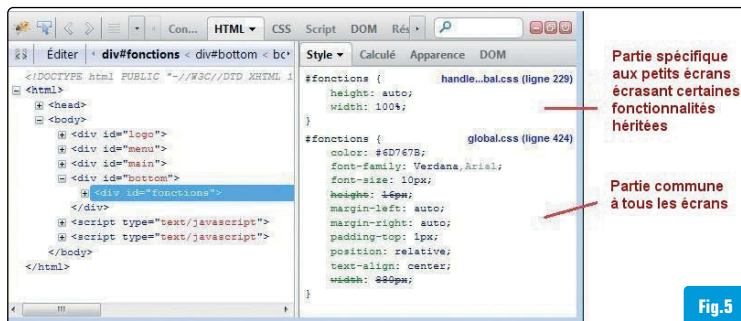
La propriété « max-width » empêche l'image de dépasser de son conteneur tout en prenant le maximum de place possible. Tandis que « height: auto » permet à l'image ne pas être déformée par le redimensionnement. Si les images sont placées dans un tableau, certains navigateurs comme Firefox ou Internet Explorer ne permettent d'utiliser la propriété max-width qu'à condition d'ajouter une propriété aux tableaux :

```
table {
  table-layout: fixed;
}
```

> Les limites du responsive web design

Faire un seul et même site pour tous les terminaux, c'est considérer à tort qu'ils sont égaux :

- La capacité de calcul d'un mobile est nécessairement plus faible qu'un ordinateur, les scripts exécutés par le client sont donc plus lents sur un smartphone. De plus, en faisant du responsive web design on utilise plus de CPU dès lors qu'on redimensionne les images par exemple.



Héritage de style de la balise « fonction » mis en évidence à l'aide de Firebug.

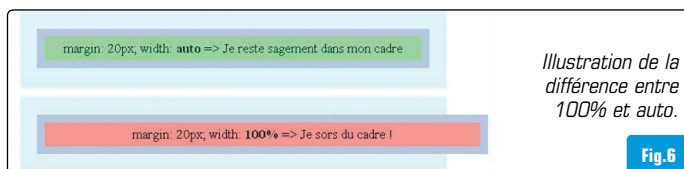


Illustration de la différence entre 100% et auto.

Fig.6

- Un mobile, la majorité du temps, n'est pas connecté en wifi. Le débit de sa connexion est alors très variable puisqu'il dépend de la distance à la borne réseaux, du mouvement du terminal (lors d'un déplacement, dans une voiture par exemple, la vitesse de connexion est réduite), de l'heure de la journée, de l'opérateur... Et pourtant, lorsque l'on fait du responsive design, on n'impose pas moins de données à charger pour afficher une page que sur un PC. En effet, utiliser les media queries et les « display: block » ne fait que masquer les éléments non affichés. De la même manière, les images redimensionnées (donc plus petites) sont téléchargées intégralement, ce qui se traduit par une quantité non négligeable de bande passante inutilement exploitée.

Ces différences, mises en évidence, tendent à montrer qu'il est bien plus efficace de développer un site spécifique aux mobiles. Or la performance n'est absolument pas à négliger. Le marketingtechblog a publié une étude montrant à quel point la performance est importante. Par exemple, une entreprise faisant 100 000 \$ de chiffre d'affaires par jour, en augmentant d'une seconde le chargement de ses pages, peut perdre jusqu'à 2,5 millions de dollars sur un an ! Un site développé exclusivement pour les mobiles ne comporte pas les inconvénients ci-dessus et a de plus un fonctionnement plus approprié puisqu'il a été pensé pour des petits terminaux. Par exemple, un site d'une compagnie de transport préfère mettre en avant, sur son site mobile, les prochains départs et les itinéraires possibles en fonction de la position de l'utilisateur et ce, sans polluer l'utilisateur avec les multiples options et possibilités des services offerts par la compagnie, plutôt réservées au site consultable depuis chez soi. Par ailleurs, le responsive web design ne se base que sur la largeur du terminal sur lequel il s'affiche avec les difficultés que l'on a vues. Cela à l'avantage d'avoir un design basé uniquement sur la taille de l'écran et de passer outre le fait que le terminal soit un téléphone, un écran d'ordinateur, ou une tablette. Nous avons d'ailleurs utilisé dans cet article, de manière abusive des mots tels que « mobiles » ou « tablettes » pour désigner les petits écrans, leur taille étant le seul critère de distinction permis par les media queries. Un élément important sur les nouveaux terminaux concerne la mise à disposition d'un écran tactile. La technique des media queries ne permet pas de distinguer un vieux et petit écran non tactile d'une grande tablette tactile. Au mieux, on peut penser qu'un petit écran est un mobile ou une tablette et qu'il est alors tactile et bénéficie d'une bande passante réduite. Dans de tels cas, on essaie alors de faire en sorte que le site fonctionne sans avoir recours à la propriété CSS hover inutile aux tactiles, et ne nécessite pas non plus une grande bande passante.



Gautier Deruette, consultant Osaxis - <http://www.osaxis.fr>

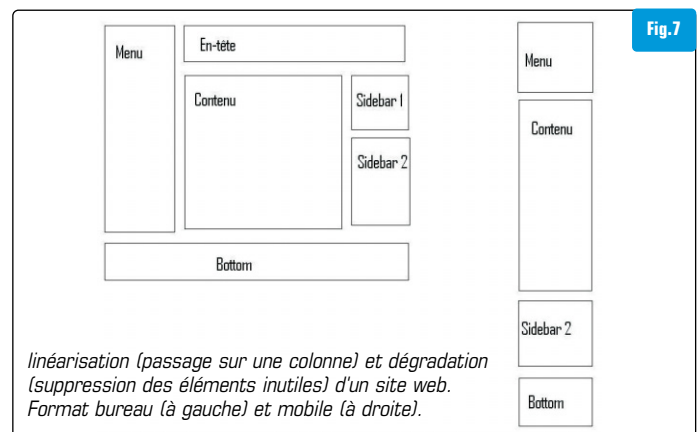


Fig.7

linéarisation (passage sur une colonne) et dégradation (suppression des éléments inutiles) d'un site web. Format bureau (à gauche) et mobile (à droite).

Votre site « responsive »

Google préconise cette solution pour la création de sites mobiles, un seul site à indexer, pas de risque de créer du contenu dupliqué, le linking reste concentré sur votre domaine principal au lieu de s'éparpiller sur plusieurs domaines.

Pour créer un site responsive, trois ingrédients primordiaux sont nécessaires : un viewport adapté, une grille flexible et l'utilisation des media queries.

> Déclarer le viewport

Le viewport représente la surface d'affichage d'une page web sur un écran.

Sur les appareils mobiles, le viewport est en général fixé à 980px. Sans viewport, le site sera adapté à ces 980px par défaut, sans tenir compte de la résolution réelle de l'écran.

Le résultat sera une réduction de la page, à l'utilisateur ensuite de zoomer pour consulter le contenu.

Le site du Monde sur un écran de 320px de large



Fixer le viewport est donc une des premières actions à mettre en place lors de la création d'un site responsive. C'est ce qui va imposer au navigateur le bon comportement pour afficher le site. Pour le déclarer, on utilise un méta tag viewport qui fut introduit par Apple et qui est maintenant reconnu par l'ensemble des navigateurs mobiles modernes. La syntaxe suivante applique au viewport une largeur de 320px :

```
<meta name="viewport" content="width=320">
```

Le problème avec cet exemple vient du fait qu'en passant l'écran en mode paysage, la largeur du viewport reste figée à 320px pour une résolution réelle de l'écran qui peut être de 480px ou plus. Le contenu se trouve alors agrandi. Pour éviter cet effet, on indique au navigateur que la largeur du viewport sera égale à la largeur de l'appareil utilisé :

```
<meta name="viewport" content="width=device-width">
```

On peut aussi définir dans ce tag le facteur de zoom initial :

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Des niveaux de zoom minimum et maximum :

```
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=3, minimum-scale=0.5">
```

Ou encore empêcher l'utilisateur de zoomer dans la page :

```
<meta name="viewport" content="width=device-width, user-scalable=no">
```

Plus rarement utilisé, il est possible de donner une hauteur au viewport :

```
<meta name="viewport" content="height=device-height">
```

> Créer une grille flexible

Créer une grille flexible c'est faire un premier pas vers le responsive web design et c'est ce qui va nous permettre par la suite de transformer une page plus facilement et plus naturellement. Pour arriver à ce résultat on utilise comme unité les pourcentages, en lieu et place des traditionnels pixels. Prenons la structure suivante :

```
<div id="wrap">
  <div id="inner">
    <div id="content-wrap">
      <div id="content"></div>
      <div id="content-aside"></div>
    </div>
    <div id="aside-wrap"></div>
  </div>
</div>
```

Imaginons qu'une maquette graphique nous donne les dimensions suivantes : 960px pour "wrap", 900px pour "inner", 750px pour "content-wrap", 550px pour "content", 200px pour "content-aside" et enfin 150px pour "aside-wrap".

Pour rendre ce design flexible nous allons devoir transformer ces valeurs en pixels, en pourcentages, en utilisant la formule $\text{cible} \div \text{contexte}$, le contexte étant l'élément parent de la cible.

En CSS cela va se traduire de la façon suivante :

```
#wrap {
  width: 90%; /* remplace les 960px */
  max-width: 1200px
}

#inner {
  width: 93.75%; /* 900÷960 */
}

#content-wrap {
  width: 83.333333%; /* 750÷900 */
}

#content {
  width: 73.333333%; /* 550÷750 */
}

#content-aside {
  width: 26.666667%; /* 200÷750 */
}

#aside-wrap {
  width: 16.666667%; /* 150÷900 */
}
```

Notre grille flexible est en place et cette logique peut être poussée plus loin en utilisant la même technique pour les marges internes et externes, pour la typographie, etc...

Pour obtenir des images flexibles dans cette grille qui l'est tout autant, l'astuce suivante remplit parfaitement son office :



```
img {
  max-width: 100%;
  width: auto; /* On pourrait s'en passer sauf IE8 */
  height: auto;
}
```

> Définir les media queries

Le viewport est déclaré, la grille du site est fluide, on peut se lancer dans ce qui est l'essence du responsive web design : les media queries. Les media queries peuvent s'apparenter à un "if", un ensemble de règles CSS sera appliqué si le navigateur rencontre les conditions d'une media query, aussi appelée point d'arrêt ou breakpoint. Une media query peut être déclarée grâce à l'attribut "media" de deux façons distinctes.

Dans la CSS :

```
@media screen
and (min-width: 768px) {
  /* Ensemble de règles CSS */
}

@media screen
and (min-width: 980px) {
  /* Ensemble de règles CSS */
}
```

Ou dans la page HTML :

```
<link rel="stylesheet" type="text/css" href="980.css" media="screen and (min-width: 980px)">
```

Les conditions les plus communément utilisées sont relatives aux dimensions du viewport (width, device-width, height, ...). Mais les media queries sont également capables de prendre en compte d'autres variables liées au périphérique de consultation. Il est en effet possible de créer des règles CSS en fonction de l'orientation (portrait, paysage), de la résolution (résolution, pixel-ratio), du support de la couleur (color, monochrome) ou du type de périphérique lui-même (screen, print, all). Les conditions peuvent, pour la plupart, être préfixées par "min-" ou "max-" et combinées en utilisant les opérateurs "and", "only", "not". Dans cet exemple les règles s'appliqueront pour un périphérique de type écran, d'une largeur de 320px à 480px et uniquement avec un pixel-ratio de 2, ce qui équivaut à un écran haute définition.

```
@media screen
and (min-width: 320px)
and (max-width: 480px)
only (pixel-ratio : 2) {
  /* Ensemble de règles CSS */
}
```

La liste exhaustive des conditions est disponible sur le site du W3C : <http://www.w3.org/TR/css3-mediaqueries/#media1>

> Quels breakpoints utiliser ?

Si dans la théorie le nombre de breakpoints n'est pas limité, dans la pratique l'idéal serait de ne pas en abuser dans un souci de maintenabilité. Mais il n'y pas de vérité écrite dans le marbre et votre projet dictera vos choix. Cependant, deux approches existent dans l'utilisation des breakpoints. L'une basée sur la résolution du périphérique

et l'autre sur le contenu. L'approche par la résolution du périphérique, comme son nom l'indique, se base sur la résolution en pixels et pourrait ressembler à ceci :

```
@media screen
and (min-device-width: 320px)
and (max-device-width: 767px) {
  /* Ensemble de règles CSS pour smartphone */
}

@media screen
and (min-device-width: 768px)
and (max-device-width: 1024px) {
  /* Ensemble de règles CSS pour tablette */
}

@media screen
and (max-device-width: 1200px) {
  /* Ensemble de règles CSS pour desktop */
}
```

Etre responsive, c'est pouvoir s'adapter à un maximum de situations. Mais la limite de cette approche est qu'elle ne prend pas en compte le confort de lecture sur un écran (qui se situe entre 45 et 70 caractères par ligne), ni le fait que les utilisateurs peuvent utiliser des aides techniques pour zoomer dans une page web, ni les spécificités de certains matériels. Avec des breakpoints en pixels, la mise en page peut être fortement dégradée par ces facteurs. L'approche par le contenu va remplacer les pixels par les ems. Puisqu'un caractère mesure environ 0.5em et qu'une ligne idéale occupe 70 caractères on peut fixer un breakpoint à 35em. Dans ce cas, le design s'adaptera à la taille de la police employée et se transformera lorsque le viewport atteindra 35em de large pour un confort de lecture idéal.

```
@media screen
and (max-device-width: 35em) {
  /* Ensemble de règles CSS pour desktop */
}
```

Pour tester son site responsive sans forcément passer par un smartphone ou une tablette on peut utiliser la "vue adaptative" dans les outils de développement de Firefox ou bien le bookmarklet Resizer disponible à cette adresse : <http://lab.maltewassermann.com/viewport-resizer/>

> Optimiser les performances

Qui dit usage mobile, dit réseaux mobiles qui n'ont pas forcément la vitesse de l'ADSL. Il est donc nécessaire d'optimiser au maximum les performances d'affichage.

Plusieurs techniques simples à mettre en place côté client existent :

- Compresser les CSS, les JS et les images
- Utiliser des sprites CSS et des icônes en Base64, pour réduire le nombre de requêtes HTTP.
- Recourir aux propriétés CSS3 lorsqu'elles vous permettent d'éviter de faire une image.
- Créer des styles utilisés par l'ensemble des périphériques, puis du spécifique déclaré dans des breakpoints, ce qui permettra par exemple à un smartphone d'utiliser les seules données qui lui sont nécessaires.

Gilles Rouyer - Consultant Soat, spécialisé en expérience utilisateur et développement front-end

Réaliser son premier site adaptatif avec l'approche **mobile-first**

Depuis son origine et jusqu'à récemment encore, le paysage web demeurait assez simple, avec le desktop comme unique support d'affichage. L'essor incroyable des terminaux mobiles a depuis totalement changé la situation.

Les ordinateurs de bureau et les PC portables ont longtemps constitué la cible unique d'affichage du web. Aujourd'hui, le nombre d'appareils pouvant accéder au web ne cesse d'augmenter et la grande variété de ces derniers amène de nouvelles problématiques d'affichage et de fonctionnement toujours plus complexes. En considérant les plus petits smartphones du marché et en allant aux TV connectées sans oublier les tablettes et les ordinateurs desktops, les tailles d'écran varient ainsi de 3 pouces jusqu'à 40 pouces et plus ! A chacune de ces tailles d'écran correspondent des contraintes d'utilisabilité : tactile, débit internet réduit, écran sans couleur ...

Dès lors, on comprend mieux toute la complexité pour proposer à l'utilisateur une expérience optimale dans chacun des contextes d'utilisation. Puisque mettre au point une version unique d'un site web fonctionnant sur tous les terminaux est utopique, une première solution consiste à créer plusieurs versions d'un site, adaptées à chacun des contextes d'affichage ou bien développer des applications spécifiques pour mobiles. Les coûts induits par ces solutions sont énormes sans parler des problèmes de maintenance et de compétences. En effet, réaliser une application mobile spécifique nécessite de cibler à minima les plateformes Android et iOS avec, en parallèle, un travail à fournir sur le site web en lui-même. Cette approche est ainsi loin d'être évidente.

> Qu'est-ce que le Responsive Web Design ?

Pour faire face à ces enjeux, la notion de Responsive Web Design est apparue. Cette dernière répond aux besoins d'adaptabilité toujours plus grands d'un paysage web qui se complexifie. Le Responsive Design regroupe ainsi différents principes et technologies permettant de concevoir un site web s'adaptant à tous les types d'écran existants. Au lieu de concevoir des versions d'un site web déclinées pour les différents terminaux d'affichage, on réalise une seule version d'un site avec une interface auto-adaptable. De fait, les sources d'information et les socles techniques ne sont pas dupliqués ce qui entraîne des économies d'échelle importantes au vu du nombre croissant de terminaux puisqu'outre les smartphones et tablettes, les télévisions connectées se positionnent sur le marché et seront des cibles dans le futur.

Si le fait de prendre en compte les besoins utilisateurs en termes de redimensionnement pour adapter un site à la taille de leur écran est une part importante du Responsive Design, l'approche ne se limite pas à cela. Il est essentiel de penser aux besoins des utilisateurs suivant le contexte d'utilisation d'un site. Ainsi, en situation de mobilité ces besoins ne sont pas les mêmes que sur le desktop. Utiliser à bon escient les possibilités de géolocalisation des terminaux mobiles est par exemple appréciable, tout comme le fait de prendre en compte les problématiques de débit internet. Enfin, les terminaux

tactiles offrent des perspectives intéressantes qu'il serait dommage de laisser de côté. Clairement, le Responsive Design doit être perçu comme une solution permettant de réaliser des sites web proposant une interface auto-adaptable en regard du contexte d'affichage, tout en tenant compte au mieux de celui-ci pour fournir un fonctionnement le plus optimisé possible.

> Comment choisir ?

Offrir une expérience d'utilisation optimale quel que soit le support est une problématique centrale et plusieurs solutions sont envisageables. Développer des applications mobiles séparées du site web desktop ou bien concevoir des versions de sites distinctes sont des solutions. S'appuyer sur le Responsive Design pour n'avoir qu'un seul site doté d'une interface auto-adaptable en est une autre. Comment choisir ? Outre les considérations budgétaires en faveur du Responsive Design, certains éléments facilitent le choix :

- La réalisation de versions mobiles séparées s'impose dans le cas où le contexte d'utilisation en mobilité peut bénéficier des spécificités de ces appareils et si les interactions utilisateurs sont riches
- Privilégier l'approche Responsive Design pour des sites majoritairement informatifs
- Plus la liste des terminaux ciblés est variée et importante, plus le Responsive Design apporte des gains importants

> Approche mobile-first

Une fois la décision prise d'appliquer les principes du Responsive Design, deux approches existent. L'approche historique consistait à partir d'un site adapté pour le desktop pour ensuite construire un affichage dégradé pour les terminaux. Aujourd'hui, la tendance est de partir d'une version mobile pensée pour écrans plus petits, pour ensuite l'adapter aux écrans plus grands allant des tablettes aux ordinateurs de bureau. On parle ainsi d'approche mobile-first. Celle-ci privilégie la conception d'une interface ergonomique pour mobiles avant d'établir ensuite l'ergonomie pour le desktop et les écrans plus larges.

Elaborer une interface ergonomique sous les contraintes des terminaux mobiles force à épurer le contenu d'un site en limitant au maximum les éléments visibles afin de présenter l'essentiel aussi rapidement que possible. Pour ce faire, le développeur met au point des petits blocs adaptables. Construire une interface optimisée pour des contextes mobiles a ainsi un impact positif sur l'affichage sous tablettes, bureau mais également de futurs contextes qui viendraient à apparaître.

> Concepts clés

Techniquement parlant, le Responsive Web Design tourne autour de 3 fonctionnalités clés :



- Les media queries, grandement améliorées avec les CSS 3, sont utilisées pour appliquer des styles spécifiques en se basant sur les capacités des supports d'affichage.
- Le concept de grille fluide. Il s'agit de travailler avec des blocs épurés au sein d'une page en définissant leur dimensionnement de manière relative.
- La définition d'images et de médias flexibles qui pourront ainsi être redimensionnés en unité relative de manière dynamique ou via CSS.

> Mise en pratique

Afin de mettre en pratique le Responsive Design, nous allons réaliser un site web constituant un store d'applications mobiles au sein duquel les utilisateurs pourront acheter des applications, visualiser des copies d'écran et laisser des commentaires. Un tel site ne peut se limiter au desktop puisqu'il doit absolument fonctionner de manière optimale sur smartphones et tablettes. Ce site s'appuiera sur HTML 5, CSS 3 et du Javascript pour les comportements dynamiques. La mise en place du site commence avec une balise meta définissant son viewport. Sans celle-ci, les navigateurs de supports mobiles réalisent un travail pour rendre la page de manière optimale sur petits écrans en zoomant. Afin de conserver la résolution d'affichage réelle et puisque notre but est d'optimiser le site pour les supports mobiles, nous définissons le viewport comme ceci :

```
<meta name="viewport" content="width=device-width, initial-scale=1" />
```

Structure

La structure du site tire parti des possibilités introduites par HTML 5 et notamment les balises sémantiques telles que header, nav, section ou footer. Au sein de la page, un header est utilisé pour la barre de navigation du site. Elle contient le logo du store ainsi que le menu représenté par une liste de rôle "navigation". Pour représenter le contenu principal du site, on utilise une grille fluide avec plusieurs colonnes qui sont des blocs div. Ces blocs seront ensuite adaptés suivant le terminal d'affichage. En dessous de la grille, on retrouve un bloc affichant les commentaires et enfin au bas de page, on affiche à nouveau le menu. A titre d'exemple, le header du site et la grille fluide tous deux présents au sein du body ont l'allure suivante :

```
<header role="banner">
<div>
  <h1 class="logo"><a href="#"></a></h1>
  <ul id="nav-anchors" class="nav-anchors">
    <li><a href="#nav" id="menu-anchor">Menu</a></li>
    <li><a href="#search" id="search-anchor">Recherche</a></li>
  </ul>
  <form id="search" action="#" method="post" class="search reveal">
    <fieldset>
      <legend>Recherche Apps</legend>
      <label for="search-site">Recherche Apps</label>
      <input type="search" id="search-site" placeholder="Recherche Apps" />
      <input type="submit" value="Search" />
    </fieldset>
  </form>
  <nav id="nav" class="nav reveal">
    <ul role="navigation">
      <li><a href="#">Android</a></li>
```

```
<li><a href="#">iOS</a></li>
<li><a href="#">WP 8</a></li>
</ul>
</nav>
</div>
</header>

<div class="content app-info" role="main">
  <section class="app-main">
    <header>
      <h1 class="app-name">Live Weather Lite</h1>
      <div class="app-overview">
        <strong class="price">$1.00</strong>
        <ol class="star">
          <li class="on">&#9733;</li>
          <li class="on">&#9733;</li>
          <li class="on">&#9733;</li>
          <li class="on">&#9733;</li>
          <li>&#9734;</li>
        </ol>
        <a href="#reviews" class="review-count">123 Revues</a>
      </div>
    </header>

    <div id="app-img" class="app-img">
      <figure class="img-container" id="img-container">
        
      </figure>
      <nav>
        <ul>
          <li><a href="images/app1_min.jpg"></a></li>
          <li><a href="images/app2_min.jpg"></a></li>
          <li><a href="images/app3_min.jpg"></a></li>
        </ul>
      </nav>
    </div>

    <div class="app-description">
      <div id="a-desc">
        <p>Description ...</p>
      </div>
    </div>

    <form method="post" action="#" id="app-form" class="app-form">
      <fieldset>
        <legend>Ajout Panier</legend>
        <div class="field qty-field">
          <label for="qty">QTE</label>
          <input name="qty" id="qty" class="qty" type="number" pattern="[0-9]*" value="1" min="1" max="999" />
        </div>
        <div class="submit-form">
          <input type="submit" class="btn" value="Ajout au Panier" />
        </div>
      </fieldset>
```



```
</form>
</section>
</div>
```

> Style

Avec l'approche mobile-first, on définit en premier les styles appliqués aux terminaux mobiles. On rajoute à notre page les balises link suivantes :

```
<link rel="stylesheet" type="text/css" href="style.css" media="screen, handheld" />
<link rel="stylesheet" type="text/css" href="enhanced.css" media="screen and (min-width: 40.5em)" />
```

On déclare en premier la feuille de style la plus générale avant de définir les règles pouvant la surcharger ou la compléter. De fait, style.css est appliquée pour les terminaux de type écran ou mobiles. Quant à enhanced.css, elle est appliquée aux terminaux écran avec une taille d'affichage minimum de 40.5em. C'est donc au sein de cette feuille que seront appliquées les media queries complétant l'affichage pour les terminaux plus grands.

La taille de 40.5em est utilisée comme point de rupture en regard du contenu de la page. C'est bien le contenu qui détermine les points de rupture pour lesquels l'affichage doit changer. En effet, les dimensions des terminaux ne peuvent être considérées comme des points de rupture fiables et pérennes dans le temps. La bonne pratique qui en ressort est l'utilisation des unités relatives telles que les pourcentages ou les em lors de la conception d'une page afin de la rendre le plus flexible possible puisque ces unités sont de loin plus compatibles avec les variations de taille d'écran, de densité de pixels et de niveau de zoom. Au sein du fichier enhanced.css, on utilise les media queries pour adapter les différents positionnements des blocs de la grille fluide :

```
@media screen and (min-width: 40.5em) and (max-width: 60em) {
  .app-img {
    width: 40%;
  }
  .app-main > header, .app-desc, .app-form, .app-description {
    margin-left: 40%;
    padding-left: 1em;
  }
}

@media screen and (min-width: 60em) {
  .app-img {
    width: 45%;
  }
  .app-main > header, .app-desc, .app-form, .app-description {
    padding-left: 2em;
    margin-left: 45%;
  }
  .img-container {
    margin-left: 7em;
  }
  .app-img nav {
    position: absolute;
    top: 0;
    left: 0;

```

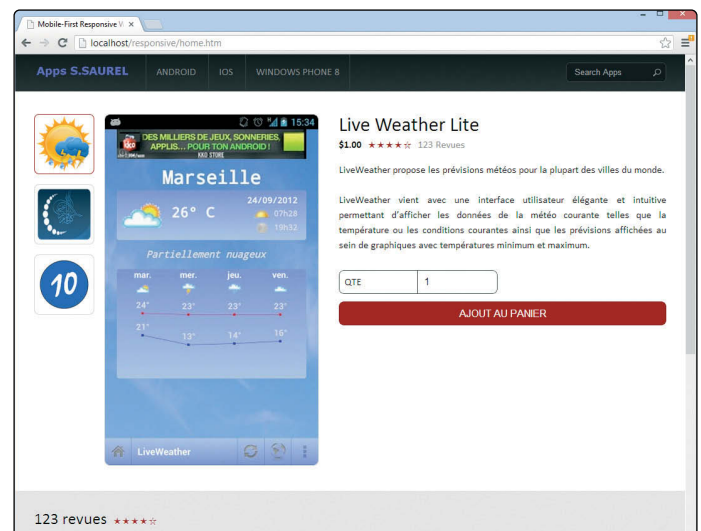
```
width: 6em;
}
.app-img nav li {
  display: block;
  margin: 0 0 1em;
}
.app-img nav img {
  width: auto;
}
.submit-form input {
  position: relative;
  z-index: 1;
}
}
```

Ces styles viennent surcharger les styles de base définis pour les mobiles au sein du fichier style.css. La grille fluide utilisée permettant de passer d'un affichage avec une seule colonne sur petits écrans à un affichage sur plusieurs colonnes pour grands écrans. Autre élément clé du Responsive Design, la gestion des images adaptables. Dans notre cas, il est nécessaire d'adapter la taille des copies d'écran des applications en fonction du contexte d'affichage. Cette adaptation est réalisée grâce aux media queries au sein de la classe app-img affichée plus haut.

> Comportement

Le comportement dynamique du site se fait via Javascript. L'intérêt est de pouvoir adapter le menu de navigation suivant l'espace disponible à l'affichage. Dans le cas de grands écrans, le menu est affiché en entier alors que pour les petits écrans il est divisé sur 2 niveaux. Ceci est réalisé via JQuery avec un listener de redimensionnement de la fenêtre parente :

```
$(w).resize(function(){
  sw = document.body.clientWidth;
  sh = document.body.clientHeight;
  checkMobile();
});
```



Affichage sur desktop



La fonction déterminant si le terminal hôte est mobile se base sur un point de rupture arbitraire pour la largeur d'écran. Il suffit ensuite de réagir en affichant le menu de navigation complet ou non :

```
function checkMobile() {
  mobile = (sw > breakpoint) ? false : true;
  if (!mobile) {
    loadAux();
    $('.aux header a').addClass('disabled').addClass('open');
    $('[role="tabpanel"]#nav,#search').show();
  } else {
    if(!$('#nav-anchors a').hasClass('active')) {
      $('#nav,#search').hide();
    }
  }
}
```

Toujours dans l'esprit mobile-first, le Javascript permet d'effectuer un chargement dynamique du contenu des commentaires pour les terminaux mobiles pour lesquels le débit internet peut être un problème. Ainsi, les commentaires ne sont chargés que lorsque le visiteur clique explicitement dessus à l'aide de la fonction loadAux qui est appelée à l'ouverture pour les terminaux desktop :

```
function loadAux() {
  var $aux = $('.aux');
  $aux.each(function(index) {
    var $this = $(this),
        auxLink = $this.find('a'),
        auxFragment = auxLink.attr('href'),
        auxContent = $this.find('[role=tabpanel]');
    if(auxContent.size()===0 && $this.hasClass('loaded')===false){
      loadContent(auxFragment,$this);
    }
  });
}

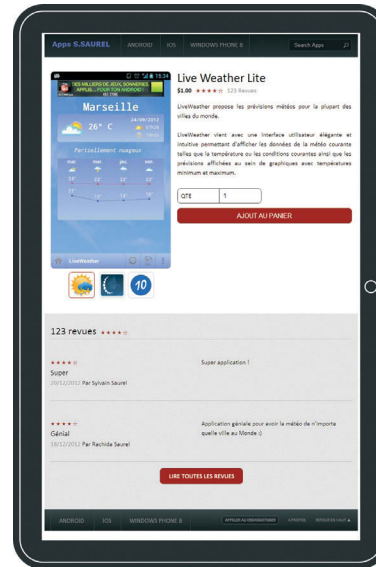
function loadContent(src,container) {
  container.addClass('loaded');
  $('<div role="tabpanel" />').load(src + ' #content > div',
  function() {
    $(this).appendTo(container);
  });
}
```

Les commentaires chargés dans la page sont définis au sein d'une page reviews.html et chargés via un appel AJAX réalisé au sein de la fonction loadContent.

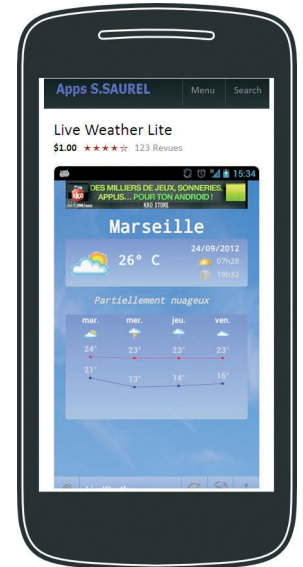
> Rendu

Le test sur différents supports de ce premier site permet de prendre conscience de la puissance de l'approche Responsive Design et des technologies mises en œuvre puisqu'en quelques lignes de code, on obtient déjà une solution répondant à la problématique d'affichage multiplateforme.

Les améliorations possibles sont nombreuses et l'on citera la possibilité d'utiliser la bibliothèque Javascript Modernizr pour détecter le support ou non du tactile sur le navigateur et d'ajouter le cas



Affichage sur tablette en mode portrait



Affichage sur smartphone

échiant la bibliothèque SwipeJS pour rendre les transitions entre images plus fluides. Enfin, des optimisations en termes de performances demeurent possibles, que ce soit via une bibliothèque Javascript plus légère que JQuery telle que Zepto ou en continuant l'effort de réduction du nombre de requêtes HTTP. En ce sens, on remarquera l'affichage des étoiles pour les revues en utilisant des caractères HTML spéciaux ou l'emploi à bon escient des gradients CSS à la place d'images.

> Les limites du Responsive Design

En adoptant le Responsive Design avec une approche mobile-first, le contenu d'un site web se concentre sur l'essentiel et se retrouve du coup mieux découpé. Néanmoins, le Responsive Design connaît également ses limites avec un affichage des publicités chahuté par les redimensionnements.

En outre, un site Responsive étant le même quel que soit le support, il est primordial de passer par une approche mobile-first pour avoir une page la plus légère possible pour soulager la connexion internet des mobiles. En ce sens, basculer un site n'ayant pas été conçu à l'origine dans cette optique peut s'avérer une mauvaise idée en termes de performances.

Enfin, le support multi-navigateurs est problématique sur les navigateurs les plus anciens ne supportant pas CSS 3 et HTML 5. Pour ceux-là (comme Internet Explorer 7 ou 8), le Responsive Design est plus difficile à mettre en place même si des outils comme Respons.js ou css3-mediaqueries.js sont des alternatives.

> Conclusion

La mise en pratique du Responsive Web Design aura mis en lumière la puissance de l'approche.

Les avantages et les promesses apportés par une approche mobile-first ne devant toutefois pas cacher les limites de la pratique qui nécessite une réflexion certaine en amont. Néanmoins, la réduction des coûts induite par un seul site auto-adaptable est un élément à mettre au crédit du Responsive Design.

Sylvain Saurel – Ingénieur d'Etudes Java / JEE
sylvain.saurel@gmail.com



© istockphoto/skynisher

Scrum

L'agilité aide le développeur au quotidien

Parler d'agilité, de bonnes pratiques, de méthodes agiles n'est jamais simple.

L'agilité est pratiquée par de plus en plus d'équipes. Les SSII l'utilisent comme soupe méthodologique. Difficile de travailler en équipe sans la moindre bonne pratique menant à l'agilité. Et pourtant... tout cela ne déclenche guère de passion de la part des développeurs.

Même sans forcément l'utiliser au quotidien, l'agilité aide le développeur à structurer son travail, à lui donner un cadre d'organisation du quotidien, à intégrer des notions simples comme l'intégration continue, le référentiel, les itérations courtes, les tests, comment gérer un bugtrack. C'est aussi une véritable philosophie. Ensuite, il ne faut pas non plus suivre aveuglément une méthode agile. Elle est justement faite pour être souple et s'adapter à chaque situation, même si le socle reste le même tout le temps. D'ailleurs Il existe des méthodes plus flexibles que d'autres. L'eXtreme Programming,

par exemple, est idéal pour aller très vite et pour travailler à deux. En revanche, cette approche n'est pas adaptée à un travail en équipe, ni sur des projets importants.

Aujourd'hui, Scrum est sans aucun doute la méthode la plus utilisée. Et on peut y piocher les pratiques qui conviennent le mieux pour soi et pour l'équipe.

Ce mois-ci, nous vous proposons un grand dossier Scrum et ScrumMaster. De nombreux développeurs, experts agiles et ScrumMasters expliquent Scrum, comment l'adapter et l'utiliser, les bonnes pratiques et celles à éviter. A la fin du dossier, vous pourrez mieux comprendre le rôle du ScrumMaster et du Product Owner, les deux colonnes vertébrales de Scrum au quotidien.

Bonne agilité !

François Tonic



L'agilité et Scrum dans la vie du développeur

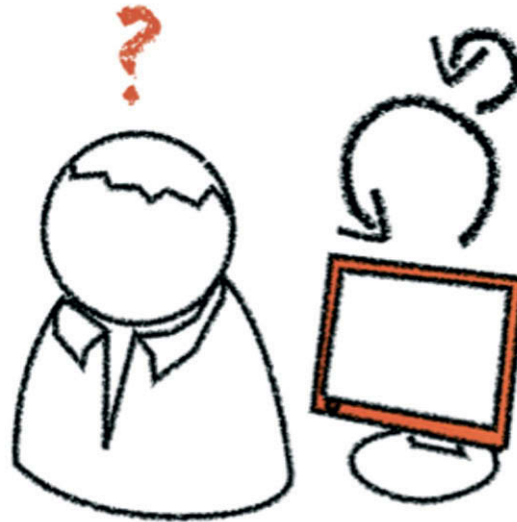
Non, l'agilité n'est pas simplement un mot à la mode qui signifie devoir tout accepter, à tout moment, pour un développeur. Non, l'agilité ne permet donc pas à un chef de projet ou un ScrumMaster de changer à tout moment l'ordre des tâches sur lesquelles travaillent les équipes de développement. Et enfin non, l'agilité ne permet pas systématiquement de développer plus avec moins, en masquant un joyeux chaos qui se nomme plus simplement « à l'arrache ».

Pour résumer l'agilité, je dirais en revanche qu'elle permet de lever les problèmes rencontrés plutôt que de les cacher, qu'elle permet de trouver des espaces de dialogue et d'amélioration pour les acteurs du projet, et surtout, qu'elle permet d'avoir un réel droit de regard sur les conditions de travail au jour le jour des développeurs, pour enfin rendre le projet « raisonnable ». En ce sens, oui les méthodes agiles changent radicalement les façons de travailler des développeurs et leur rôle au sein d'une équipe projet.

> Vous avez dit agilité ?

Commençons par savoir ce que l'on entend par « agilité ». Les méthodes agiles ont pour but principal de rendre plus pragmatique la conduite d'un projet et de maximiser la satisfaction du client, tout en s'adaptant aux changements survenant lors de l'élaboration du projet.

A la différence des projets dits prédictifs ou traditionnels comme le « cycle en V » ou le « modèle en cascade », on ne procède plus en étapes successives, qui traitent des moindres détails, planifiés dès le départ et sur toute la durée d'un projet. Les étapes conception, spécification, développement, test et déploiement sont donc complètement revisitées avec les méthodes agiles pour tenter d'éviter les effets tunnel, les problèmes de mauvais chiffrage, les changements imprévus en cours de route ou l'insatisfaction client devant le produit terminé. Née du milieu du développement, l'agilité a été formalisée à l'aide d'un certain nombre de valeurs qui précisent sa philosophie, à



- ✓ Les individus et leurs interactions plus que les processus et les outils.
- ✓ Des logiciels opérationnels plus qu'une documentation exhaustive.
- ✓ La collaboration avec les clients plus que la négociation contractuelle.
- ✓ L'adaptation au changement plus que le suivi d'un plan.

Fig.1

Les 4 valeurs du Manifeste agile

travers le manifeste agile écrit en 2001, par une poignée de passionnés.

Ces valeurs, au nombre de quatre, mettent en avant l'écoute et la communication au sein même de l'équipe, garantissent une qualité optimale et visible du logiciel au cours de sa création, prônent une interaction forte avec le client tout au long du projet et imposent d'intégrer le changement dans nos habitudes de travail.

Pour appliquer ces différentes valeurs, des méthodes ont vu le jour et permettent de donner des pistes, des axes à suivre pour gérer un projet en mode agile. Parmi les méthodes les plus connues, nous pouvons citer Scrum, Kanban ou l'eXtrem Programming. Attention donc à ne pas faire d'amalgame en résumant l'agilité à Scrum et vice-versa [Fig.1].

Quelle que soit la méthode retenue, la discipline, l'amélioration continue, la transparence et une clarification des rôles, sera de mise pour assurer la conduite d'un projet agile réussi. Ces méthodes, radicalement différentes des méthodes prédictives (comme le modèle en cascade ou le cycle en V), bouleversent la vision même d'un projet en changeant les rôles et les façons de faire de chacun, et notamment des développeurs.

> Oui, mais concrètement ?

Pour essayer de voir les changements apportés dans la vie de tous les jours pour les développeurs, prenons comme exemple la méthode Scrum. Sans retracer tous les détails proposés par cette méthode,

je vous propose de voir les différences qu'elle amène par rapport aux méthodes dites classiques, à travers l'œil du développeur.

> Le développeur s'engage lui même

Pour commencer, les développeurs estiment leur propre travail. En Scrum comme en agilité en général, le principe est de faire confiance à ceux qui réalisent pour l'estima-

tion de la charge de travail. Comprenez donc qu'un chef de projet, un ScrumMaster, un Product Owner ou tout autre rôle n'estimera pas à la place du développeur la charge de son travail qui sera à réaliser au cours des prochains jours. A chaque nouvelle itération (que l'on nomme « sprint » en Scrum et qui dure au maximum 4 semaines), l'équipe estime donc ce qu'elle est capable de faire et s'engage à réaliser différentes parties fonctionnelles de l'application (des « User stories ») selon la priorité décidée par le représentant du client, le Product Owner.

Exit donc le système de chiffrage réalisé uniquement par un chef de projet qui découpe et impose les tâches à chacun dans un modèle traditionnel. Il s'agit ici de repositionner les responsabilités de chacun et de s'y tenir en rétablissant la confiance au sein d'une équipe projet.

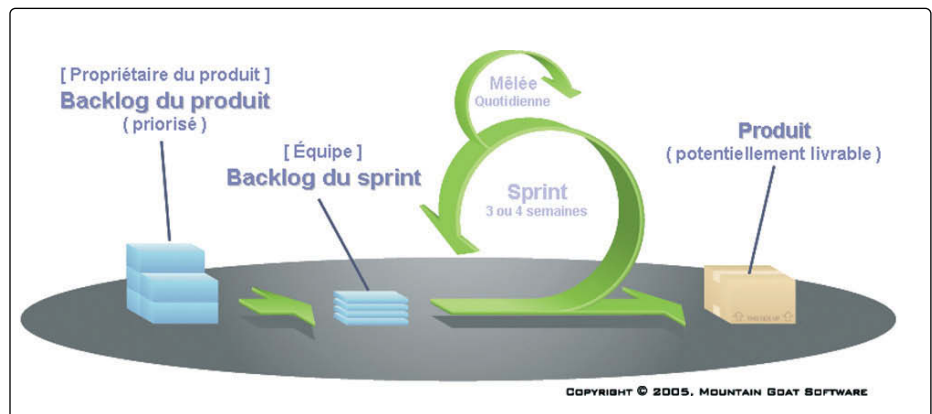
> Auto-organisation

Avec Scrum, l'équipe est auto-organisée. Elle s'organise donc comme elle le souhaite, pour réaliser ce qui a été engagé à l'issue de ces quelques semaines qui composent ce sprint. Ce principe invite notamment les développeurs à échanger entre eux via des réunions journalières (les « Daily stand-up meeting » par exemple), à partager leur savoir-faire devant un « White Board », à demander des précisions au Product Owner tout au long du sprint ou à s'entraider si besoin via différentes pratiques comme, par exemple, le « binôme ». Libre à l'équipe également de s'employer aux « Test Driven Development » (le TDD), à l'automatisation des tests fonctionnels, à avoir une couverture probante des tests unitaires ou à utiliser l'intégration continue, pour pouvoir garantir par la suite une organisation efficace qui délivre un travail de qualité [Fig.2].

La principale idée de l'auto-organisation est donc de se concentrer sur la finalité des fonctionnalités voulues plutôt que de définir précisément la manière de les réaliser. Cette façon de procéder et de s'organiser, permet aux développeurs d'être de véritables acteurs du projet, en participant activement à son élaboration interne.

> Le développeur n'est plus isolé

Toujours en Scrum, les développeurs pourront assurer en fin de



sprint, une démonstration du travail réalisé. Ce temps privilégié qui réunit décideurs et développeurs permet de valoriser le travail de l'équipe, en exposant les User Stories réalisées durant le dernier sprint. Le développeur se trouve donc au centre de la scène et parle à des personnes qu'il ne voyait généralement pas pendant des mois avec des méthodes plus traditionnelles.

De manière générale, la transparence est également de mise avec Scrum. Les développeurs affichent continuellement l'avancement, problèmes et travail en cours de réalisation. A travers un « white board », un « burndown chart », un wiki ou même la démonstration qui est ouverte à tous, les développeurs exposent leur travail pour favoriser les échanges et éviter au maximum les zones d'ombres.

> Prendre le temps de bien faire les choses

Pour finir, la rétrospective permet à chaque fin de sprint, de marquer un temps d'arrêt entre deux sprints. Avec Scrum, les sprints s'enchaînent les uns à la suite des autres et il est indispensable de prendre un peu de temps pour s'auto-améliorer. Des bugs a

répétitions qui arrivent ? Une information que l'on a du mal à récupérer ? Du refactoring à effectuer ? Des difficultés ? Ou même étudier un nouveau framework qui permettrait de faciliter les développements ?

Tous les sujets sont bons à aborder pendant ce laps de temps pour ensuite trouver des actions qui permettront, sans surcoût de travail, d'améliorer continuellement le confort de travail de chacun.

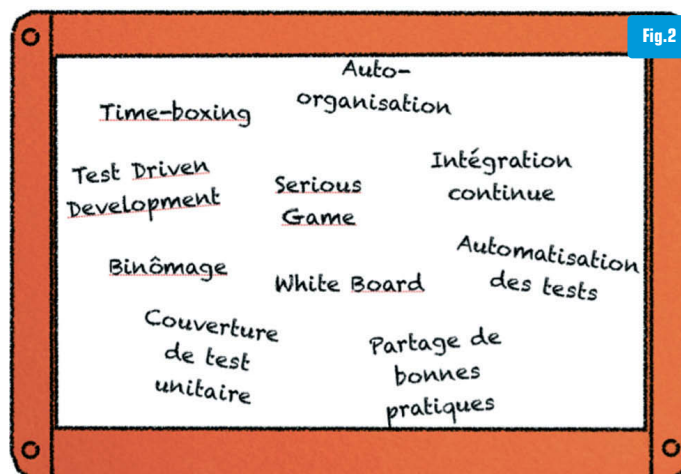
Ces espaces de temps, que l'on réserve à l'équipe pour s'améliorer sur le plan technique ou organisationnel est obligatoire : il participe à une qualité raisonnée du projet et autorisera un rythme soutenable à l'équipe dans le temps.

> Une redéfinition des rôles

Comme vous pouvez le constater, ces façons de travailler diffèrent quelque peu des méthodes traditionnelles.

Là où un chef de projet traditionnel, chiffre, priorise et impose les différentes tâches pour chaque membre de son équipe, Scrum déporte en partie cette responsabilité sur l'équipe qui possède un droit de regard sur ce qu'elle a réalisé. Du côté Product Owner, l'engagement est pris de rédiger ces user story de manière claire et priorisée et d'être disponible pour répondre aux questions émanant de l'équipe.

Ainsi, ce système permet de recentrer le travail de chacun : les développeurs développent sur ce qui leur semble possible et le Product Owner s'occupe de définir les fonctionnalités les plus attendues. Les changements de priorités se font entre chaque sprint, et non pendant, ce qui permet d'assurer un minimum de perturbation pour l'équipe qui avance de manière plus réguliè-





re. Cela permet donc de mesurer l'avancement d'un projet de manière plus réaliste et rend raisonnable le travail de chacun. Pour autant, la recette n'est pas magique et pour que ça marche, il faut quelques ingrédients qui n'ont rien de miraculeux.

> Le rôle de ScrumMaster

La discipline et le respect des règles du jeu sont des aspects primordiaux dans toute méthode agile et notamment dans Scrum. Vous ne pouvez pas avoir de métriques réelles si vous ne faites pas de sprints réguliers, vous ne pouvez pas enchaîner les sprints si vous n'avez pas une qualité irréprochable et vous ne pouvez pas mesurer votre vitesse d'avancement si vous « cassez » l'engagement de l'équipe au cours d'un sprint.

Pour assurer une cohérence de ces règles, le rôle du ScrumMaster est l'une des clés que propose Scrum pour assurer cet équilibre subtil.

A la différence d'un chef de projet, le ScrumMaster n'a pas de relation hiérarchique avec les développeurs et se trouve faire partie intégrante de l'équipe qu'il anime. Le plus souvent, le ScrumMaster est un ancien développeur, qui doit posséder des compétences de communication, d'écoute, de maîtrise technique et doit savoir motiver et guider sans imposer. Il s'apparente finalement à un coach d'équipe dont les objectifs sont triples :

> Un garant de la méthode

Il va tout d'abord assurer la cohérence des sprints en gardant les métriques et outils de suivi à jour, en assurant une communication auprès des différents acteurs du projet et en proposant des solutions si le sprint rencontre des difficultés.

Tout ceci implique de garder une certaine forme de recul sur chaque sprint pour en déceler les axes d'amélioration. Générale-

ment, c'est également lui qui veille à la régularité et au bon déroulement des meetings prévus par la méthode.

> Un facilitateur

Son périmètre inclut également un rôle de facilitation. A savoir, aider à définir à un niveau raisonnable les actions qui suivent une rétrospective par exemple.

Il peut également aider à la récupération d'informations utiles pour l'équipe, pour éviter de lui faire perdre un peu de temps sur cette recherche qui peut



s'avérer parfois difficile. La mise en place d'outils à destination de l'équipe (de l'intégration continue par exemple) peut également entrer dans ce périmètre, afin là aussi de faciliter le travail de l'équipe au quotidien. Ce catalyseur d'informations et d'outils qu'est le ScrumMaster permet à l'équipe de ne se concentrer que sur son propre travail.

> Un protecteur

Pour finir, le ScrumMaster est aussi un protecteur, en s'assurant que l'équipe ne soit pas coupée ou dérangée pendant la réalisation d'un sprint, afin de ne pas casser son engagement. Fini donc les surcharges de travail inattendues et urgentes qui arrivent à 18h30 une veille de week-end. L'idée est ici de se tenir à ce qui a été défini et engagé par l'équipe en début de sprint et d'attendre un sprint prochain pour proposer de toutes

nouvelles fonctionnalités. De cette manière, il est donc proscrit d'ajouter ou modifier le travail engagé par l'équipe durant un sprint afin de garder un confort de travail qui rend « raisonnable » le travail des développeurs.

> Scrum comme outil

Ce changement de façon de voir et concevoir un projet ne se limite pas uniquement aux développeurs, les décideurs doivent également changer leurs habitudes de travail pour aider à créer ce nouveau cadre.

Scrum ainsi que les autres méthodes agiles propose une piste, un axe d'organisation qui se doit d'être adapté au contexte et problématiques de chaque projet.

Scrum peut donc être considéré comme un outil, à bien comprendre avant d'être utilisé et qui n'a rien de magique.

Aidés par le ScrumMaster, les développeurs d'une équipe Scrum pourront ainsi librement s'engager sur leurs propres tâches et auront les outils nécessaires pour assurer leur réalisation.

Néanmoins, ce confort de travail a un coût, celui de la discipline. Assurées par des ateliers (des « Agiles games »), préconisées par un coach agile ou définies par tous les acteurs avertis d'un projet, des règles du jeu claires doivent trouver leur place concernant les rôles, les outils et les processus utilisés. De cette manière et en gardant à l'esprit que tout peut être amélioré continuellement, ces règles permettront d'assurer une réelle cohérence agile sur votre projet, qui rendront efficaces et agréables vos journées de développement au sein d'une équipe agile.

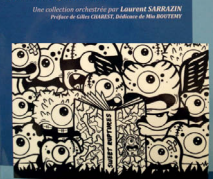


Axel Villechalane

Coach et formateur agile au sein de la société Soat dont il dirige l'activité agile. Co-auteur du livre « Rupture Douce » qui traite des transformations agiles à grande échelle, membre du French Scrum User Group

Rupture Douce

Petites histoires de Grenouilles Agiles, à l'usage des détracteurs de néo-mondes organisationnels, des métamorphoseurs d'équipes, des ostéopathes de projets à miraculer, bref... des agilitateurs !



Agilité

RUPTURE DOUCE : le manuel des « agilitateurs »

Axel Villechalane est un des auteurs de l'ouvrage collectif « Rupture Douce », rédigé par une «tribu» d'«agilitateurs». Des dessins et schémas, un ton drôle et profond, vous ouvrez le livre

au hasard et dévorez une des 25 "histoires" : généralement un cas réel d'entreprise raconté avec précision et une touche de dérision. Le premier chapitre : «Sweet rupture», de Laurent Sarrazin raconte le projet d'une grande banque. Un autre article de Laurent Sarrazin, «facilita-

tion Lud'innovante» propose une approche ludique pour créer un réseau de volontaires, ambassadeurs, promoteurs, pour faire décoller le changement. Car c'est bien de changement dont il s'agit, les auteurs se présentant comme des «métamorphoseurs d'équipes», des «ostéopathes de projets».

Disponible au format e-book : 6,99€. Disponible également au format papier.
<http://www.lulu.com/shop/laurent-sarrazin/rupture-douce/ebook/product-20518255.html>

Jean Kaminsky

Utiliser (concrètement) Scrum

Les logiciels sont de plus en plus complexes, et changent régulièrement, surtout en phase de développement. Du coup, les méthodologies de développement traditionnelles ont montré des limites devant ces évolutions, surtout pour les grands projets.

En effet, avec les méthodologies classiques, il n'y a pas d'interaction entre le client (propriétaire du logiciel ou produit) et l'équipe de développement. Le client ne voit le résultat qu'à la livraison finale et rien, ou presque rien, pendant toute la phase de développement. D'où l'effet tunnel. Cet effet tunnel en lui-même génère un autre risque au projet. C'est le risque de modification ou d'ajout de fonctionnalités. Avec les méthodologies classiques, ce genre de modifications, qui sont demandées par le client, surviennent généralement à une phase très avancée du projet où toute modification est très risquée. Une des méthodes qui s'est distinguée et qui a montré son efficacité face à cette évolution est la méthodologie agile ou «Scrum».

> A quoi ressemble une équipe Scrum ? Quel est le rôle de chacun ? Qui fait quoi ?

L'équipe Scrum (dans l'idéal, NDLR) est constituée d'un **Product Owner** (propriétaire de produit), de l'**équipe de développement** et d'un **ScrumMaster**.

Le **Product Owner** est le client, il est chargé de maximiser la valeur du produit et du travail de l'équipe de développement. Cela se concrétise d'une manière qui varie considérablement selon les organisations, les équipes Scrum et les individus. Le Product Owner est la seule personne responsable de la gestion du backlog de produit.

La gestion du backlog de produit consiste à :

- Exprimer clairement les items du backlog de produit
- Hiérarchiser les items du backlog de produit pour mieux atteindre les objectifs et les missions
- S'assurer de la valeur du travail que l'équipe de développement réalise
- S'assurer que le backlog de produit est visible, transparent et clair pour tous et qu'il décrit bien le travail à venir pour l'équipe.
- S'assurer que l'équipe de développement comprend suffisamment les items du backlog de produit.

L'**équipe de développement** est composée de professionnels qui livrent à chaque sprint

une build du produit. Seuls les membres de l'équipe de développement réalisent ce livrable.

Les équipes de développement ont les caractéristiques suivantes :

- Elles sont auto-organisées. Personne (pas même le **ScrumMaster**) ne peut dicter à l'équipe comment traduire le backlog de produit en incréments de fonctionnalités potentiellement livrables.
- Elles sont pluridisciplinaires, ayant toutes les compétences nécessaires pour mettre en œuvre un livrable du produit.
- Individuellement, les membres d'une équipe peuvent être spécialisés dans certaines compétences ou dans certains domaines, mais la responsabilité appartient à l'équipe de développement dans son ensemble.
- Les équipes de développement ne contiennent pas de sous-équipes dédiées à un domaine en particulier comme les tests ou l'analyse métier (fonctionnelle).

La taille d'une équipe de développement doit être optimale :

- assez petite pour demeurer agile
- assez grande pour effectuer du travail significatif.

Moins de trois membres dans l'équipe de développement diminue les interactions et entraîne des gains de productivité moindres. Une équipe de développement trop petite peut rencontrer des contraintes de compétences pendant le sprint, menant l'équipe à être incapable de terminer un incrément potentiellement livrable.

Plus de neuf membres exige trop de coordination. Les grandes équipes de développement génèrent trop de complexité pour être gérées par un processus empirique. Les rôles du Product Owner et du ScrumMaster ne sont pas inclus dans ce nombre, sauf s'ils participent également à la réalisation des travaux du backlog de sprint.

Le **ScrumMaster** est responsable de la compréhension et de l'application de Scrum. Pour cela, il ou elle s'assure que l'équipe Scrum adhère aux valeurs, pratiques et règles de Scrum. Le rôle de **ScrumMaster** est celui de meneur au service de l'équipe Scrum « Servant Leader ».

Le ScrumMaster sert le Product Owner de plusieurs façons, notamment :

- En trouvant des techniques pour la gestion efficace du backlog de produit
- En communiquant clairement la vision, les objectifs et les items du backlog de produit à l'équipe de développement
- En enseignant à l'équipe de développement comment créer des items de backlog clairs et concis
- En comprenant la planification à long terme du produit dans un environnement empirique
- En comprenant et pratiquant l'agilité
- En facilitant les événements Scrum à la demande ou lorsque c'est nécessaire.

Le ScrumMaster sert l'équipe de développement de plusieurs façons, notamment :

- En aidant l'équipe de développement à apprendre comment s'auto-organiser et développer sa transversalité
- En enseignant et en menant l'équipe de développement à livrer des produits de haute valeur
- En supprimant les obstacles nuisant au progrès de l'équipe de développement
- En facilitant des événements Scrum à la demande ou lorsque c'est nécessaire
- En accompagnant l'équipe de développement dans les environnements organisationnels dans lesquels Scrum n'est pas encore complètement adopté et compris.

> Mais quels sont les événements Scrum ?

Des événements prescrits sont utilisés dans Scrum afin de créer de la régularité et de minimiser la nécessité de réunions qui ne sont pas définies dans Scrum.

Ces événements sont :

- Réunion de planification de sprint
- Le sprint
- Daily meeting (mêlée quotidienne)
- Revue de sprint
- Rétrospective de sprint.

Réunion de planification de sprint

Les travaux à effectuer durant le sprint sont planifiés lors de la réunion de planification de sprint. Ce plan est créé par la collaboration



de toute l'équipe Scrum. La réunion de planification de sprint est, généralement, un bloc de temps de huit heures

La réunion de planification de sprint se compose de deux parties, chacune utilisant la moitié du temps total alloué pour la planification. Les deux parties de la réunion de planification de sprint répondent aux questions suivantes :

• **Qu'est-ce qui sera livré dans l'incrément résultant du prochain sprint ?**

Dans cette partie, l'équipe de développement choisit les fonctionnalités qui seront livrées au cours du sprint. Le Product Owner présente à l'équipe de développement les items hiérarchisés du backlog de produit et toute l'équipe Scrum travaille alors ensemble sur la compréhension du travail à effectuer dans le sprint.

Les items sélectionnés du backlog de produit pour ce sprint sont appelés **backlog de sprint**

• **Comment le travail nécessaire pour réaliser l'incrément sera-t-il accompli ?**

Après avoir choisi le travail du sprint, l'équipe de développement décide comment elle va réaliser, durant le sprint, un **livrable** (incrément de produit).

L'équipe de développement décompose le backlog de sprint en tâches et s'auto-organise pour estimer et dispatcher ces tâches entre les membres.

Une fois les tâches estimées, l'équipe aura une visibilité sur la durée du sprint et estime la date de fin du sprint.

Le Product Owner peut être présent lors de la deuxième partie de la réunion de planification de sprint pour clarifier les items sélectionnés du backlog de produit et afin de contribuer à trouver des compromis. Si l'équipe de développement détermine qu'elle a trop de travail, ou bien qu'elle n'en a pas assez, elle peut renégocier avec le Product Owner les items du backlog de sprint. L'équipe de développement peut également inviter d'autres personnes à la réunion afin de fournir des conseils techniques ou sur le domaine (consultants métiers, ...).

Daily Meeting (Mêlée quotidienne) :

Le daily meeting est une réunion limitée à un bloc de temps de 15 minutes et destinée à permettre à l'équipe de développement de synchroniser ses activités et planifier les prochaines 24 heures. Pour ce faire, le tra-

vail réalisé depuis le dernier daily meeting est inspecté et une prévision du travail qui pourra être réalisé avant la prochaine mêlée est énoncée. Le daily meeting a lieu tous les jours à la même heure et au même endroit afin de réduire la complexité. Pendant la réunion chaque membre de l'équipe de développement décrit :

- Ce qu'il a réalisé depuis la dernière réunion ;
- Ce qu'il réalisera avant la prochaine réunion ;
- Les difficultés qu'il rencontre.

L'équipe de développement se sert du daily meeting pour mesurer l'atteinte de l'objectif du sprint ainsi qu'évaluer comment se fait la progression vers la finalisation du travail du backlog de sprint.

Le ScrumMaster s'assure que le daily meeting a lieu mais c'est l'équipe de développement qui est responsable du déroulement de la réunion. Le ScrumMaster apprend à l'équipe de développement comment limiter le daily meeting à un bloc de temps de 15 minutes.

Le ScrumMaster veille à l'application de la

“ Les daily meetings améliorent la communication et la compréhension du projet par l'équipe de développement ”

règle stipulant que seuls les membres de l'équipe de développement participent au daily meeting. Le daily meeting n'est pas une réunion d'avancement et est destinée aux personnes en charge de transformer les items du backlog de produit en fonctionnalités livrables.

Les daily meetings améliorent la communication, réduisent le nombre de réunions, identifient et suppriment les obstacles qui perturbent le développement, mettent en avant et encouragent la prise de décision rapide et améliorent la compréhension du projet par l'équipe de développement. Il s'agit d'une réunion d'inspection et d'adaptation primordiale.

La Revue de sprint :

La revue de sprint est une réunion qui se déroule à la fin de chaque itération et comprend les éléments suivants :

- Le Product Owner identifie ce qui a été terminé et ce qui n'a pas été terminé
- L'équipe de développement discute de ce qui s'est bien déroulé durant le sprint, quels problèmes ont été rencontrés, et

comment ces problèmes ont été résolus

- L'équipe de développement démontre le travail terminé et répond aux questions sur l'incrément
- Le Product Owner discute du backlog de produit tel qu'il est. Il ou elle détermine des dates probables d'achèvement en fonction des progrès à ce jour
- L'ensemble du groupe convient de ce qu'il faut faire pour la suite, de sorte que la revue de sprint fournisse une contribution précieuse aux réunions subséquentes de planification de sprint

Le résultat de la revue de sprint est un backlog de produit révisé qui définit les items probables pour le prochain sprint. Le backlog de produit peut également être complètement revu pour répondre à de nouvelles opportunités.

Rétrospective de sprint

La rétrospective de sprint est une occasion pour l'équipe Scrum de s'inspecter et de créer un plan d'améliorations qui sera mis en place au cours du sprint suivant.

Le but de la rétrospective de sprint est :

- D'inspecter la manière dont le dernier sprint s'est déroulé en ce qui concerne les personnes, les relations, les processus et les outils
 - D'identifier et ordonner les éléments majeurs qui se sont bien déroulés et les améliorations potentielles
 - De créer un plan pour améliorer les processus de travail de l'équipe Scrum.
- Le ScrumMaster encourage l'équipe Scrum à améliorer, dans le cadre du processus Scrum, son processus de développement et ses pratiques afin de les rendre plus efficaces et agréables pour le prochain sprint. Lors de chaque rétrospective de sprint, l'équipe Scrum planifie des moyens adéquats d'accroître la qualité du produit, en adaptant sa définition de 'terminé'.

À la fin de la rétrospective de sprint, l'équipe Scrum devrait avoir identifié les améliorations qu'elle mettra en œuvre durant le prochain sprint. La mise en œuvre de ces améliorations au cours du prochain sprint est l'adaptation à l'inspection de l'équipe de développement elle-même. Bien que des améliorations puissent être mises en œuvre à tout moment, la rétrospective de sprint fournit un événement dédié et axé sur l'inspection et l'adaptation.

A vous d'être agile maintenant !

Ahmed Amine **Mezri**
Ingénieur Concepteur IPTECH (Tunisie)

Du Slow Scrum à l'hyperproductivité : le rôle et les outils du ScrumMaster

Lors de son speech à la soirée de lancement de l'association Agile.NET France en décembre dernier, Jeff Sutherland, co-fondateur de Scrum, a parlé d'un « turning point », aux Etats-Unis. Il y aurait maintenant plus d'offres d'emploi en méthodes agiles là-bas que sur les méthodes traditionnelles de gestion de projets informatiques.

Bien que la France ne soit pas encore à ce niveau de maturité, la tendance est néanmoins forte et – espérons-le – durable. Jeff Sutherland parle couramment d'équipes ayant adopté Scrum, dont la vélocité aurait été multipliée par 2, 3, ou même 30. Seulement, force est de constater qu'une grande partie des équipes que l'on croise sur des projets ou durant des événements reste à un état plus proche du Slow Scrum que de l'hyperproductivité. Cet article s'attachera à analyser et étudier les différents leviers à la disposition du ScrumMaster pour améliorer significativement la vélocité de son équipe.

> Le rôle du ScrumMaster et sa place dans l'équipe

Dans sa forme la plus simple et la plus pure, Scrum dit que le ScrumMaster a pour but de faire respecter le processus Scrum. Le terme de facilitateur est donc celui qui revient le plus souvent. Il a aussi pour rôle de protéger l'équipe, en faisant office de tampon entre celle-ci et l'extérieur et en résolvant ou faisant résoudre les différents problèmes la ralentissant.

Récemment, un article du blog de Xebia <http://blog.xebia.fr/2013/02/08/le-scrum-master-est-le-majordome-de-lequipe>, le cousin germain javaïste mais lui aussi agiliste de Cellenza, comparait le ScrumMaster à un majordome. Et mentionnait les qualités nécessaires au « servant leader ».

Cette définition est donc à mon sens extrêmement élastique, en plus d'être parfois ambiguë. Au point qu'il m'est arrivé de discuter tout récemment avec des gens qui se demandaient quel était leur rôle dans l'organisation : étaient-ils ScrumMaster, Product Owner, Proxy Product Owner ? Ou encore autre chose... ?

Et il me semble que la réponse est finalement là : le ScrumMaster est un rôle **intrinsèquement élastique**. Son rôle, au final, doit



être celui dont a besoin l'équipe. On parlait un peu plus haut d'être un tampon entre l'équipe et l'extérieur. Mais il doit aussi être un piston.

Une équipe est un organisme vivant, composée elle-même d'organismes vivants, et est en contact permanent avec d'autres organismes vivants. Elle vit, évolue, et change avec le temps.

Le ScrumMaster doit donc être d'une grande flexibilité, afin d'être le plus efficace en permanence. Si l'équipe nécessite un leader, c'est à lui d'en trouver un, ou à défaut de prendre ce rôle. Son but ultime, à mon sens, doit être de se rendre **totale-ment dispen-sable**. S'il parvient à ce but, c'est qu'il aura rempli l'ensemble de ses missions.

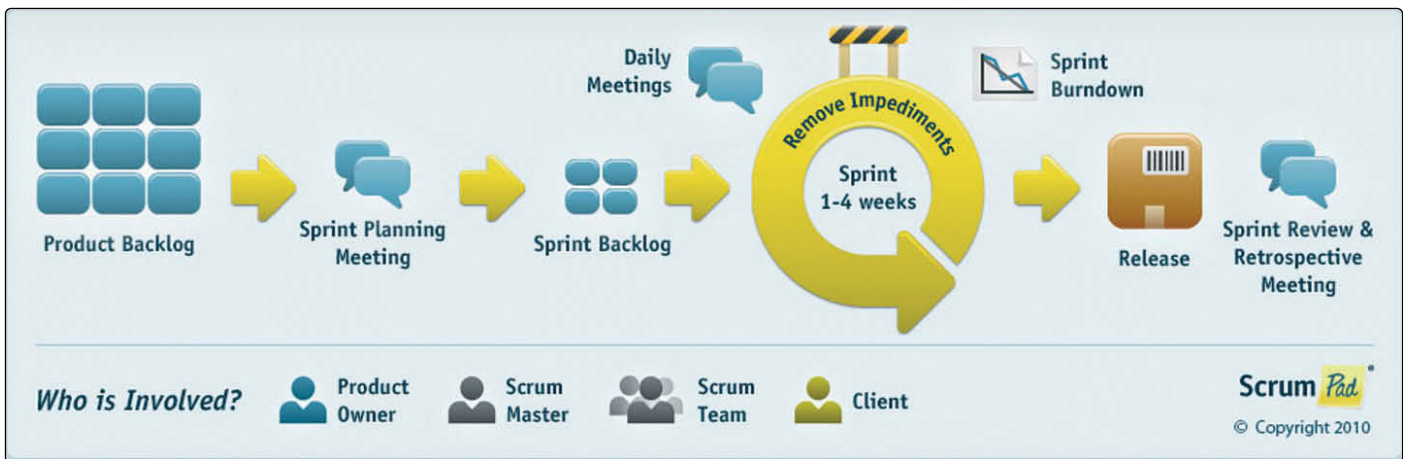
Les points de blocage auront été levés, son équipe travaillera de façon autonome et selon un rythme régulier, soutenu et soutenable. Elle aura maîtrisé les fondamentaux

de Scrum et les cérémonies se dérouleront de façon efficace.

A minima, le ScrumMaster est donc au service de son équipe, excellent communicant, volontaire et doté d'un certain sens du sacrifice. Ses autres casquettes possibles peuvent être développeur, testeur, architecte, administrateur, etc. Tant qu'il a envie d'assumer le rôle de ScrumMaster et que les membres de l'équipe lui reconnaissent les qualités pour ce rôle. En contrepartie, on doit lui donner les moyens d'agir pour le bien de l'équipe en termes de temps et de leviers d'action. Nous verrons des types de leviers plus loin dans l'article.

> Comprendre son équipe

Le ScrumMaster est au service de son équipe. Mais il ne peut pas se contenter de répondre aux problèmes levés explicitement par les membres de son équipe. Sans aller



jusqu'à en faire un psychologue, il doit porter une attention toute particulière aux personnalités des membres de son équipe mais surtout aux interactions entre ces derniers.

Les mécaniques de la vie d'une équipe peuvent être complexes, et il est primordial pour le ScrumMaster de les comprendre pour pouvoir non seulement en huiler efficacement les rouages, mais aussi de proposer des axes d'amélioration de façon proactive. L'exercice peut avoir des côtés schizophrènes. Car le ScrumMaster doit faire partie intégrante de l'équipe afin de comprendre les besoins et prendre du recul en permanence afin d'identifier ces axes d'amélioration et les réels problèmes remontés par l'équipe.

On ne le repètera jamais assez, mais il est primordial et même nécessaire d'être un excellent communicant pour ce rôle. Il faut parfois être intrusif et aller chercher les causes réelles d'un point de blocage, tout comme il faut parfois se rendre discret, voire transparent afin de laisser remonter

naturellement certaines informations. Une forte capacité d'adaptation, ainsi qu'une bonne dose d'empathie seront donc d'excellents alliés pour une forme d'audit permanent.

> Améliorer et faire améliorer son équipe

Scrum, d'après son père Jeff Sutherland est est un *framework*. Pas une méthode à proprement parler. Il n'y a donc aucune batterie d'outils clés en main pour améliorer la productivité et les conditions de vie de son équipe. En revanche, il y a une certaine liberté de mouvement, ainsi que le droit et le devoir d'expérimenter. Le premier rôle du Scrum-Master est de s'assurer du bon déroulement du processus. Il devra donc s'attacher à faire respecter les rôles, les artefacts et les cérémonies de Scrum.

Les rôles

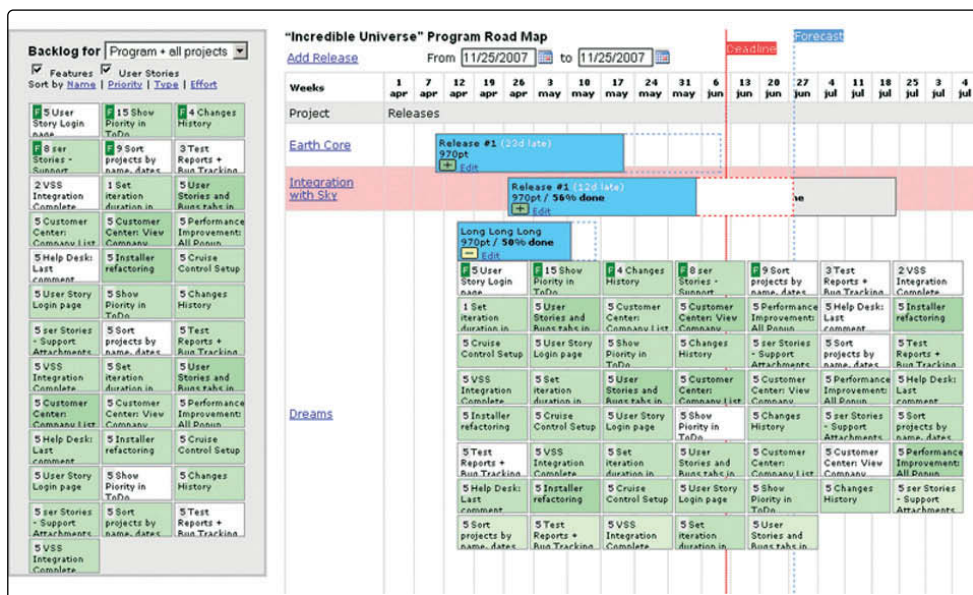
La première chose à faire est de maîtriser les tenants et aboutissants de son propre

rôle. Il veillera ensuite à s'assurer que le Product Owner et les Team Members comprennent les leurs. Dans une équipe Scrum débutante, tout comme parfois dans une équipe confirmée, le rôle de Team Member peut être dévoyé. Par définition, un **Team Member** n'est pas seulement un développeur. Il peut aussi être tour à tour concepteur, testeur, architecte, ergonomiste... Le ScrumMaster devra veiller à ce que les membres de l'équipe ne se spécialisent pas à 100%. Théoriquement, une tâche d'une User Story peut être confiée à n'importe quel membre de l'équipe.

Dans les faits, c'est souvent plus complexe, fonction des affinités, expériences et motivations de chacun. Néanmoins, il ne faut pas en arriver au point où chacun à « ses » stories attitrées, et surtout à ce que la connaissance soit partagée par tous les membres de l'équipe. Le **Pair Programming**, venant d'eXtreme Programming (XP) peut être un excellent outil pour cela. Il assure une répartition de la connaissance, mais peut aussi être utilisé pour faire monter en compétence un membre plus junior de l'équipe. Dans tous les cas, le code écrit n'est pas la propriété personnelle d'un seul développeur.

Le ScrumMaster sera aussi le plus souvent amené à travailler en étroite collaboration avec le **Product Owner**, afin de gérer le Backlog le plus parfait possible.

Ce dernier devra tout d'abord être sur site, et si possible à portée directe de l'équipe, de façon à fluidifier au maximum les échanges. Si l'on caricature, le Product Owner a pour but de faire sortir les fonctionnalités, le plus rapidement possible, alors que le Scrum-Master préserve son équipe. Les intérêts sont donc divergents. Tout le but ici est de travailler en bonne intelligence afin de satisfaire les besoins des deux parties. Ainsi, un binôme Product Owner/ScrumMaster effi-



cace est un atout non négligeable pour la réussite d'un projet.

Les artefacts

Le **Backlog** est la colonne vertébrale de Scrum. Il est donc à surveiller en toute priorité. Si le listage des fonctionnalités, leur priorisation et l'alimentation du Backlog sont de la responsabilité du Product Owner, le ScrumMaster a pour responsabilité de s'assurer que les User Stories en tête de liste - et donc amenées à être priorisées dans les prochains Sprints - sont suffisamment matures et précisément définies pour être développées efficacement par l'équipe. Une table des lois est donc à écrire pour définir de degré de maturité minimal d'une User Story. C'est la **Definition of Ready**.

Elle vient en complément de la **Definition of Done**, qui permet de déterminer si une User Story est terminée ou non. Ces deux outils, simples et dont la définition est propre à chaque équipe, permettent de faciliter fortement la tâche de l'équipe.

Le ScrumMaster aura aussi pour rôle d'éviter autant que possible la modification du contenu du Sprint, et d'alerter dans le cas contraire. Il pourra par exemple utiliser des post-it de couleur différente pour mettre en valeur les tâches rentrées en cours de sprint.

Il est aussi du ressort du ScrumMaster que le **Scrum Board** soit à jour et correctement alimenté. Cela permet de générer un **Burn-down Chart** correspondant à la réalité.

Les cérémonies

Un premier levier possible est de jouer sur la **durée du Sprint**. Il dure généralement entre 1 et 4 semaines. Il est conseillé de commencer avec une durée de 2 semaines, et d'ajuster en fonction des résultats. Mais il est conseillé d'expérimenter, en fonction des contraintes : pression sur le Time To Market, facilité à sortir des User Stories entièrement terminées, durées des cérémonies, etc.

Durant le **Sprint Planning**, le ScrumMaster veillera à ce que tous les membres de l'équipe participent aux discussions et au chiffrage, et pas à ce que la réunion soit phagocytée par ceux parlant le plus fort.

Durant le **Daily Scrum**, il s'attachera à ce que le Scrum Board soit bien à jour, et à ce que les tâches s'exécutent de façon fluide, et lèvera les alertes en cas de forte déviance vis-à-vis de la pente idéale du Burndown Chart. En cas d'embourbement dans la colonne « in progress », par exemple, il pour-

ra utiliser un **quota maximum de tâches** exécutées en parallèle par chacun.

Avant la **démonstration**, il s'assurera que le produit est bien avancé et démontrable. Il s'attachera à noter scrupuleusement les retours faits par les utilisateurs ou le Product Owner.

Durant la **rétrospective**, il mettra une attention particulière à ce que chacun évoque les problèmes rencontrés durant le Sprint et pourra par exemple utiliser la méthode des 5 Pourquoi ou une Root Cause Analysis pour arriver à dégager des actions réellement efficaces. Et surtout il s'alimentera un **backlog de problèmes**, qu'il dépilera au fur et à mesure, permettant d'accroître la vitesse de l'équipe.

De manière générale, les règles de Scrum sont extrêmement simples. Il est fortement conseillé d'y coller autant que possible. Ce n'est qu'une fois que l'équipe sera arrivée aux limites de ces dernières qu'elle pourra éventuellement en détourner certaines, si cela permet d'améliorer l'efficacité de l'équipe.

> Utiliser les bonnes pratiques d'ingénierie d'eXtreme Programming

Nous évoquons un peu plus haut le **Pair Programming** et le **Planning Poker**. Ce sont deux des briques d'eXtreme Programming (XP) qu'il est intéressant d'utiliser. Mais ce ne sont pas les seules. En effet Scrum s'attache à la gestion de projet, mais s'adosse nécessairement à « l'excellence technique », tout en restant vague à ce sujet. Heureusement, XP répond à ce besoin.

Les autres briques les plus intéressantes à utiliser sont :

- **L'Intégration Continue**, voire le Déploiement Continu, de façon à automatiser le plus possible les tâches de packaging, déploiement, documentation, tests de non-régression, etc.
- **Le Test Driven Development** : l'assurance d'un code répondant au besoin, tout en automatisant la non-régression.
- **Le Refactoring** : ne jamais laisser pourrir du code, toujours l'améliorer.
- **KISS** : Keep It Simple & Stupid. Le code le plus simple est souvent le meilleur : plus efficace, plus maintenable, plus pérenne.
- **YAGNI** : You Aren't Going To Need It. Ne tentez pas de deviner des besoins futurs. N'écrivez pas du code pseudo-générique pour la beauté de l'art. Quand le besoin arrivera, s'il arrive, il sera de toute façon différent de ce que vous aviez imaginé, et

le refactoring sera obligatoire de toute façon. En revanche, rien ne vous empêche – bien au contraire – d'écrire du code **SOLID**.

> Gérer le temps qui passe

Ce dernier point n'est pas le plus simple à adresser. Il est facile d'arriver sur un projet, et trouver des centaines de leviers d'amélioration. Mais gérer l'accroissement de la vitesse sur le long terme est souvent plus compliqué.

La première règle à appliquer est celle des **baby steps**. Il est le plus souvent contre-productif de vouloir tout faire d'un coup. Dans Scrum, le rythme doit être soutenable. Une erreur fréquemment faite est de ne voir que la vitesse, et que l'équipe fasse des efforts supplémentaires pour la voir augmenter. Mais ce rythme n'est pas tenable sur le long terme. Donc il est important de dépiler le backlog de problèmes à rythme bas mais régulier. Cela permettra également une mesure plus précise du ROI de chaque amélioration effectuée.

Aussi, le ScrumMaster n'est pas un surhomme. Lui aussi peut finir par s'engluer dans la routine. Il devra donc faire un effort supplémentaire pour prendre le recul nécessaire à la bonne exécution de son rôle sur le long terme. Il est important de lui laisser du temps pour ses tâches de ScrumMaster. Il pourra également faire partie des nombreuses communautés dans les métiers de l'informatique, de façon à échanger avec d'autres et trouver de nouvelles idées.

> Conclusion

Le rôle de ScrumMaster est protéiforme et variable dans le temps. Il existe autant de types de ScrumMasters qu'il existe d'équipes et de projets.

J'espère que cet article vous aura permis de comprendre certains des écueils dans lesquels il est facile de tomber, et quelques leviers permettant de travailler dans de bonnes conditions. Le ScrumMaster doit avoir pour but de se rendre *dispensable*, voire inutile. Mais, avouons-le, le cas est rare. Et le chemin peut être très long avant d'y arriver. Le ScrumMaster a encore une longue vie devant lui.



Nicholas Suter

Consultant Senior chez Cellenza
Cellenza - Software Development Done Right

Blog: <http://www.nicholassuter.com>



Intégrer une équipe Scrum

Comme nous l'avons vu dans les pages précédentes, Scrum est une méthode agile de gestion de projet qui a été conçue lors de projets de développement de logiciels. Le terme Scrum est emprunté au rugby et signifie mêlée. Ce processus s'articule en effet autour d'une équipe soudée, qui cherche à atteindre un but, comme c'est le cas en rugby pour avancer avec le ballon pendant une mêlée, le but étant ici la satisfaction des clients.

Des prototypes du produit peuvent être montrés aux clients, à des intervalles courts, afin de recueillir des commentaires et ainsi pouvoir fournir des logiciels de qualité supérieure et surtout qui correspondent aux besoins de ces derniers. Scrum repose fortement sur une équipe très motivée, en étroite collaboration et auto-organisée. En revanche, Scrum ne fournit pas une liste complète et détaillée de la façon dont les choses doivent être faites. C'est l'équipe Scrum qui a le pouvoir de décider du comment. C'est l'équipe qui saura le mieux résoudre un problème qui se présentera. Cette flexibilité est la clé du succès de Scrum. Si on se place du point de vue d'un membre d'équipe, quels sont les rôles et responsabilités d'une équipe Scrum ? Pour un développeur, quels sont les intérêts et les bénéfices qu'il aura de passer d'une équipe en Cycle en V à une équipe Scrum ?

> Equipe Scrum Rôles et responsabilités

A la fin de chaque sprint, l'équipe Scrum est responsable pour la livraison d'un produit de qualité et dans les temps vis à vis des engagements pris sur le sprint. Afin de garantir cette qualité, j'estime qu'il est indispensable d'y inclure les tests unitaires.

Elle est multi fonctionnelle et polyvalente. Ses membres connaissent leurs forces et travaillent ensemble sur d'éventuels problèmes rencontrés et bloquants pour l'un ou plusieurs d'entre eux. Ils ne sont pas tous des experts dans tous les domaines mais ils forment à eux tous un large éventail de compétences et domaines d'expertises. Une équipe peut comprendre des architectes, des développeurs, des testeurs, des designers...

L'ensemble de l'équipe Scrum est responsable de ses engagements mais aussi de ses actes et décisions. Ils ne font qu'un, ce qui implique à mon sens, de prendre des décisions techniques ensemble. A mon avis, le mieux est d'organiser des réunions tech-

niques cadrées dans le temps (on parle de « timeboxing ») et au niveau du sujet et des éléments à valider.

L'équipe Scrum doit savoir prendre positivement les changements fonctionnels (« Welcome change ! ») ou les re-priorisations des User Stories extérieures au sprint. Le but est que le produit livré corresponde aux besoins fonctionnels du métier.

C'est pourquoi elle s'auto-organise afin de respecter les engagements pris sur le sprint. Il peut s'agir d'instaurer des règles et traditions d'équipe. Pour ma part, je conseillerai également d'y inclure le Pair Programming sur des tâches plus complexes. Et par ailleurs, c'est un excellent moyen pour faire monter en compétences un autre membre de l'équipe. L'équipe fait donc tout ce qui est nécessaire afin de fournir un livrable de la meilleure qualité possible en fin de sprint.

Enfin, l'équipe doit s'efforcer de s'améliorer et d'améliorer la qualité de son livrable à travers chaque sprint, tout en gardant un rythme de travail soutenable.

> Participation aux événements Scrum en tant que membre d'équipe

Poker Planning : il faut travailler en équipe pour se mettre d'accord sur la complexité de chacune des stories à estimer. Partager les points de vue entre les membres qui ont donné la plus haute et la plus basse estimation et ré estimer une nouvelle fois. On prend en compte chaque avis. Comme on l'a vu précédemment, afin de garantir un produit de qualité, il faut inclure dans l'estimation de chacune des stories les tests unitaires.

Sprint Planning : il faut se mettre d'accord avec le Product Owner afin de définir les objectifs du sprint à venir et travailler ensemble pour se mettre d'accord sur les User Stories à embarquer et les prioriser. Les User Stories doivent ensuite être découpées en tâches atomiques d'une journée ou moins, et peuvent être estimées en heures.

Daily Scrum Meeting : l'équipe se rassemble chaque jour pour communiquer et synchroniser son travail. A tour de rôle, chacun informe sur ce qu'il a fait depuis le dernier meeting et sur ce qu'il va faire aujourd'hui. Cette petite réunion est aussi essentielle pour s'assurer qu'il n'y ait aucun problème bloquant qui empêcherait l'équipe d'avancer. Si des obstacles ont été identifiés, ils sont transmis au ScrumMaster pour qu'il puisse mettre en place une action pour chacun d'entre eux. Ce dernier prend également note de l'état d'avancement afin de pouvoir mettre à jour le Burn Down Chart (ou le Burn Up).

Sprint Review : à la fin de chaque sprint, l'équipe montre le travail réalisé durant ce sprint devant le Product Owner, le ScrumMaster, les personnes du métier. Cela se fait sous forme d'une démonstration des nouvelles fonctionnalités. Le Sprint Review permet au Product Owner de valider le travail réalisé au regard des critères d'acceptation que lui et l'équipe ont définis. Il me paraît donc important que l'équipe prépare bien la démonstration en vérifiant les critères d'acceptation de chacune des stories à présenter.

Sprint Retrospective : à chaque fin de sprint, le ScrumMaster va faire un bilan avec l'équipe où on va identifier les points positifs et les points négatifs sur le sprint qui vient de se terminer. Le ScrumMaster va en tirer des axes d'améliorations à prendre en compte pour le prochain sprint. A mon sens, il est vraiment important que chacun des membres de l'équipe exprime réellement son ressenti, le but étant vraiment d'améliorer les points négatifs et de faire en sorte que chacun garde sa motivation intacte.

> Intérêts et bénéfices pour un développeur avec l'approche Scrum

Les membres de l'équipe sont amenés à travailler ensemble et communiquer régulièrement entre eux, que ce soit lors du Daily meeting ou dans la journée, ce qui permet

de partager les connaissances techniques et fonctionnelles. Cela permet également aux membres de l'équipe de rester impliqués et d'éviter que l'équipe ne devienne trop dépendante d'une seule personne. Il faut que plusieurs membres soient en mesure de prendre une tâche quelconque.

Les livraisons régulières motivent l'équipe et l'impliquent dans la réussite du projet. Les démonstrations des fonctionnalités en fin de sprint, lors du Sprint Review, mettent en valeur le travail de l'équipe.

L'équipe travaille dans de bonnes conditions puisque le ScrumMaster est chargé de lever les obstacles rencontrés et fait en sorte qu'elle ne soit pas perturbée par des éléments extérieurs du projet. Elle peut se concentrer sur le code en priorité, et ensuite sur la documentation. La documentation est d'ailleurs revue, dans sa forme. Par exemple, elle peut se matérialiser sous forme de schémas, de scénarios de tests, de commentaires dans le code...

L'équipe travaille dans un rythme soutenable. A la fin de chaque sprint, la capacité de production (la vélocité) de l'équipe est mesurée et en fonction de celle-ci, la planification des stories (fonctionnalités) pour le sprint suivant est ajustée.

> Equipe Scrum Devoirs et contraintes

On a vu que passer sur un projet en mode Scrum, peut apporter au développeur des

points positifs. Cependant, cette méthode engendre également des implications aux membres de l'équipe.

Une équipe polyvalente est cruciale pour que le projet puisse se finir dans de bonnes conditions. Cependant, beaucoup d'équipes Scrum ne sont pas ou ne parviennent pas à devenir polyvalentes. Parmi les raisons les plus fréquentes qui les en empêchent se trouvent les "Rôles prédéfinis" au sein de l'équipe. Les membres d'équipe eux-mêmes contribuent à ce que l'équipe ne tende pas vers la polyvalence.

Cela se produit quand ils décident eux-mêmes de prendre des tâches qui sont liées directement à leurs compétences et pas plus (« Ce n'est pas à moi de le faire ! »). Ces membres ne sont pas prêts à faire ce qui dépasse leurs compétences.

Un membre d'équipe Scrum n'est pas qu'un simple exécutant, il est pro actif. L'équipe étant auto-organisée, chacun a un rôle à jouer que ce soit la participation aux prises de décisions de l'équipe, se mettre volontairement en pair programming pour un transfert de connaissances ou encore essayer d'améliorer ce process au cours des sprints suivants.

L'équipe étant garante de la livraison des stories et des engagements pris en début de sprint, on va mettre en avant le travail de l'équipe. De même, si des stories sont considérées comme étant KO par le Product Owner, c'est toute l'équipe qui en est respon-

sable. C'est lors du Sprint Retrospective que l'équipe pourra se pencher sur les raisons de ces échecs afin de ne plus reproduire les mêmes erreurs.

> Conclusion

Scrum a été créé pour améliorer la satisfaction de l'équipe et sa productivité, la qualité du produit, la réactivité et la transparence face aux demandes des clients. La transparence est d'ailleurs vitale dans le processus Scrum, elle permet à chacun de voir et de comprendre ce qui se passe sur chacun des sprints.

Une équipe Scrum se démarque d'une équipe traditionnelle de développement. Elle est spéciale de par sa composition et ses capacités. A chaque sprint, l'équipe est libre de déterminer la façon dont le travail sera fait. Cela lui procure une certaine autonomie mais cette liberté s'accompagne aussi d'une responsabilité, celle d'atteindre les objectifs du sprint. L'équipe Scrum joue donc un rôle primordial dans le projet. Une équipe soudée avec une bonne entente entre les membres favorisera le travail.



David Tran
Consultant chez Cellenza
Cellenza - Software Development Done Right

Formez et fidélisez vos équipes techniques en leur offrant un

Abonnement à Programmez !

Une formation et une veille technologique, à prix réduit



1 an d'abonnement = **39 € seulement** par abonné (à partir de 10 abonnés)

Pour 3,25 € par mois par développeur,

+Offre de lancement : 1 page de Publicité offerte pour votre entreprise !

À partir de 100 abonnés, offre de lancement

Adressez un mail à diff@programmez.com, en indiquant le nombre d'abonnés souhaité, pour recevoir le devis.

www.programmez.com



Le Product Owner : « Le métier au cœur de la Technique ! »

Scrum a reçu les faveurs de nombre de sociétés, souvent par conviction, parfois par mimétisme. L'agilité, cependant, n'est pas sujet facile, et sa mise en œuvre ne requiert pas moins de discipline et de maîtrise que la méthode traditionnelle.

Avant toute chose, résumons Scrum simplement. Scrum est un framework de gestion de projets, d'inspiration « amélioration continue ». Par ailleurs, et c'est notamment ce qui l'a rendu populaire dans les cercles techniques, Scrum peut être aisément conjugué avec des pratiques d'excellence, propres au développement logiciel, comme eXtreme Programming. Dans cet article, nous allons apporter un éclairage sur le rôle du Product Owner pour Scrum, à travers un retour d'expérience depuis les tranchées.

François est créateur d'entreprise. Il souhaite proposer un service innovant aux entreprises, qui s'appuiera sur une solution logicielle spécifique. Problème : comment construire ce logiciel ? Amis lecteurs, mettez-vous à sa place : êtes-vous prêts à puiser plusieurs dizaines de milliers d'euros de vos économies, pour les confier à un tiers ?

> Qui veut être responsable ?

Savez-vous qu'un projet informatique sur deux échoue ? Que les fonctionnalités réellement utilisées d'un logiciel ne représentent statistiquement que 20% des fonctionnalités développées ? Que le budget annoncé par les fournisseurs consultés est très souvent dépassé ?

Celui qui accepte la responsabilité du pilotage de la production logicielle, et garantit sa qualité et son adéquation aux besoins (non encore identifiés) est, dans Scrum, appelé Product Owner. François, riche de plusieurs années d'expérience dans le secteur informatique, assume ce rôle difficile.

« Au début, la sérénité n'est pas vraiment le sentiment qui prédomine ! », témoigne-t-il. « Les enjeux sont élevés, et les risques nombreux. Nous n'avons aucun droit à l'erreur. ». Maîtrise des risques et création de valeur s'imposent de fait au Product Owner comme deux préoccupations à constamment garder



© andres

quand même fallu un peu de temps avant que nous trouvions un terrain d'entente, avec les développeurs, sur le niveau de détail des critères d'acceptation des stories », sourit François. Mais une communication régulière avec l'équipe, quasi-quotidienne, permet de lever efficacement tous les doutes fonctionnels et d'éviter blocages ou dérives.

Aujourd'hui, François est satisfait de son logiciel, produit dans le respect du budget initial et des contraintes de Time-to-Market.

à l'esprit. Le projet, avant ses itérations officielles, démarre par une phase de cadrage. Ateliers avec Innovation Games (<http://innovationgames.com/>) pour mieux appréhender les besoins, exercices de définition et de sizing des User Stories, construction d'un prototype fonctionnel levant le maximum d'incertitudes techniques. « A l'issue de cette première étape, j'étais nettement plus confiant pour la suite », se souvient François. « Nous avons démontré la faisabilité technique, travaillé sur chaque risque identifié, et commencé à cerner la solution de manière plus précise. ».

> Ce n'est pas toujours simple

Commence alors la seconde phase, la production proprement dite, qui s'étendra sur sept sprints de deux semaines. Le premier sprint – toujours particulier dans Scrum – livre un produit non fonctionnel, que le Product Owner accepte pourtant volontiers : « Pour ce premier sprint, nous avons élaboré toutes les maquettes significatives des écrans du produit. Les boutons sont bien là, mais ne fonctionnent pas encore. A ce stade, l'essentiel est de nous projeter vers la cible, de visualiser la valeur potentielle. ».

Et les sprints s'enchaînent, les fonctionnalités fleurissent. L'écriture des User Stories, néanmoins, n'est pas toujours facile. « Il a

Lorsqu'on lui demande quels facteurs ont été clés pour la réussite du projet, il cite notamment la transparence et le feedback. « J'étais sur site quasiment tous les jours, dans le bureau des développeurs. Je pouvais répondre à leurs questions en temps réel, je mesurais leurs progrès et leurs difficultés tout aussi facilement. Voir le produit grandir sous mes yeux était non seulement satisfaisant, mais aussi nécessaire : j'ai pu ajuster les priorités de mon backlog au fil de l'eau ». Pour conclure, comment François a-t-il vécu cette expérience ? « Scrum est loin d'être facile, ou tout du moins, pas plus facile qu'une quelconque autre méthodologie. Si je devais caractériser le rôle du Product Owner, je dirais que c'est un rôle central mais surtout engageant. Ceci dans la mesure où il faut être réactif, être capable de prendre et parfois d'imposer des décisions face à un parterre de techniciens dont les aspirations ne sont pas nécessairement les mêmes. Néanmoins, une formation « Product Owner », couplée à une écoute partagée et une présence soutenue, ont permis de surmonter tous les obstacles avec succès ».

François Bouloré,
Directeur associé TWIN CORP,
Product Owner (certifié)

Yann Perio,
Consultant technique ZENIKA,
ScrumMaster (certifié)

ScrumMaster : devenir un expert Scrum

Vous avez suivi une formation initiale sur la méthode Scrum ou bien lu des ouvrages, le dossier de ce mois dans Programmez ! et vous connaissez les aspects théoriques ainsi que les points-clés de cette méthode. Vous souhaitez maintenant la déployer et l'utiliser concrètement ? Devenir Scrum-Master ? Voici quelques conseils qui vous aideront à la mise en œuvre !

> Les axes d'expertise

Trois axes permettent de mener à bien l'adoption de Scrum :

- 1 La **méthode** et les pratiques associées,
- 2 Les pratiques d'**ingénierie**,
- 3 Le troisième axe est lié au développement d'un **état d'esprit agile** pour l'**amélioration continue**. Cela résulte d'une transformation allant au-delà de la simple adoption de pratiques. Cette transformation vers l'agilité est souvent synonyme de changement culturel. Elle a pour référence les valeurs du Manifeste Agile et s'applique certainement au-delà de l'équipe elle-même.

L'expertise Scrum se construit sur le terrain. En ce sens, devenir un expert Scrum nécessite d'acquérir des connaissances dans chacune de ces trois directions (méthode, ingénierie, culture de l'amélioration continue) et surtout de diffuser ces connaissances de manière efficace.

Hâtez-vous lentement, et sans perdre courage, Vingt fois sur le métier remettez votre ouvrage.

Nicolas Boileau

L'Art poétique - 1674

> La méthode et les pratiques

Abordons tout d'abord la question de la méthode et des pratiques associées. La première étape passe par la mise en place du cérémonial de cette méthode avec l'équipe. Rappelons que les membres de l'équipe sont :

Le **Product Owner** (ou PO). Son rôle consiste à définir le « Quoi », le contenu d'un produit ou d'un projet et à déterminer les priorités du backlog de produit,

Les **développeurs**, dont le rôle est d'élaborer une solution pour répondre aux besoins fonctionnels émis par le PO, ils définissent le « Comment »,

Le **Scrum-Master** (vous-même ?). Garant de la mise en œuvre de la méthode, il doit protéger l'équipe des perturbations et enlever les obstacles qui bloquent la progression.

Rappelons aussi que l'équipe travaille à partir d'un backlog de stories. Une **story** est le

conteneur (l'expression) d'un besoin fonctionnel élémentaire dont la réalisation apportera une plus-value au produit et le **backlog** est le conteneur des stories, ordonnées ou « priorisées ». L'équipe sera plus efficace si elle est co-localisée. Les équipiers (n'oublions pas la métaphore sportive Scrum signifiant « Mêlée ») sont peu nombreux afin de favoriser la communication. Ne freinez pas cette communication en dispersant l'équipe aux quatre coins d'un open-space. Incitez au rapprochement, idéalement dans un espace dédié. L'équipe y installera son « radiateur d'informations » ou tableau d'équipe : [Fig.1]. L'équipe décide de son rythme de travail. La cadence est donnée par les sprints. Ensemble, fixez une durée pour les sprints, la plus courte possible (entre deux et quatre semaines) de façon à recueillir du feedback très rapidement. Le concept de feedback – ou boucle de retour – est un des concepts majeurs, sous-jacents à l'agilité.

Le cadre temporel ainsi défini, mettez en place les réunions du cérémonial :

Le **Scrum quotidien** : définissez-en l'horaire, pour une durée maximale de 15 minutes. Rassemblez l'équipe devant le radiateur d'informations. Affichez clairement sur ce radiateur l'objectif de cette réunion qui est que chacun réponde à trois questions : qu'ai-je fait la veille ? Sur quoi vais-je travailler aujourd'hui ? Quels sont les obstacles qui m'empêchent de poursuivre mon travail ? En tant que Scrum-Master, vous veillez à l'écoute active et à la concentration des équipiers, au respect de la parole de chacun. Vous écartez du cadre de cette réunion tout ce qui ne concerne pas directement l'avancée de l'équipe et vous incitez au respect de la durée. A la fin de cette courte réunion, mettez à jour les indicateurs que sont les graphiques de burndown (« burndown chart ») et de burnup (« burnup chart »). Les don-

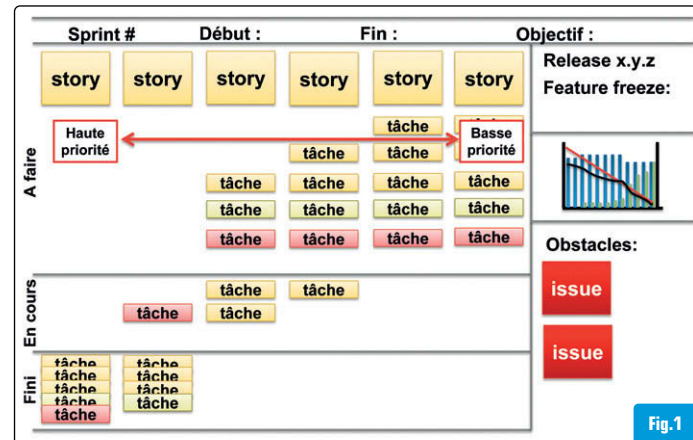


Fig.1

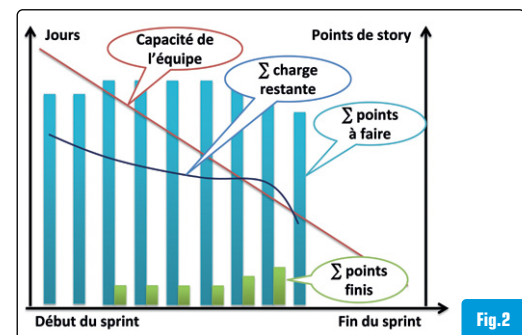


Fig.2

nées à mettre à jour sont : la charge totale restante (en nombre d'heures ou de jours) ainsi que le nombre total de points des stories finies. Ceci permettra à l'équipe de vérifier la tenue de son engagement ou éventuellement de modifier le contenu du sprint. Ci-dessus un exemple de graphique cumulant le burndown (charge) et le burnup (points) : [Fig.2].

La réunion de **planification de sprint** : elle est animée par le Product Owner. Ce dernier présente les stories à venir, une par une. Pour chaque story, l'équipe vérifie que la story est « prête » et en estime la complexité relative. Si la story n'est pas assez décrite, n'essayez pas de la définir en séance, mais consacrez-y plutôt une réunion spécifique avec les seules personnes concernées. Globalement, vous facilitez la réunion de planification et veillez, là-encore, à sa durée. La « timebox » ou « durée contrainte » est un deuxième concept important de Scrum et des méthodes agiles. Apprendre à respec-



ter cette contrainte est un bon entraînement à rester concentré et induit des réunions d'une plus grande efficacité. Ne dépassez pas une heure et demie. Si nécessaire – par exemple si tout le backlog de sprint n'a pas été évalué – demandez au PO de planifier une réunion supplémentaire. Dans le cas où l'équipe rencontre des difficultés récurrentes à décider si une story est « prête » ou non, établissez conjointement une définition de « prêt ». Définissez les critères qui permettront d'apposer le tampon « prêt » sur une story, et affichez cette liste lors des réunions de planification ! Des données telles que la capacité de l'équipe (en tenant compte des congés, arrivées ou départs d'équiers, etc.) ainsi que la vélocité lors des sprints précédents permettent de conforter l'engagement de l'équipe sur le périmètre du sprint concerné par cette planification.

La **revue** incluant la **démonstration** : le Product Owner est l'animateur de cette réunion qui a lieu à la fin du sprint. Il s'agit de passer en revue les stories du backlog du sprint qui s'achève. Le PO accepte ou rejette les stories suivant que les critères d'acceptation sont remplis ou non. Scénariser la revue des stories permet de les présenter dans un contexte propice à la récupération de feedback. En effet, le PO a invité l'équipe de réalisation, bien entendu, mais aussi toutes les parties prenantes : les clients (internes ou externes), le management, etc. Le PO tient compte du feedback fourni pour modifier la description des stories à venir, ajouter ou supprimer des stories. A la fin de la revue, le backlog de produit est mis à jour ainsi que le backlog du sprint suivant. La durée de la revue dépend du contenu du backlog de sprint, comptez une heure et demie à deux heures pour des sprints de deux ou trois semaines. Et, encore une fois, veillez au respect de la « timebox » ! Comme la planification de sprint nécessite une définition de « prêt », lors de la revue, l'équipe utilise la définition de « fini ». Cette définition est une checklist dans laquelle tous les points considérés par l'équipe comme nécessaires à la réalisation sont mentionnés (par exemple : la revue du code a été réalisée). Affichez cette liste sur le radiateur d'information, ce sera un excellent moyen de la mettre en pratique lors du scrum quotidien.

La **rétrospective** : idéalement planifiée juste après la revue, cette réunion des membres de l'équipe permet de se pencher sur le « comment » (la revue s'intéressait au « quoi »). Animée par le ScrumMaster (ou

par un « facilitateur » dans l'équipe), elle permet de collecter des données sur le déroulement du sprint qui se termine. De ces données, événements, points notables positifs ou négatifs, l'équipe dégage des thématiques et définit des actions de façon à résoudre les problèmes rencontrés. Faites réagir l'équipe pour aller au-delà de la partie visible de ces problèmes, sous la surface et jusqu'aux causes racines, les actions prises devront s'attaquer à ces causes racines de façon à résoudre les problèmes de manière durable. Prêtez une attention particulière à la définition de ces actions. Pour que l'amélioration soit sensible, préférez peu d'actions (une ou deux), rendez ces actions visibles (par exemple sur le tableau de l'équipe) et faites en sorte que ces actions puissent être terminées lors du sprint suivant. Les critères « SMART » (Spécifique/Simple, Mesurable, Acceptable, Réaliste, Temporel) doivent pouvoir s'appliquer à ces actions. Des rétrospectives régulièrement organisées participent efficacement au processus d'amélioration continue. Nous avons vu que la revue permettait de recueillir du feedback sur ce qui a été produit, la rétrospective permet de récupérer du feedback sur le vécu de l'équipe ! Comme en musique, un rythme est établi. Les réunions du cérémonial s'inscrivent dans une cadence qui devient un automatisme. La régularité participe au « lissage » de l'activité et l'équipe peut mettre en œuvre sa créativité de manière soutenable. Au fil du temps, les rétrospectives peuvent perdre de leur attrait. De nombreux moyens existent pour conserver une dynamique et l'adhésion de l'équipe. Dans tous les cas, maintenez la rétrospective ! Si l'équipe a atteint un haut degré de maturité, réduisez-en la durée. Variez les approches dans l'animation de ces réunions, diversifiez les perspectives ou les angles d'attaque. De nombreuses ressources existent sur le sujet des rétrospectives mais aussi sur l'animation et la facilitation des réunions ainsi que sur le management visuel.

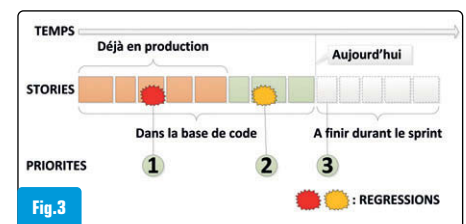
La revue et la rétrospective sont autant de moyens d'inspection et d'adaptation pour une équipe qui évolue dans un environnement complexe. La complexité peut être inhérente au produit ou au projet, mais peut aussi résider dans l'organisation incluant l'équipe. Cette complexité s'accompagne de risques dont la mitigation est plus aisée lorsque l'inspection et l'adaptation sont fréquentes. On en déduit que la méthode Scrum se positionne idéalement dans un

contexte de complexité. Profitez donc des rétrospectives pour *adapter* aux besoins de l'équipe les pratiques précédemment mises en place. Des sujets comme la définition de « prêt » (« definition of ready ») ou de fini (« definition of done ») ou bien l'outillage d'ingénierie sont couramment abordés. L'outillage, en particulier, va s'enrichir au fur et à mesure que les besoins seront exprimés lors de ces réunions.

> L'ingénierie

Un cas classique d'outillage est l'environnement pour intégrer en continu, sujet qu'il est difficile d'éviter aujourd'hui dès que l'on souhaite développer du logiciel. L'objectif d'un tel environnement est de construire les artefacts du produit lors de chaque « commit » et/ou de manière séquencée (les « nightly builds » par exemple). A la base, l'idée est d'éliminer toutes les étapes manuelles de construction du produit et d'exécution des tests (unitaires et d'intégration) de façon à ce que l'équipe puisse se consacrer pleinement à la production de valeur.

La construction et l'exécution des tests en continu et de manière automatique participent également à l'établissement des priorités pour les équipiers. De même que l'« andon » dans la méthode Lean, une erreur rencontrée pendant l'exécution des tests est le signal d'une régression introduite dans le code. Sans tarder, l'équipe doit se consacrer à l'analyse de l'erreur et à sa résolution. L'échelle des priorités dans le travail quotidien est décrite ci-dessous : [Fig.3](#).



La priorité de résolution des erreurs est issue de la priorité dans la réalisation des stories. Les stories déjà déployées en production sont prioritaires par rapport aux stories pas encore déployées (mais déjà codées) qui elles-mêmes sont prioritaires par rapport aux stories que l'équipe n'a pas commencé à développer. Utiliser concrètement Scrum, c'est essentiellement gérer des priorités !

L'intégration continue apporte énormément de qualité aux développements mais le déploiement continu va encore plus loin ! Son principe est d'automatiser toutes les étapes

techniques de la fabrication du logiciel, depuis le commit jusqu'au déploiement en production. Cette notion traverse la « frontière » de l'équipe et introduit des notions propres au support opérationnel. Le mouvement DevOps préconise d'ailleurs une collaboration forte entre Développement et Opérations (ou Exploitation) de façon à converger vers un objectif commun : maximiser la valeur apportée par un logiciel opérationnel, stable et régulièrement enrichi pour répondre aux nouveaux besoins des clients.

J'ai évoqué l'exécution de tests automatiques dans le cadre de l'intégration continue. L'exécution correcte de ces tests va donner un feu vert pour aller plus loin dans un cycle de livraison. Un deuxième feu vert peut être donné par les résultats de l'analyse statique du code produit. Différents outils sont disponibles, commerciaux ou gratuits, qui traitent la majorité des langages. Ces logiciels fournissent des tableaux de bord regroupant des indicateurs sur la qualité de la base de code. Par exemple la couverture du code par les tests automatiques, les dépendances cycliques, le niveau de complexité des méthodes, les violations des conventions de codage, etc. Encouragez l'équipe à mettre en place un tel outil, à la fois pour une vérification à court terme (la prochaine « release » à fabriquer) et pour une analyse des tendances à long terme afin de planifier des remaniements de code (« refactoring »).

Dans le domaine de l'ingénierie logicielle liée à une méthode agile telle que Scrum, les potentialités d'amélioration sont nombreuses ! Je ne voudrais pas passer sous silence ce qui concerne la tenue à la charge et les performances d'un logiciel. Dans le cas où les contraintes de charge et de performance sont des éléments importants dans les spécifications, l'équipe intègre l'utilisation d'un outillage spécifique dans le cycle des sprints et du déploiement. Attention ! L'optimisation des performances ne doit pas être prématurée. Dans un environnement complexe, il est préférable d'exécuter des tests de performance, puis de les analyser pour détecter les goulots d'étranglement plutôt que de présupposer l'emplacement de ces goulots et de vérifier a posteriori que l'hypothèse prise était la bonne.

La méthode est maintenant déployée, l'équipe constituée et motivée, le cérémonial est appliqué, les pratiques d'ingénierie se mettent en place et le produit s'enrichit de nouvelles fonctionnalités à chaque sprint.

Comment acquérir de l'expertise et maintenir une progression constante ?

Le rôle de ScrumMaster est un apprentissage permanent. En ce sens, l'expertise se bâtit sur l'expérience. Donc... expérimentez !

> Les expériences et l'amélioration continue

Tout d'abord expérimentez en fonction des feedbacks de l'équipe, formalisés lors des rétrospectives. Une pratique intéressante est de maintenir un backlog des points d'amélioration. Comme un backlog de produit ou un backlog de sprint, définissez une priorité pour les éléments et revoyez régulièrement cette priorité. Planifiez ces éléments dans les sprints et évaluez les résultats de ces expériences à la fin du sprint. Persévérez ou non, en fonction des résultats.

Dans ce backlog, ajoutez aussi des éléments du feedback informel que vous pouvez recueillir à tous les moments de la vie de l'équipe. Soyez à l'écoute, un événement particulier peut générer une discussion informelle et des propositions d'amélioration surgissent. Détectez les expressions caractéristiques (« patterns ») telles que : « ce serait bien si... », « si on pouvait... », « pourquoi ne ferait-on pas... », « il nous faudrait... pour ... ». Utilisez ces événements comme source de réflexion dans les rétrospectives à venir mais surtout n'imposez pas une solution de votre propre initiative, l'équipe détient le pouvoir de décider ce qu'elle souhaite améliorer dans sa méthode de travail. Ne pas subtiliser ce pouvoir est une des conditions pour que l'équipe acquière de la confiance en elle.

D'autres propositions d'amélioration proviendront de l'extérieur immédiat à l'équipe : des autres équipes qui interagissent avec elle ou du management.

Mais il existe d'autres sources de propositions. De nombreuses entreprises ont aujourd'hui initié ou complété une transition vers l'agilité, la méthode Scrum a été déployée dans de multiples structures. Le milieu de l'agilité est un milieu particulièrement ouvert dans lequel la mise en commun des « bonnes pratiques » est courante. Parcourez les blogs, les forums de discussion, les réseaux sociaux. Il y a peu de questions que vous vous posez (ou que l'équipe se pose) qui n'aient pas déjà été abordées par d'autres. Profitez des conférences telles que l'« Agile Tour » (des sessions à l'automne, dans le monde entier et dans plusieurs villes en France) ou le Scrum Day (à Paris, au prin-

temps). Le plus souvent ces conférences sont organisées par des associations dont les membres peuvent eux-aussi vous transmettre leur expérience.

La plupart des feedbacks que l'on peut collecter à l'extérieur montrent que le point crucial est la communication. Travaillez les compétences (« soft skills ») telles que : la facilitation, la résolution des conflits, l'animation de réunions. Ne confondez pas ScrumMaster et manager. L'expertise acquise en tant que ScrumMaster est au service de l'équipe, et le ScrumMaster doit savoir s'effacer au point que certains préconisent un rôle de ScrumMaster « tournant » parmi les membres de l'équipe. L'auto-organisation ne doit pas être limitée au domaine de la réalisation du logiciel !

Quelques organismes proposent une certification issue de l'Alliance Agile permettant de devenir « Certified ScrumMaster ». Deux journées de formation par un « Certified Scrum Trainer » sont suivies par un examen sous la forme d'un questionnaire à choix multiples. D'autres approches existent, telle celle venant de la Fédération Agile. Cette fédération composée de praticiens reconnus par leurs pairs propose des formations et de l'accompagnement dans un état d'esprit proche du « mentoring ». La certification ou une formation initiale donnent certainement une base minimale de connaissances mais ne remplacent pas une expérience qui se construira sur plusieurs mois de pratique quotidienne, voire sur plusieurs années !

> Aller plus loin...

Vous êtes maintenant expert, ScrumMaster « certifié » ou pas, et vous constatez que Scrum a structuré la mise en place de l'agilité dans l'équipe et a permis de développer une culture d'amélioration continue. La maturité est atteinte dans l'exécution de cette méthode, rien n'interdit de regarder plus loin ! Par exemple, la méthode Kanban peut constituer une évolution « naturelle » pour des équipes auto-organisées qui souhaitent optimiser leur « temps de cycle », ou qui veulent s'adapter à un contexte évoluant vers des livraisons fréquentes mais irrégulières. Vous voilà lancé, ne vous arrêtez pas en chemin !



Pascal Rieux

ScrumMaster chez Sierra Wireless.
Membre de l'association Agile Toulouse. Il est passionné par l'animation d'équipes et le partage des connaissances.
pascalrieux64@gmail.com

OFFREZ-VOUS

UN ABONNEMENT jusqu'à -50%

Code, gestion de projets, développement web, mobilité, Programmez ! est à la fois votre outil pratique, des articles de code par les meilleurs experts et votre veille technologique.

1 Abonnement 1 an au magazine

49 €* (au lieu de 65,45 €, prix au numéro)

2 Abonnement Intégral : 1 an au magazine + Archives Internet et PDF 59 €*

3 Abonnement 2 ans au magazine

78 €* (au lieu de 130,90 €, prix au numéro)

4 Abonnement intégral 2 ans au magazine + Archives Internet et PDF 88 €*

5 Abonnement Etudiant 1 an au magazine 39 €* + Archives Internet et PDF 49 €*



(*) Tarifs France métropolitaine

Toutes les offres en ligne : www.programmez.com

Abonnez-vous à partir de 3,80 € seulement par mois

Oui, je m'abonne

à retourner avec votre règlement à

Programmez, 17 route des Boulangers 78926 Yvelines cedex 9

- ☐ **Abonnement 1 an au magazine** : 49 €* (au lieu de 65,45 €, prix au numéro)
- ☐ **Abonnement Intégral : 1 an au magazine + archives Internet et PDF** : 59 €* (au lieu de 65,45 €, prix au numéro)
- ☐ **Abonnement 2 ans au magazine** : 78 €* (au lieu de 130,90 €, prix au numéro)
- ☐ **Abonnement intégral 2 ans au magazine + archives Internet et PDF** : 88 €*
- ☐ **Abonnement Etudiant : 1 an au magazine** : 39 €* (joindre copie carte étudiant)
- ☐ **Abonnement Etudiant : 1 an au magazine + archives Internet** : 49 €* (joindre copie carte étudiant)

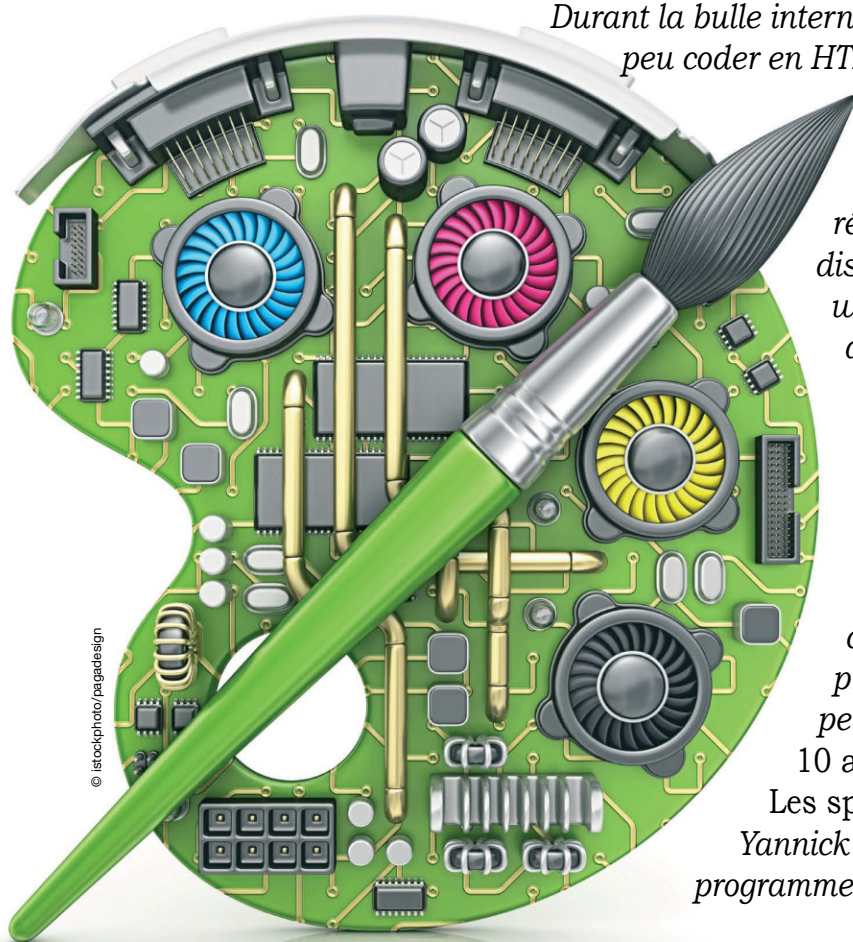
(*) Tarifs France métropolitaine

☐ M. ☐ Mme ☐ Mlle Entreprise : _____ Fonction : _____
Prénom : _____ Nom : _____
Adresse : _____
Code postal : _____ Ville : _____
Tél : _____
E-mail : _____ @ _____

(Attention, e-mail indispensable pour les archives sur internet)

☐ Je joins mon règlement par chèque à l'ordre de Programmez ! ☐ Je souhaite régler à réception de facture

La fin du webmaster Vive les métiers du web !



Durant la bulle internet de 1999-2000, n'importe qui sachant un peu coder en HTML, mettre en place un site web, se disait webmaster. Aujourd'hui, le métier a considérablement évolué. En 2013, le profil même de webmaster n'existe plus réellement. Depuis 3 ans, le profil webmaster disparaît. La raison en est simple, le webmaster était celui qui s'occupait du développement web, du déploiement, de l'administration des serveurs, de la maintenance. Bref, un poste multi-compétences tenu par une seule personne. Mais aujourd'hui, avec une complexité toujours croissante des technologies web, la multiplication des terminaux et des canaux de diffusion, les profils spécialisés priment sur la centralisation autour d'une personne. « Le métier, si nous remontons à 10 ans, on y mettait tout et n'importe quoi. Les spécificités ont gagné le métier », précise Yannick Lejeune (Directeur de la stratégie et des programmes de SUP'Internet).

> Du webmastering aux métiers du web

Le webmastering est aujourd'hui assuré par plusieurs métiers du web. Ainsi, nous pouvons trouver jusqu'à 7 familles de métiers, à savoir :

- développement
- gestion de contenu, production de contenu
- interface & design
- infrastructure et réseau
- formation, assistance, support
- conception et gestion de projet
- communication et marketing

Sur cela peuvent se greffer deux autres compétences qui ne seront pas forcément des profils indépendants : les testeurs web et les responsables de maintenance (infrastructure, contenu, site web, etc.).

Nous pouvons y voir trois grands profils : profils techniques, profils créatif et profils business / marketing. « Le web est un univers où les frontières sont [parfois] floues. Prenez l'exemple du référencement. Il est à la limite entre le marketing et la technique », recadre Yannick Lejeune.

Une des nomenclatures des métiers du web est la suivante [1] :

Famille de métier	Profil
développement / technique	développeur web développeur spécialisé dans un langage développeur orienté mobile intégrateur web lead développeur
création	web designer UX designer ergonome directeur de création ou directeur artistique
marketing	chef de projet community manager rédacteur / content manager responsable social responsable référencement / fidélisation...
direction	directeur technique directeur business directeur marketing / emarketing

(1) Nomenclature issue d'Altaïde

> Des profils transverses et verticaux

Tous les profils et métiers indiqués ne sont pas perméables. Et une personne peut endosser plusieurs rôles. Le web est à la fois transverse car il inclut le business, le marketing, les équipes mais aussi vertical car cela oblige à utiliser une multitude de techniques, d'outils. Ainsi le chef de projet web devra savoir communiquer avec un designer, le marketing, le développeur, comprendre la stratégie de l'entreprise et ce qu'elle fait, pour ensuite aligner la stratégie web à l'entreprise. « *Il ne faut pas être spécialiste d'un (seul) côté. Il faut comprendre la logique Business. Le projet web devient un hub* », précise Yannick Lejeune.

« *Il y a une complexité du web. Avant, on était davantage intégrateur, maintenant, il faut faire du JavaScript, intégrer des frameworks, faire du HTML 5. Il faut pouvoir avoir une formation de développeur mais aussi être un peu architecte. Et savoir aussi prendre en compte les demandes de web mobile* », poursuit Yannick Lejeune.

Beaucoup de projets web sont en réalité pilotés par le marketing, la communication. La technique perd son leadership même si elle continue à jouer un rôle important pour la conception, le déploiement, la maintenance.

Mais aujourd'hui, de nombreuses problématiques touchent l'entreprise dans son ensemble.

Le référencement concernera à la fois le marketing, la communication, la gestion de projet et les développeurs qui devront modifier, adapter les procédures de référencement.

> Le webmaster existe toujours dans les petites entreprises et chez les indépendants

Malgré tout, dans les entreprises ayant peu ou pas de service informatique, les indépendants, le webmaster existe toujours sous une forme ou une autre. On parlera davantage de responsable internet. Il aura des compétences élargies et s'occupera de nombreux sujets mais plus forcément de technique, celle-ci pouvant être externalisée. Ce responsable se concentrera davantage sur l'alignement business du web et les attentes de l'entreprise.

Si nous prenons le dernier rapport de l'association des designers interactifs « sondage sur l'emploi et les salaires du design interactif en France, édition 2013 », le webmaster comme expertise principale atteint un tout petit 3 % !

François Tonic

Des attentes technologiques fortes et une pression du « toujours plus vite »

La pression technologique :

- notion de social et de communauté
- le web mobile (smartphone, tablette)
- responsive interface / design
- HTML 5 / CSS

Mais attention, la technologie pour la technologie n'apporte rien au projet web.

La pression interne / marché :

- des cycles de projet court
- ROI court et rapide
- Respect des réglementations et cadre légal

SALAIRES : UNE ÉNORME DISPARITÉ

L'association des designers interactifs dans son dernier sondage sur les salaires fournit quelques chiffres intéressants :

- salaire moyen brut (tous profils) : 33 670 €

Bien entendu, ce salaire varie selon que l'on est à Paris ou en province, dans une agence, en SSII, etc. Il faut aussi noter une disparité homme / femme. A l'international, ce salaire moyen monte à 52 000 €.

Métier	Salaire moyen (€, brut) junior*	Salaire moyen (€, brut) senior
Architecte de l'information/ UX designer	34 000	44 000
Chef de projet	32 900	37 000
Développeur web**	26 200	35 600
Designer d'interaction	34 000	38 800
Webdesigner	24 800	28 600
Directeur artistique	33 150	38 500

* salaire arrondi. Chiffre indicatif.

** le développeur flash spécifiquement sera revalorisé de 10/15 %

Blue Search a fourni une étude très détaillée sur les salaires (relayée par quozzy.fr) des métiers du web pour 2012, les principaux chiffres à retenir sont :

Métier	Fourchette salariale (€, brut), junior / senior*
Développeur (web)	35-40 k€ / 40 à 65 k€
Directeur artistique	30-45 k€ / 45-60 k€
Intégrateur web	25-35 k€ / 35-50 k€
Webdesigner	25-35 k€ / 35-50 k€
Community manager	30-35 k€ / 40-50 k€

* La fourchette inclut les seniors ayant une expérience de - et + 6 ans

Altaïde, spécialisée sur le web, a dévoilé en février 2013, une estimation des salaires, en voici le résumé :

Métier	Fourchette salariale (€, brut), junior / senior
Développeur (web, front end)	27-32 k€ / 35- 40 k€
Directeur artistique	24-30 k€ / 40-48 k€
Webdesigner	env. 24 k€ / 30-36 k€
Community manager	24-30 k€ / 34-38 k€

Les métiers techniques [développeur web / intégrateur] seront sur des salaires de développeurs classiques, voire, parfois en-deçà. Mais là encore, la disparité est grande comme le montrent les différentes sources citées.

Des profils très spécifiques comme le développeur Flash seront en général mieux valorisés mais attention, ces compétences risquent d'être obsolètes dans un proche avenir. Par contre, un développeur mobile aura tendance à être lui aussi revalorisé, mais sur le long terme. Autre métier peu valorisé actuellement, l'intégrateur web.

Les chefs de projet, les responsables web marketing / référencement seront en général dans des tranches salariales supérieures, mais là, encore, attention aux disparités entre entreprise et Paris / Province.

Formations et compétences

Après avoir évoqué les différents métiers, les salaires, abordons un aspect important : les compétences et la formation. Si l'autodidacte et l'auto-formation sont toujours possibles, mieux vaut avoir une certification reconnue ou un diplôme. Le webmaster 1.0 est définitivement mort !



© Agence Prisme - Pierre Jayet.

Le sondage sur l'emploi 2013 de l'association des designers interactifs propose un profil intéressant de celles et ceux qui travaillent dans le web et plus spécialement sur le design interactif : plutôt jeunes 21-30 ans (+50 %), 52 % ont au moins une formation Bac + 4, il y a moins d'indépendants (27 % contre 41 % en 2011). Ils sont plutôt en poste dans des agences (52 %), Paris et sa région domine toujours les emplois et les femmes représentent un tiers des postes. Sur la formation, près du tiers ont été formés dans des écoles d'informatique, de nouvelles technologies. Les écoles de design représentent 18 % des formations, 20 % pour les universités. L'expérience est très variable : 23 % disent avoir 2 ans et moins d'expérience (ce qui est peu), mais 29 % ont au moins 3 ans et 26 % au minimum 5 ans.

Notre enquête a montré un effet générationnel qu'il ne faut pas négliger pour les métiers du web. La génération Y (que l'on soit d'accord ou non avec ce concept) sera vue comme plus proche des usages actuels, plus proche des nouvelles technologies et des évolutions constantes. Donc après 35-40 ans, nous serions perdus pour ces métiers ? C'est malheureusement une réalité qu'il ne faut pas négliger surtout si vous êtes en reconversion.

Formations & cursus

Plusieurs écoles sur les métiers du web, le design existent en France : Gobelins, l'Ecole Multimédia, Paris 8, Ingemedia, IESA, Hetic, Efficom, Strate Collège, Sup'Internet (groupe Ionis), etc. Le journal économique « Les Echos » avait listé les nouvelles écoles sur les métiers du web : Télécom Ecole de

Management, Ecole d'Internet et de Télévision, Ecole Européenne des Métiers de l'Internet, Ecole Supérieure de Commerce et d'Economie Numérique, Supdeweb, Web School Factory. Ces formations sont payantes (5 000 à 8 000 € / année) et se déroulent sur un cycle de 3 à 5 ans.

Récemment, l'Ecole Européenne des Métiers de l'Internet (EEMI) a annoncé un partenariat avec l'Ecole de Management de Grenoble pour proposer une formation Bac + 6 autour du management numérique et former aux métiers du web. « Nous avons deux cursus, un sur le management du système d'information, en alternance, avec une dimension internet et un master spécialisé en anglais, orienté sur la stratégie internet. Un

3e cursus sera proposé à partir de septembre, avec l'EEMI sur les métiers du web », précise Renaud Cornu-Emieux (le Directeur de l'EMSI Grenoble - Ecole de Management des Systèmes d'Information).



© Agence Prisme - Pierre Jayet.

Aujourd'hui, pour pouvoir créer du web, il faut comprendre la stratégie internet dans son ensemble, celle de l'entreprise et coordonner tout cela. « Le web est aujourd'hui multi-canal. Cela bouge très vite et il y a une difficulté à bien appréhender les choses », poursuit Renaud Cornu-Emieux. Pour ce nouveau master, l'objectif n'est pas de faire de fausses promesses aux candidats. Le « recrutement » sera rigoureux sur les profils.

Même si on ne demande pas à un développeur web, un designer, un chef de projet web de comprendre l'ensemble de la dimension web de l'entreprise, la tendance est clairement à la double compétence : technique

bien entendu mais aussi marketing / social / communication. Yannick Lejeune ne dit pas autre chose. D'ailleurs à Sup'Internet, en soutenance, l'étudiant qui sera plus marketing doit pouvoir dire pourquoi telle technologie a été choisie, mais un développeur web doit aussi être capable de parler de la vision globale web du projet et de l'entreprise. Objectif : avoir une vue transverse de l'internet et pas seulement se focaliser sur son code, un site web.

« Les étudiants n'ont pas conscience de cette complexité quand ils arrivent à l'école. » insiste Yannick Lejeune. D'ailleurs, les étudiants voulant intégrer Sup'Internet sont testés sur le Web dans sa globalité. Ainsi, il faut dépasser, par exemple, la notion de social et de réseau social qui ne doit pas être réduite à Facebook ou Twitter. L'école fournit un cursus de 3 ans autour de 3 axes : développement / technologie web, création et design (avec le contenu), eCommerce / eMarketing. L'école propose des cours transversaux pour faire prendre conscience aux étudiants de la porosité des métiers du web et qu'il ne faut plus être cantonné à un unique profil.

Nos conseils :

- soyez curieux
- n'hésitez pas à acquérir une double ou triple compétence
- faites de la veille technologique constante et formez-vous aux nouvelles techniques et technologies, allez aux conférences
- soyez ouvert et intéressez-vous aux projets web d'entreprise
- n'hésitez pas à aller vers la mobilité, le web mobile et le multi-canal pour étendre vos possibilités de carrière.

F.T.

» IONIS Education Group veut continuer à croître et dévoile ses ambitions. Il proposera un premier cycle gratuit !

IONIS regroupe différentes écoles, dont EPITA, SUP/Internet, Epitech, ISEG Business School. Quatre principaux axes motivent ce volontarisme : le développement international, l'implication sociale, la création d'un pôle numérique et créatif et enfin le développement de la formation continue. Un des grands changements concernera l'école ETNA du groupe qui deviendra dès septembre 2013, une école en 5 ans, avec la mise en place d'un 1er cycle de 2 ans, la PrepEtna, qui accueillera 250 étudiants par an. Surtout, ce cycle sera gratuit ! Le recrutement des 250 étudiants se fera selon les profils, la motivation du candidat et différents critères sociaux. L'objectif est d'aider les jeunes les plus motivés. A partir de la 3e année, l'étudiant est en alternance dans une entreprise. Il doit contribuer au financement du programme du 1er cycle gratuit à hauteur de 250 € / mois durant 3 ans.

D'autre part, le groupe annonce la création courant 2013 d'un fonds de dotation, avec le soutien d'entreprises, pour former les étudiants et mieux intégrer les diplômés dans le monde du travail. Ce fonds doit permettre d'intégrer dans le cursus des élèves qui n'auraient pas les ressources nécessaires pour le faire. Pour en savoir plus : <http://www.etna-alternance.net/prep-etna.aspx>

» Proservia, filiale de ManpowerGroup Solution, a lancé pour 2013 une campagne de 500 recrutements !

Plus de 300 collaborateurs avaient été recrutés en 2012. La SSII cherche majoritairement des Techniciens Helpdesk ; des Chefs de Projets ; des Managers Opérationnels ; des Techniciens, Administrateurs, Ingénieurs Systèmes & Réseaux ; des Experts en infogérance ; des Analystes d'Exploitation et des Architectes. Pour en savoir plus : www.proserviarecruite.fr

» Adyax, agence spécialisée sur l'Open Source et Drupal, recrute différents profils :

chef de projet, développeur Objective-C, profils marketing, etc. Actuellement, l'agence a une centaine de collaborateurs. Pour en savoir plus sur les recrutements : <http://www.adyax.com/fr/jobs>

» Oxalide est spécialisé dans l'infogérance et l'hébergement web open source.

Il procède à une vague de recrutements. Ces postes, basés à Paris, sont à pourvoir immédiatement et en CDI. Oxalide recherche aussi un responsable technique d'équipe, un account manager et de nombreux ingénieurs techniques web. Les annonces sont disponibles sur : <http://www.oxalide.com/emploi/nos-offres/> Les candidatures, CV et lettres de motivation sont à adresser à : job@oxalide.com

» ITS Group recrute plus de 200 collaborateurs.

Les profils sont axés sur les domaines Système (Unix, Linux, Windows), Exploitation, Réseaux et Production sur des postes d'Architecte, Ingénieur, Analyste, Administrateur, Technicien, ainsi que les directeurs et chefs de projets. Le groupe accueille aussi de nombreux stagiaires. Pour en savoir plus : <http://www.itsgroup.fr/fr/nous-rejoindre>

» Econocom recrute 300 collaborateurs.

Pour ce faire, un tour de France (jobmeeting) est organisé jusqu'au 9 avril. « Ces événements nous permettent de rencontrer plusieurs centaines de candidats talentueux en quelques semaines et de leur proposer de belles opportunités de projets et d'évolution de carrière. Cela nous permet aussi de communiquer sur notre groupe, notre passion pour l'innovation, ainsi que nos valeurs d'entrepreneuriat et de partage. » précise Philippe Salaün. Lien : www.econocom.com

Le cabinet Hays publie son baromètre 2013

« En études et développement, les profils les plus recherchés sont ceux de Développeurs et Lead développeurs ainsi que les profils d'Architectes et Concepteurs. Les technologies les plus demandées sont celles en Java JEE, PHP Back End, Asp.net, C#, C et C++ ainsi que Objective C », indique le baromètre Hays. La grille salariale indicative pour les métiers du développement est la suivante :

MÉTIER expérience	-2 ans	2-5 ans	5-10 ans	+ 10 ans
Analyste-programmeur	25-30	30-35	35-40	40-45
Ingénieur développeur	40	45-50	55-65	65-75
Ingénieur qualité / méthode	30-33	33-38	38-42	42-45
Ingénieur intégration	32-38	38-43	45-55	-
Responsable R&D	-	-	50-55	55-65
Chef de projet études et développement	38	38-45	45-55	55-60
Architecte	-	45	55-65	65-75
Directeur technique	-	-	65-75	80-100

En milliers €

Source :

http://www.hays.fr/prd_consump/groups/hays_common/@fr/@content/documents/digitalasset/hays_692085.pdf

» eNovance, important acteur Open Stack et Cloud en France, ouvre au recrutement une quinzaine de postes :

développeur OpenStack ou Python, expert Cloud Computing, chef de projet agile, ingénieur système, etc. Les postes sont basés à Paris, certains se partagent entre Paris et Montréal. Pour en savoir plus : <http://www.enovance.com/fr/a-propos/recrutement>



ABONNEMENT
PDF

30 € par an soit **2,73 € le numéro**
www.programmez.com

Abonnement INTÉGRAL

Pour un supplément de 10 € an
accès illimité aux archives

Cette option est réservée aux abonnés pour 1 an au magazine, quel que soit le type d'abonnement (Standard, Numérique, Etudiant). Le prix de leur abonnement normal est majoré de 10 € (prix identique pour toutes zones géographiques). Pendant la durée de leur abonnement, ils ont ainsi accès, en supplément, à tous les anciens numéros et articles/ dossiers parus.

Utilisons Emacs

Logiciel fondateur du projet GNU, Emacs est l'éditeur mythique du monde Unix. Une excellente compatibilité avec OSX (et Windows) et le retour sur le devant de la scène de la programmation fonctionnelle (Clojure, Erlang, F#) ont rajeuni une communauté désormais visible et active sur des sites tels que GitHub ou Stackoverflow.

Après une courte présentation consacrée à l'installation et à la configuration d'Emacs, cet article introduit le langage Lisp et la manière de concevoir un module qui se chargera d'indenter et colorer un fichier dont nous aurons défini la syntaxe.

INSTALLATION ET CONFIGURATION

Alors qu'Emacs est généralement pré installé sous Linux (ou disponible en un clic avec le gestionnaire de packages), les utilisateurs d'OSX ou de Windows devront procéder à une installation manuelle après avoir téléchargé une archive sur emacsmacosx.com ou <http://ftp.gnu.org/gnu/emacs/windows>. [Fig.1]

> Raccourcis

Une fois lancé, Emacs se pilote via une multitude de raccourcis clavier qu'il conviendra le plus rapidement possible de préférer aux menus de l'application. Citons quelques raccourcis vitaux :

C-x C-c	quitter Emacs
C-x C-f	ouvrir un fichier
C-x C-s	enregistrer les modifications
C-g	annuler une commande / un raccourci

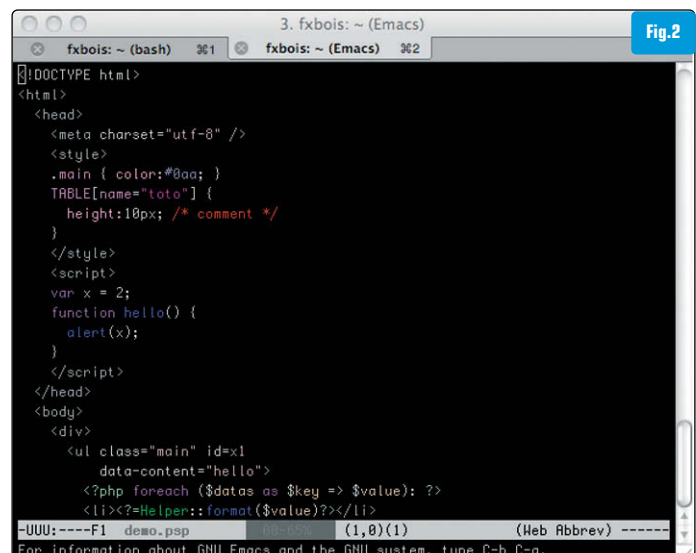
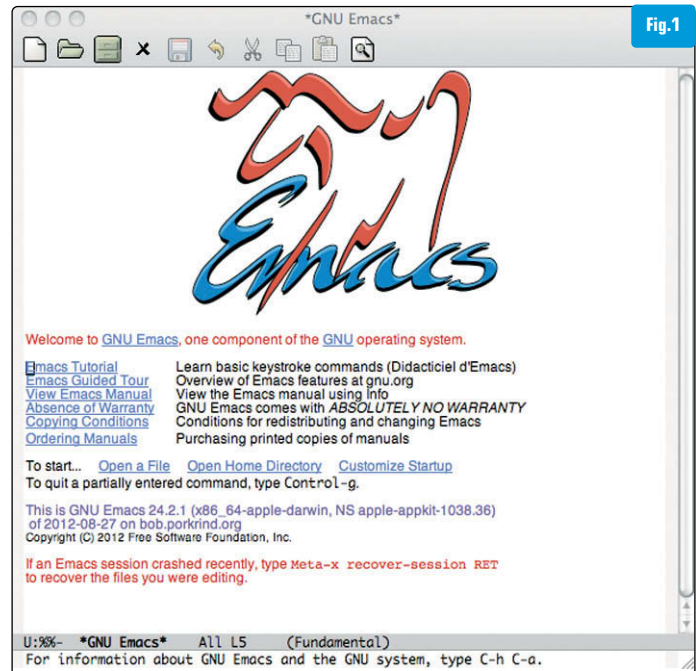
Pour rappel, C correspond à la touche Contrôle et M (meta), à la touche Echap (Esc). En ce qui concerne le copier/coller, Emacs utilise des raccourcis spécifiques.

C-Espace (+ déplacement)	sélection d'un texte
M-w	copier
C-w	couper
C-k	supprimer le texte depuis le curseur jusqu'à la fin de la ligne (après avoir placé le texte supprimé dans le «presse-papier»)
C-y	coller

Ces raccourcis sont d'autant plus utiles à connaître qu'ils peuvent également être utilisés lorsqu' Emacs est exécuté en mode texte dans un terminal. Vous pouvez vous en rendre compte sous OSX en tapant Emacs dans un terminal [Fig.2].

> Configuration

Le fichier de configuration d'Emacs se trouve à la racine du répertoire personnel (~/.); il est nommé `.emacs`. Il permet de modifier différents aspects du logiciel : personnalisation d'éléments visuels (barre d'icônes, barres de défilement, police, couleurs), définition de nouveaux raccourcis clavier, chargement de nouvelles extensions, adaptation de la valeur par défaut de certains paramètres (ex. tabulation, vitesse de défilement). Commençons par quelques ajustements cosmétiques : suppression de l'afreux barre d'icônes, désactivation de l'affichage de l'écran d'accueil et des beeps intempestives sous OSX.



```
(tool-bar-mode -1)
(setq initial-scratch-message nil)
(setq inhibit-startup-screen t)
(setq ring-bell-function 'ignore)
```

Petite remarque au passage, les utilisateurs de Windows rencontreront probablement un souci pour créer un fichier commençant par un point. Il est donc préférable de créer le fichier `~/ .emacs` directe-

ment depuis Emacs [Fig.3]. Notez que de nombreux utilisateurs d'Emacs partagent leur fichier de configuration sur GitHub. La lecture de `.emacs` est souvent le premier pas vers une utilisation plus poussée d'Emacs. Les lignes que nous venons de placer dans le fichier `.emacs` correspondent à des instructions écrites en Lisp, langage que nous allons désormais rapidement présenter.

LISP

Les utilisateurs les plus acharnés utilisent Emacs comme un véritable système d'exploitation. Emacs peut en effet être utilisé en tant que terminal, navigateur web, client IRC, lecteur de mail/news et même console de jeux. Cette capacité à être personnalisé et programmé à l'infini est sans nul doute la plus grande force d'Emacs [Fig.4 et 5]. Cette extensibilité unique est issue de la nature même du logiciel : Emacs peut être considéré comme un interpréteur Lisp dont une des utilisations possibles est l'édition de texte.

Bien que très ancien (il fut inventé par John MacCarthy l'un des pères de l'intelligence artificielle en 1958), le Lisp est profondément respecté par les créateurs de langages. Il est le plus grand des langages selon l'immense Alan Kay (Smalltalk) et une source d'inspiration directe pour Brendan Eich (Monsieur JavaScript).

Ecrire du code en Lisp consiste à manipuler des expressions symboliques. Une s-expression correspond à une liste d'éléments entourés de parenthèses. Elle permet de représenter aussi bien des instructions que des données. Dans le cadre d'une instruction, le premier élément représente une fonction et les éléments suivants, les paramètres de cette fonction. L'expression `(+ 1 2)` correspond à l'addition des nombres 1 et 2. La nature récursive du langage devient évidente dès lors que l'on découvre que chaque élément de la liste peut lui-même correspondre à s-expression : `(+ 1 (- 3 2))`.

Il est possible d'empêcher l'interprétation d'une s-expression. Nous plaçons pour cela une apostrophe devant la parenthèse ouvrante : `'(1 2 3)`. Inversement, la fonction `eval` permet de forcer l'évaluation d'une s-expression. Emacs dispose tout naturellement d'un shell Lisp (`M-x ielm`) qui nous permet de manipuler des expressions symboliques [Fig.6]. L'affectation d'une variable est réalisée avec la fonction `setq` que nous avons rencontrée plus haut dans la partie configuration. `(setq x 1)` affecte la valeur 1 à la variable `x`. La structure de contrôle if s'écrit de la manière suivante : `(if (condition) (alors) (sinon))`.

```
(if (= (% x 2) 0)
    (message «le nombre %S est pair» x)
    (message «le nombre %S est impair» x))
```

Une fonction est créée de la manière suivante (`defun <nom fonction> (<arguments>) (instruction1) (instruction2) ...`).

```
(defun carre (n)
  (* n n))
```

Ces différents éléments de syntaxe connus, l'écriture d'une fonction factorielle devient triviale.

```
(defun factorielle (n)
  (if (= n 0)
      1
      (* n (factorielle (- n 1)))))
```

Utilisons les fonctions pour écrire notre premier « mode ».

LES MODES

Plutôt que de parler d'extensions ou de modules, Emacs utilise le terme *mode*. Deux catégories de modes coexistent : les modes majeurs et mineurs. Les fonctionnalités principales prises en charge par un mode majeur (*major mode*) sont la coloration syntaxique et l'indentation. Un seul mode majeur peut être en cours d'exécution lors de l'édition d'un document (sous Emacs, un document en cours d'édition est appelé un *buffer*). Plusieurs modes mineurs (*minor mode*) peuvent en revanche être en cours d'exécution lors de l'édition d'un buffer ; citons quelques fonctionnalités proposées par des modes mineurs : affichage du numéro de ligne, affichage des espaces de fin de ligne. L'activation d'un mode est réalisée à l'aide d'un raccourci de la forme : `M-x <nom du mode>`.

`M-x linum-mode` active l'affichage des numéros de lignes ; ré exécuter ce raccourci désactive ce mode mineur.

> Création d'un mode majeur

Notre objectif est d'être en mesure de colorer et d'indenter des fichiers de paramètres dont l'extension serait `.prm` et dont le format serait le suivant :

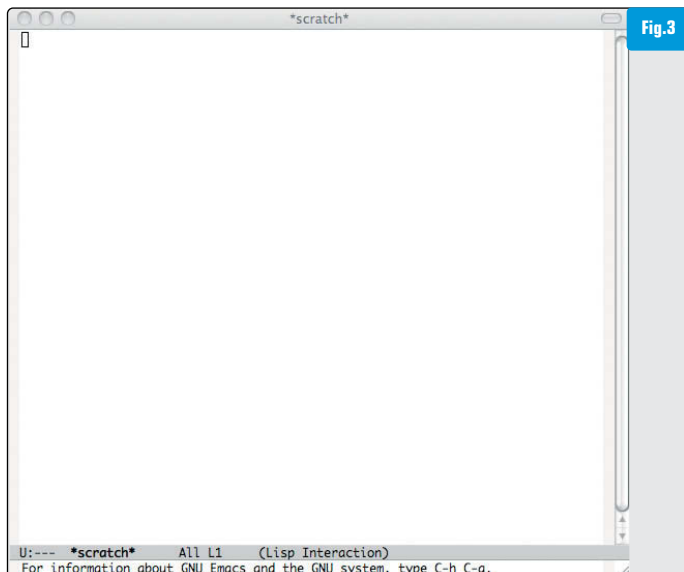


Fig.3

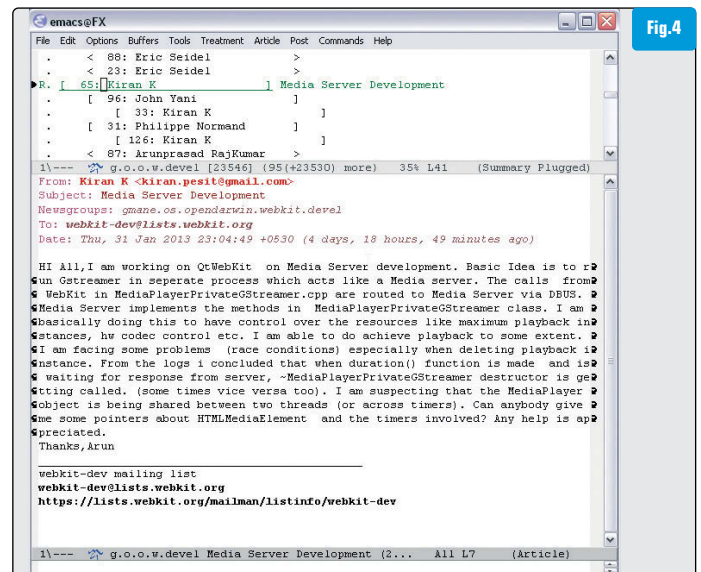


Fig.4


```
==rubrique1
  clef1:valeur1
  clef2:valeur2

==rubrique2
...
```

Le point d'entrée d'un mode majeur est l'utilisation de la fonction `define-derived-mode` qui permet de définir un mode en le faisant hériter d'un autre mode plus général. Nous allons faire hériter `param-mode` du mode le plus fondamental, le bien nommé `fundamental-mode`.

```
(define-derived-mode param-mode fundamental-mode «Param»
  «Mode majeur pour éditer des fichiers de paramètres.»
)
```

Cette fonction nous permet de configurer la coloration syntaxique et de préciser le nom de la fonction en charge de l'indentation.

> Coloration

La manière la plus simple de colorer un buffer sous Emacs consiste à utiliser le mécanisme de `font-locking`. Il s'agit d'un mode mineur qui a le double avantage d'être facilement paramétrable et de prendre à sa charge tous les aspects complexes de la colorisation syntaxique (invalidation des zones modifiées, prise en compte des commentaires/chaînes de caractères).

Paramétrer le `font-locking` consiste à indiquer le nom d'une constante qui contiendra la liste de toutes les règles nécessaires à la coloration du buffer.

Nous souhaitons définir deux règles :

- les lignes qui commencent par deux caractères '=' correspondent à des rubriques
- les lignes de type clef / valeur sont construites de la manière suivante : des caractères blancs (espaces ou tabulations) suivis de caractères alphanumériques (les clefs) suivis du caractère : lui-même suivi de nouveaux caractères quelconques (les valeurs).

Chaque règle est représentée par une liste dont le premier élément est une expression régulière qui va permettre d'associer un style (une *face*) au motif qui a correspondu.

Ecrivons la première règle :

```
'(«^=.*$» 0 font-lock-type-face)
```

Le premier élément de la liste correspond à une expression régulière que nous avons construite ainsi :

- ^ : caractère de début de ligne
- == : deux caractères '='
- .* : une série de caractères quelconques
- \$: une fin de ligne

Le deuxième élément de la liste, le nombre 0, signifie que nous souhaitons appliquer le style (le troisième élément, `font-lock-type-face`) à l'intégralité de la ligne qui correspond à cette expression régulière. Notre seconde règle est écrite de la manière suivante :

```
'(«[[[:blank:]]*\\([[:alnum:]]+\\):\\(\\.+\\)\\$» (1 font-lock-variable-name-face) (2 font-lock-string-face))
```

Nous utilisons dans ce cas deux groupes de capture (ils sont entourés par \\(et \\)) pour associer à chacun d'eux un style spécifique : `font-lock-variable-name-face` pour la clef et `font-lock-string-face` pour la valeur.

Ces deux règles sont regroupées au sein de la constante `config-font-lock-keywords` que nous définissons ainsi :

```
(defconst param-font-lock-keywords
  (list
    '(«^=.*$» 0 font-lock-type-face)
    '(«[[[:blank:]]*\\([[:alnum:]]+\\):\\(\\.+\\)\\$»
      (1 font-lock-variable-name-face)
      (2 font-lock-string-face))
  ))
```

> Indentation

Indenter une ligne de code consiste à décider de combien d'espaces (ou tabulations) cette ligne doit être décalée sur la droite.

Dès que nous appuyons sur la touche TAB, Emacs indente la ligne en faisant appel à la fonction associée à la variable `indent-line-function`. Nous allons construire une telle fonction qui devra mettre en œuvre la règle suivante :

- si les deux premiers caractères non blancs de la ligne correspondent à «==» l'indentation est égale à 0 ;
- sinon l'indentation est égale à 2.

Nous utilisons la fonction `thing-at-point` pour obtenir, sous la forme d'une chaîne de caractères, la ligne sous laquelle se trouve le curseur. Un test de correspondance avec une expression régulière nous permet de détecter si la ligne correspond à une rubrique.

```
(defun config-indent-line ()
  (if (string-match-p «^[[[:blank:]]*==» (thing-at-point 'line))
      (indent-line-to 0)
      (indent-line-to 2)))
```

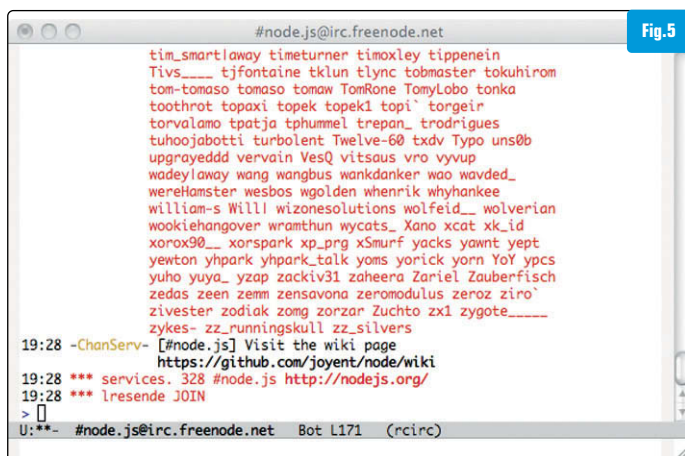


Fig.5

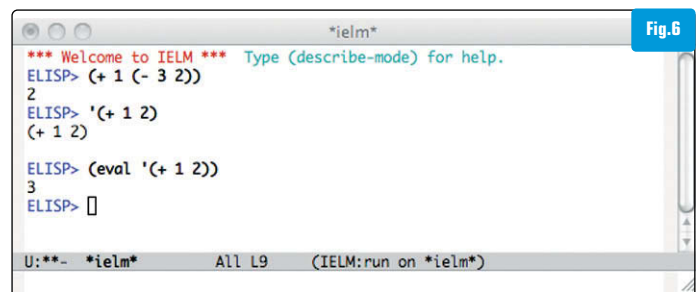


Fig.6

> Association automatique

Regroupons les différentes parties du mode dans un fichier portant le nom `param-mode.el`.

```
(define-derived-mode config-mode fundamental-mode «Config»
  «Mode majeur pour éditer des fichiers de configuration.»
  (make-local-variable 'font-lock-defaults)
  (make-local-variable 'indent-line-function)
  (setq font-lock-defaults '(config-font-lock-keywords))
  (setq indent-line-function 'config-indent-line))

(defun config-indent-line ()
  (if (string-match-p "^[:blank:]*==" (thing-at-point 'line))
      (indent-line-to 0)
      (indent-line-to 2)))

(defconst config-font-lock-keywords
  (list
   '(("^=.*$" 0 font-lock-type-face)
     '("[[:blank:]]*\\([[:alnum:]]+\\):\\(\\.+\\)$"
       (1 font-lock-variable-name-face)
       (2 font-lock-string-face))
   ))
```

Plaçons ce fichier dans le répertoire `~/ .emacs.d/` qui est utilisé par Emacs comme « répertoire de travail » (certains modes y placent des fichiers temporaires).

Nous allons désormais ajouter dans notre `.emacs` deux directives qui ont pour objectif d'indiquer à Emacs :

- 1 - où trouver les modes qui ne font pas partie de la distribution standard ;
- 2 - dans quel fichier se trouve la fonction `param-mode`.

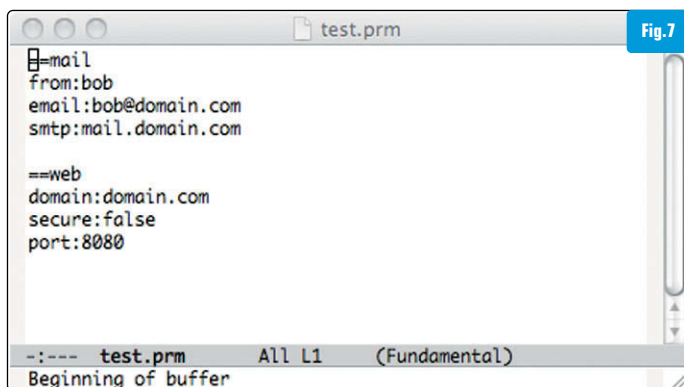


Fig.7

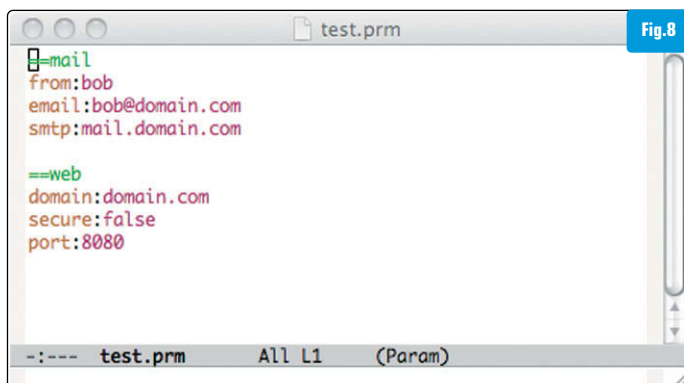


Fig.8

```
(add-to-list 'load-path «~/ .emacs.d/»)
(autoload 'param-mode «param-mode» nil t)
```

Lançons Emacs et éditons un fichier `demo.prm` [Fig.7]. Chargeons notre mode majeur de la manière suivante : `M-x param-mode` [Fig.8]. En nous déplaçant sur la deuxième ligne et en tapant TAB nous constatons que la ligne s'indente bien de deux espaces. Accélérons le processus en sélectionnant tout le buffer (`C-x h`) puis en appuyant sur la touche TAB pour indenter cette sélection [Fig.9]. L'idéal serait maintenant qu'Emacs charge automatiquement le `param-mode` pour tous les fichiers dont l'extension est `.prm`. Nous allons une nouvelle fois enrichir le `.emacs` pour préciser ce besoin :

```
(tool-bar-mode -1)
(setq initial-scratch-message nil)
(setq inhibit-startup-screen t)
(setq ring-bell-function 'ignore)
(add-to-list 'load-path «~/ .emacs.d/»)
(autoload 'param-mode «param-mode» nil t)
(setq auto-mode-alist '(("*.prm" . param-mode)))
```

CONCLUSION

Emacs est une créature informatique loin d'être évidente à dompter. Son extrême évolutivité récompensera cependant bien vite tout développeur qui fera l'effort d'apprendre les premiers raccourcis et qui acceptera de consacrer un peu de temps à l'élaboration d'un fichier configuration. Les possibilités d'enrichissement offertes par le langage Lisp sont infinies et permettent au développeur de se créer un environnement où toutes ses opérations et actions sont personnalisées et adaptées à ses besoins [Fig.10].

François-Xavier Bois

Cofondateur de KernIX. En charge de l'innovation. Auteur de plusieurs ouvrages consacrés aux technologies web (PHP, MySQL, WordPress). Créateur de `web-mode.el`.

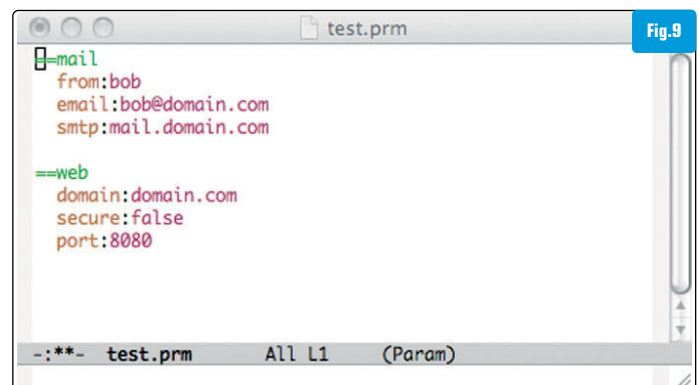


Fig.9

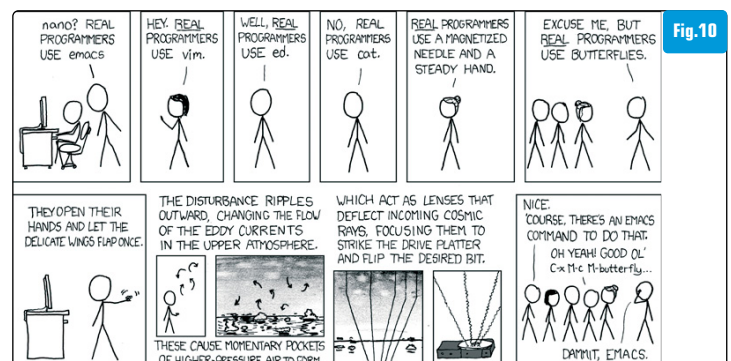


Fig.10

Démystifier le développement des plug-ins d'Eclipse

5^e partie

Evaluer les expressions XPath dans une fenêtre

Nous allons ajouter un menu, avec une icône qui s'affiche à la barre d'outils d'Eclipse. Ce menu associé à une commande permet d'afficher ou cacher une fenêtre (View). A l'intérieur de cette fenêtre, on affiche le résultat d'une expression XPath en utilisant le browser intégré à Eclipse. Cette fenêtre est la partie V.XPath du plug-in.

AJOUTER UNE COMMANDE AU PROJET

Dans l'éditeur de **MANIFEST.MF** -> sélectionnez l'onglet **Extensions** -> un clic sur le bouton **Add**.

Sélectionnez **org.eclipse.ui.commands** -> **Finish**.

Un clic droit sur l'extension ajoutée -> **New** -> **command**.

> Donner à la commande un ID

Cet ID (identificateur) va être utilisé pour attacher cette commande à un déclencheur, comme par exemple un menu, ou bouton. Le code dans la classe **defaultHandler** de cette commande est exécuté lorsque le déclencheur est cliqué. Donnez aussi un nom à cette commande **name**.

> Associer une classe à la commande «defaultHandler»

La classe assignée à **defaultHandler** doit étendre la classe **org.eclipse.core.commands.AbstractHandler**, dans laquelle nous allons fournir une méthode personnalisée de **execute(ExecutionEvent)**. Les instructions exécutées par une commande sont dans cette méthode. Dans le plug-in, cette méthode sert à afficher ou cacher, la fenêtre associée au V.XPath.

Donnez un nom à la classe dans **defaultHandler**, le nom utilisé dans l'exemple est **CommandHandler** -> Un clic sur le lien **defaultHandler** pour ouvrir le dialogue de création de classes Java **New Java Class**. Cliquez sur **Browse** pour ajouter une super classe.

Sélectionnez **org.eclipse.core.commands.AbstractHandler** -> **OK**.

Entrez le paquetage ou le chemin complet vers la classe (package).

Supprimez l'interface

Cliquez sur **org.eclipse.core.commands.IHandler** dans **Interfaces** -> **Remove**.

Le bouton **Finish**, va créer la classe et l'ouvre dans l'éditeur actif.

Pour mémoire, les instructions à exécuter sont à placer dans la méthode **execute(ExecutionEvent)**, nous allons y revenir.

AJOUTER UN MENU

Nous allons attacher ce menu, à la commande définie avant. Quand on clique sur le menu, les instructions de la commande seront exécutées. L'onglet **Extension** -> Un clic sur le bouton **Add** [Fig.1].

Sélectionnez **org.eclipse.ui.menus** dans la liste

Finish [Fig.2]. Clic droit sur **org.eclipse.ui.menus** -> **New** -> **menu contribution** [Fig.3].

> Renseigner le champ locationURI

Entrez la ligne suivante dans locationURI

toolbar:org.eclipse.ui.main.toolbar?after=additions

Note : LocationURI désigne l'endroit où le menu doit apparaître, ici on demande à Eclipse de l'afficher dans la barre d'outils d'Eclipse (ou toolbar).

Un clic droit sur **menuContribution** -> **New** -> **toolbar**.

Un clic droit sur **toolbar** qu'on vient d'ajouter -> **New** -> **command**.

La commande qu'on vient d'ajouter est le déclencheur de la commande ajoutée comme extension (**myCommand**). Son id par conséquent est le même **commandId**. Eclipse exécutera **myCommand**, quand le menu attaché est cliqué.

Entrez une étiquette ou **label**. Elle est affichée dans la barre d'outils d'Eclipse. Si vous voulez ajouter une icône dans la barre d'outils, cliquez sur le bouton **Browse** dans le champs **icon** pour sélectionner une icône (l'icône doit être dans le projet)

On termine toujours par **Save All**.

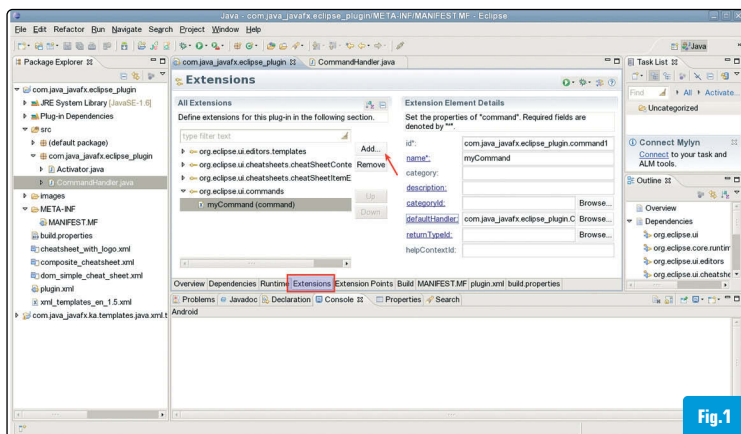


Fig.1

Ajouter une extension au menu

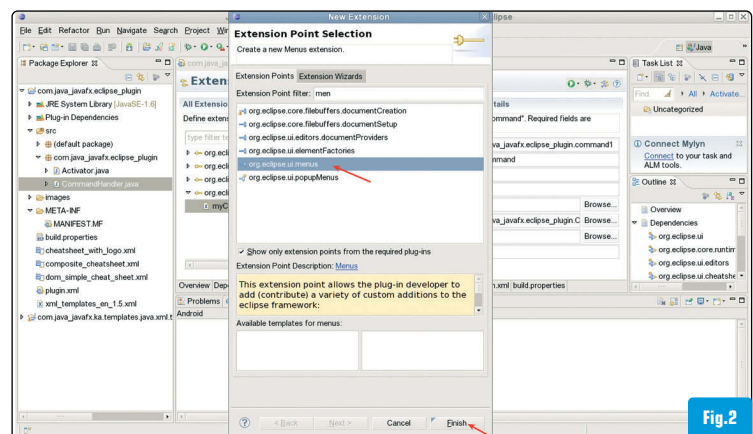


Fig.2

Ajouter un point d'extension

> Un petit test

Dans l'éditeur **MANIFEST.MF** -> l'onglet **Overview** -> rubrique **Testing** -> Un clic sur le lien **Launch Eclipse application** [Fig.4]. Notez que le menu avec son icône est affiché dans la barre d'outils.

AJOUTER UNE FENÊTRE (VIEW) AU PROJET

Nous avons besoin d'une fenêtre ou **view**, pour afficher la partie Visual XPath (**V.XPath**) du plug-in. On commence donc par ajouter une fenêtre au projet.

Dans l'éditeur de **MANIFEST.MF** -> dans l'onglet **Extensions** -> un clic sur le bouton **Add** [Fig.5].

Sélectionnez **org.eclipse.ui.views** dans la liste. **Finish** [Fig.6]. Un clic droit sur **org.eclipse.ui.views** -> **New** -> **view** [Fig.7].

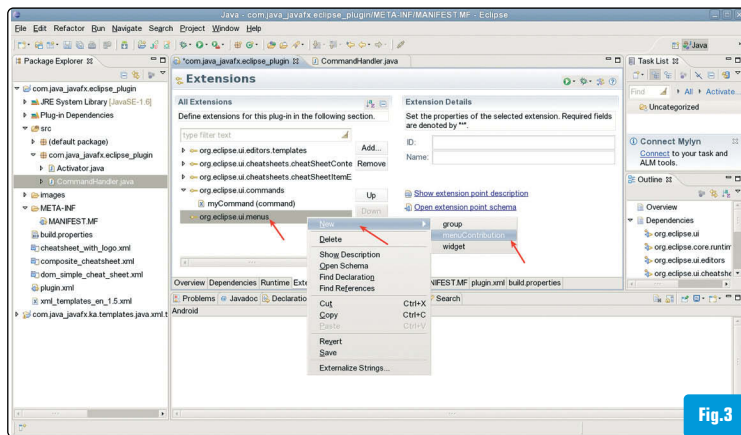


Fig.3

Ajouter une contribution

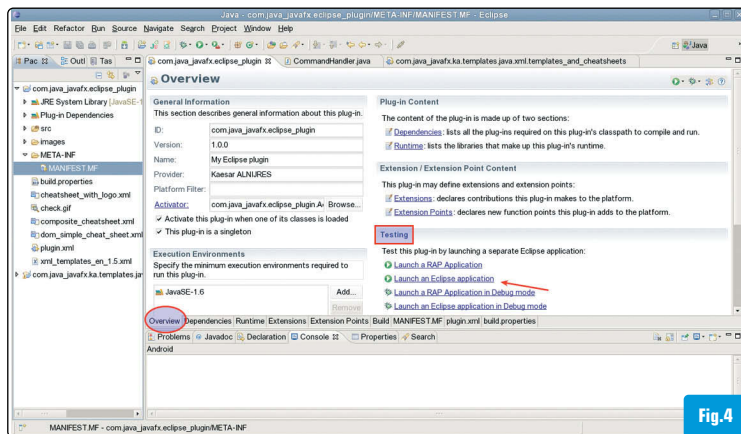


Fig.4

Tester le menu

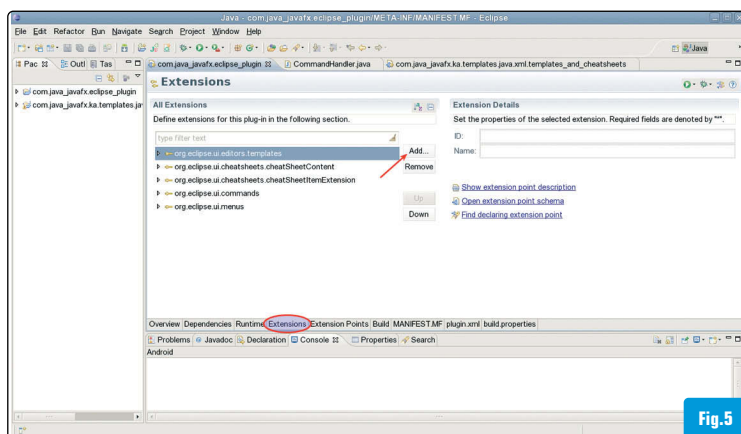


Fig.5

L'extension pour la view

> Les options de la view

Donnez à la **view** un ID **id(1)**, un nom dans **name(2)**. Le nom est affiché dans la barre de la fenêtre (**view**).

Donnez un nom au choix à la classe, qui va être attachée à la **view class(3)**. Cette classe est appelée quand la view est affichée. Maintenant, pour ajouter la classe, un clic sur le lien **class**.

Ceci ouvrira le dialogue **New Java Class**.

Pour afficher une icône dans la barre de titre de la view. Dans le champ **icon(4)**, cliquez sur **Browse** pour sélectionner une icône dans le projet [Fig.8].

Dans **New Java Class** dialog, la **Superclass** est ajoutée automatiquement par Eclipse -> **Finish**

La classe attachée à la view est ouverte dans l'éditeur actif [Fig.9].

L'affichage au sein de la fenêtre ou **View** se passe dans la méthode

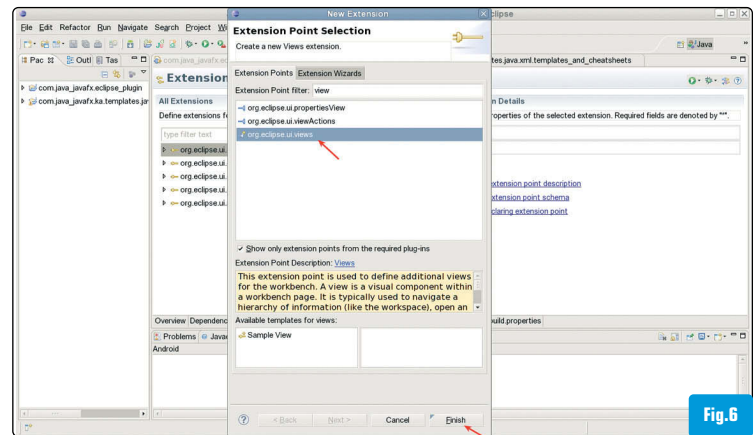


Fig.6

Sélectionner le point d'extension

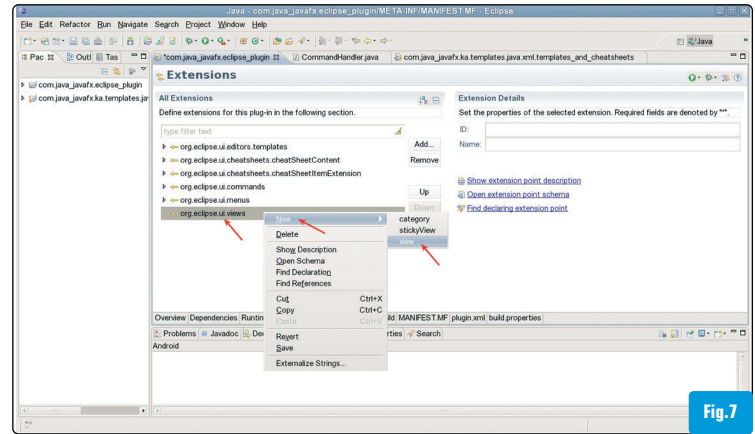


Fig.7

Ajouter une fenêtre ou view

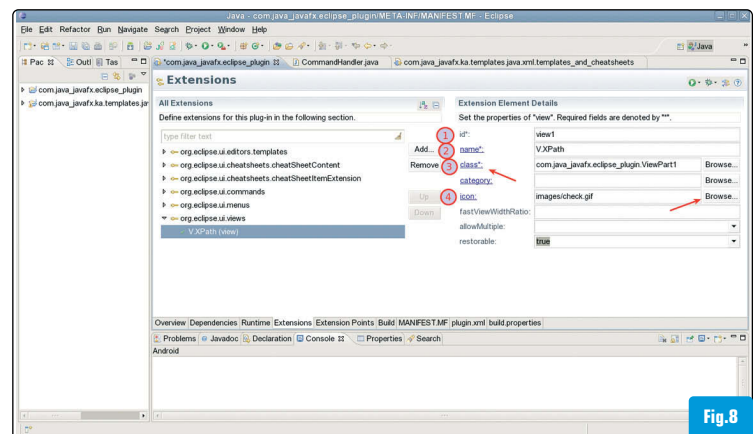


Fig.8

Options de la view

createPartControl(Composite). Cette méthode est appelée après la construction de la view [Fig.10].

Note : La view va être affichée ou cachée en utilisant la classe de la commande principale **defaultHandler**

> Ajouter des instructions pour afficher la view

Comme test, nous allons afficher la view (vide pour l'instant), en utilisant la classe **defaultHandler**, attachée à la commande principale, nommée **CommandHandler** et définie avant.

Pour ouvrir la classe **CommandHandler** dans l'éditeur actif -> dans l'éditeur de **MANIFEST.MF** -> dans l'onglet **Extensions** -> un clic sur le lien **defaultHandler** de la commande (**myCommand**) [Fig.11].

execute(ExecutionEvent) méthode de la classe **CommandHandler**

listing 1 : la méthode execute de CommandHandler

```
@Override
public Object execute(ExecutionEvent event) throws Execution
Exception {
    try {
        //obtenir la fenêtre active
        IWorkbenchWindow
        fWindow=PlatformUI.getWorkbench( ).getActiveWorkbenchWindow();

        //obtenir la page active (perspective)
        IWorkbenchPage page=fWindow.getActivePage();

        //afficher la view dont l'ID est «view1»
        page.showView(«view1»);

    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return null;
}
```

[Fig.12]

> Un petit test

On lance une nouvelle instance d'Eclipse. Un clic sur l'icône dans la barre d'outils d'Eclipse, ouvre la view (fenêtre, vue), ajoutée.

EXÉCUTER LA CLASSE XPATH EVAL DANS LA VIEW

Souvenez-vous, l'affichage dans la view, se fait dans la méthode **createPartControl(Composite)**, de la classe attachée à la view et qui étend la classe **ViewPart**

listing 2 : la class viewPart1

```
public class ViewPart1 extends ViewPart {
    private File file;
    private XPathEval path;

    ...

    ...

    ...

    @Override
    public void createPartControl(Composite parent) {
```

```
//créer un composant parent (Composite), modifier sa disposition
Composite c = new Composite(parent, 0);
c.setLayout(new FormLayout());
c.setBounds(0, 0, 800, 800);

FormData fd_browser = new FormData();
fd_browser.top = new FormAttachment(0, 5);
fd_browser.left = new FormAttachment(0, 2);
fd_browser.bottom = new FormAttachment(0, 400);
fd_browser.right = new FormAttachment(0, 500);

final Browser browser;

try {

    // obtenir la racine des fichier dans le fichier .jar du plug-in
    File f = new File(FileLocator.toFileURL(
        Platform.getBundle(Activator.getDefault()).
        PLUGIN_ID).getEntry(«/»).toURI());

    //utiliser example.xml, car l'utilisateur n'a pas sélectionné
    un autre fichier
    file = new File(f, «example.xml»);

    //construire un objet
    path = new XPathEval(file);

    //l'expression XPath passée à la méthode evalAll(String)
```

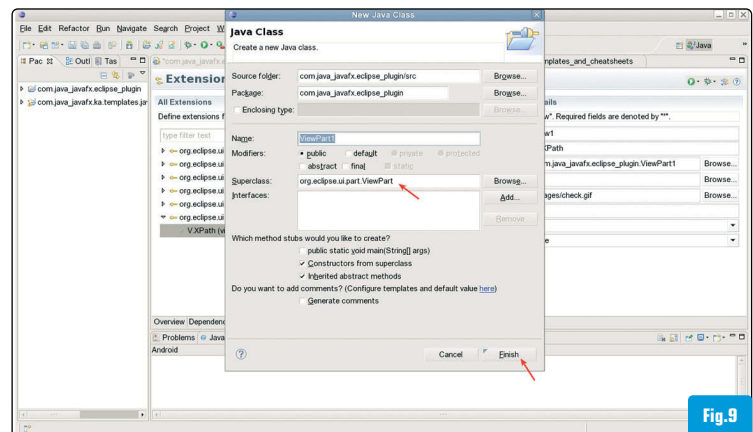


Fig.9

La super class de ViewPart

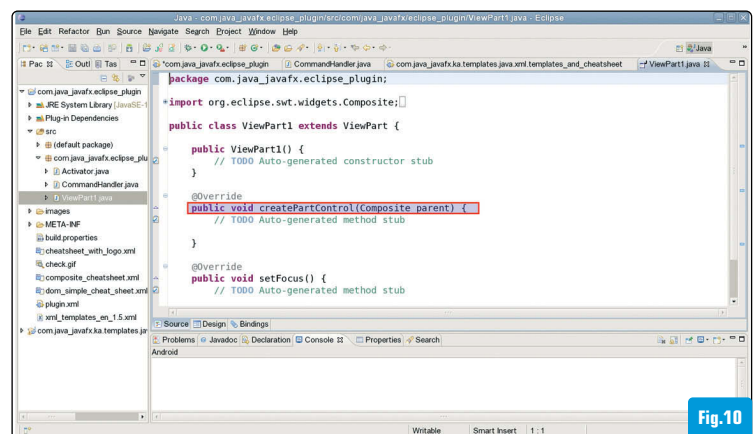


Fig.10

La méthode createPart pour l'affichage

```
est ici une chaîne de caractères
//vide. Pas de modification dans exemple.xml, car une chaîne
vide n'est pas une expression //XPath valide
String htmlContent = path.evalAll(«»);

//Construire un objet Browser
browser = new Browser(c, SWT.NONE);
browser.setBounds(0, 0, 400, 400);
browser.setLayoutData(fd_browser);
c.pack();

//si htmlContent n'est pas null, l'afficher dans le browser
if (htmlContent != null)
    browser.setText(htmlContent);

//une zone de texte pour entrer une expression XPath
final Text text = new Text(c, SWT.BORDER);
text.setBackground(c.getDisplay().
    getSystemColor(SWT.COLOR_YELLOW));
FormData fd_text = new FormData();
fd_text.top = new FormAttachment(msg, 2);
fd_text.left = new FormAttachment(0, 100);
fd_text.right = new FormAttachment(0, 250);
text.setLayoutData(fd_text);

//un bouton pour lancer l'évaluation de l'expression
final Button btn = new Button(c, SWT.PUSH);
```

```
FormData fd_button = new FormData();
fd_button.top = new FormAttachment(text, 2);
fd_button.left = new FormAttachment(0, 100);
fd_button.right = new FormAttachment(0, 250);
btn.setLayoutData(fd_button);

btn.setText(«evaluate»);

//attacher un SelectionListener au bouton
btn.addSelectionListener(new SelectionAdapter() {
    public void widgetSelected(SelectionEvent event) {
        //construire un nouvel objet XPath
        path = new XPathEval(file);

        //évaluer l'expression entrée par l'utilisateur
        String html = path.evalAll(text.getText());

        //afficher le XML modifié sous forme d'HTML (si l'expression
est valide)
        browser.setText(html);
    }
});

//ajouter un autre bouton pour laisser l'utilisateur sélectionner
un fichier .xml
final Button btnFile = new Button(c, SWT.PUSH);
```

```
FormData fd_button2 = new FormData();
fd_button2.top = new FormAttachment(btn, 2);
fd_button2.left = new FormAttachment(0, 100);
fd_button2.right = new FormAttachment(0, 250);
btnFile.setLayoutData(fd_button2);

btnFile.setText(«select file»);
```

```
//ajouter un SelectionListener au deuxième bouton
btnFile.addSelectionListener(new SelectionAdapter() {
```

```
    public void widgetSelected(SelectionEvent event) {
        //ajouter un dialogue FileDialog pour sélectionner un
fichier sur le disque dur
```

```
        FileDialog dialog = new FileDialog(btn.getShell(), SWT.OPEN);
```

```
//ouvrir FileDialog et retourner le résultat
String sf = dialog.open();
//la chaîne de caractères (String) retournée n'est pas
null, donc utiliser le fichier sélectionné
//pour construire un objet XPathEval et afficher le
résultat de l'expression dans le browser
```

```
    if (sf != null)
        file = new File(sf);

    {
        path = new XPathEval(file);
        String html = path.evalAll(«»);
```

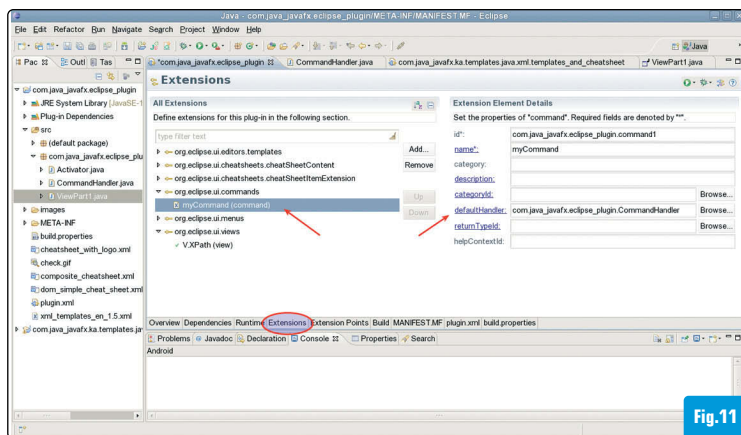


Fig.11

Ouvrir defaultHandler

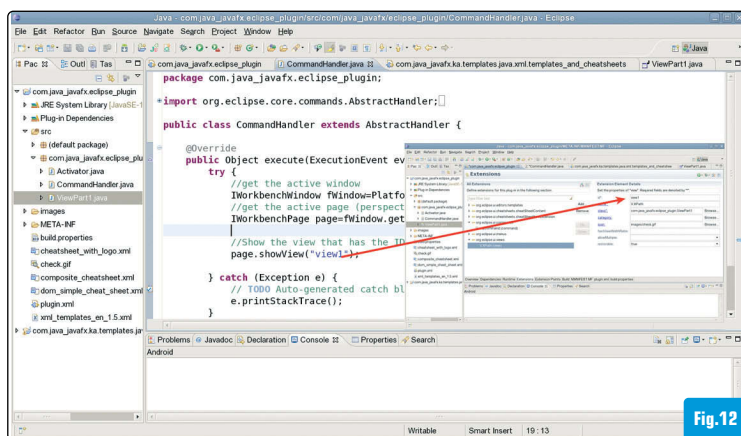


Fig.12

La méthode execute de CommandHandler



```
browser.setText(html);
}

}

});

} catch (Exception e) {
// TODO Auto-generated catch block
e.printStackTrace();
}

}
```

AFFICHER ET CACHER LA VIEW EN UTILISANT DEFAULTHANDLER

Nous allons modifier le contenu de la méthode **execute(ExecutionEvent)**, utilisée précédemment pour afficher la view (vue) vide.

Nous voulons ici afficher ou cacher la view en utilisant la classe **CommandHandler (defaultHandler)**. L'icône ajoutée à la barre d'outils, joue le rôle d'un bouton à bascule. On peut sauvegarder l'état d'une commande dans une sous-classe de **org.eclipse.core.commands.State**. De cette façon on peut afficher ou cacher la view, selon son état précédent. L'état d'une commande a un identificateur ou ID. Par exemple, pour un état à deux positions ou bascule, on utilise un ID à valeur boolean de **org.eclipse.ui.commands.toggleState**. Pour faire encore plus facile nous allons utiliser une classe prédéfinie **RegistryToggleState**. L'avantage, de cette classe prédéfinie, est de sauvegarder l'état de la commande, entre deux sessions d'Eclipse (persistance). Pour modifier l'état d'une commande, on utilise la classe **HandlerUtil**. Sa méthode **toggleCommandState(Command)** modifie l'état et retourne l'ancienne valeur.

> Ajouter un état à la commande myCommand

Dans l'éditeur de **MANIFEST.MF** -> Dans l'onglet **Extensions** -> un clic droit sur **myCommand** -> **New** -> **state**.

> L'ID et la classe de l'état

Entrez **org.eclipse.ui.commands.toggleState** dans le champs **id**, et **class=org.eclipse.ui.handlers.RegistryToggleState:true** (true ou false par défaut).

> La nouvelle version de la méthode execute(ExecutionEvent)

Cette version affiche ou cache la view.

listing 3 la méthode execute de CommandHandler

```
public class CommandHandler extends AbstractHandler {
//garder une instance de la view. A utiliser avec hideView
private ViewPart1 viewPart1=null;

@Override
public Object execute(ExecutionEvent event) throws ExecutionException {

try {
//la commande courante
Command command = event.getCommand();
```

```
//cette instruction change l'état de la commande et retourne
l'ancienne valeur
boolean oldValue = HandlerUtil.toggleCommandState(command);

IWorkbenchWindow fWindow=PlatformUI.getWorkbench().getActiveWorkbenchWindow();
IWorkbenchPage page=fWindow.getActivePage();

//si oldValue (ancienne valeur de l'état) est false, affiche
la view
if(!oldValue)
viewPart1= (ViewPart1)page.showView("view1");

else
{
//la view est visible
//si ce n'est pas la même session d'Eclipse viewPart1 est null,
//l'instance d'une view n'est pas persistante, uniquement
l'état de la commande (true or false)
//nous utilisons un autre moyen pour cacher la view
if(viewPart1 ==null)
{
//obtenir toutes les références des views visibles dans
cette page (perspective)
IViewReference[] viewReferences=page.getViewReferences();

for(IViewReference reference:viewReferences)
if(reference.getId().equals("view1"))
page.hideView(reference);
}
else //nous sommes dans la même session viewPart1 n'est pas null
page.hideView(viewPart1);
}

} catch (Exception e) {
// TODO Auto-generated catch block
e.printStackTrace();
}

return null;
}
```

> Un petit test

Dans l'éditeur de **MANIFEST.MF** -> **Overview** -> **Launch Eclipse application**

> Afficher et cacher la view

L'icône est pressée, la fenêtre est affichée

La fenêtre est cachée si l'icône n'est pas pressée

ESSAYER V.XPATH

Entrez une expression XPath. Quand vous cliquez sur **evaluate** le résultat est surligné dans la fenêtre V.XPath.

Kaesar Alnijres

Développeur Java - Leader jug-Cergy - <http://www.java-javafx.com>

Comprendre le fonctionnement de la machine virtuelle Java

Depuis 1996, Java et la JVM ont envahi nos équipements pour devenir des éléments incontournables de notre quotidien. Avant de s'intéresser aux détails et aux forces de la JVM, il est important de comprendre la relation entre le langage Java et cette dernière.

Au démarrage, Java se voulait un langage multiplateforme, essentiellement guidé par le principe WORA : "Write Once, Run Anywhere". La JVM ne comportait qu'un nombre limité d'optimisations par rapport à aujourd'hui, et Java était critiqué, à raison, pour sa lenteur. C'est en 2000 que le langage a bénéficié d'un second souffle avec l'intégration de Hotspot dans la JVM 1.3. La technologie Hotspot, quant à elle, est motivée par une toute autre problématique : concevoir une technologie qui maintienne l'aspect multiplateforme de Java, tout en profitant des optimisations de chaque machine (type de CPU, architecture matérielle).

Java est le langage dans lequel nous développons nos applications, Hotspot est le support par lequel nos applications sont de plus en plus optimisées à chaque nouvelle version de la JVM.

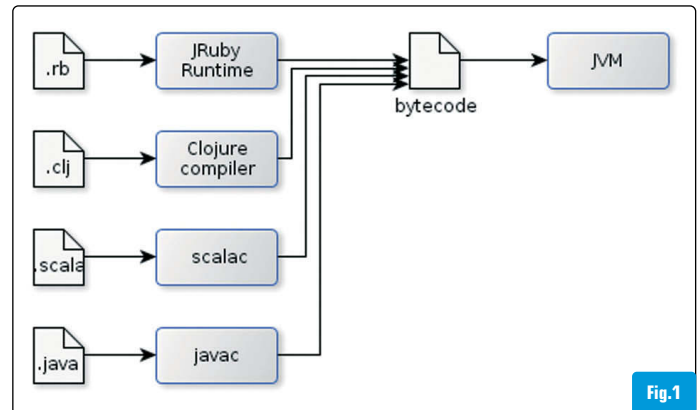


Fig.1

> Concepts majeurs

Chaque application Java est constituée de fichiers source (.java) qui sont compilés en bytecode. Ce bytecode est ensuite utilisé par la JVM et transformé en code natif.

Grâce à cette séparation entre le langage et la plateforme d'exécution, nous tirons aujourd'hui de nombreux bénéfices : les applications Java sont de plus en plus optimisées sans avoir à être recompilées, d'anciens langages ont été portés sur la JVM pour pouvoir profiter de ces mêmes optimisations (Jython, JRuby) et de nouveaux langages ont émergé pour combler les lacunes de Java (Clojure, Scala) [Fig.1]. La JVM s'occupe de la gestion de la mémoire en segmentant la heap en plusieurs sections : la "Young Generation" est l'espace dans lequel tous les objets sont créés; la "Old Generation", quant à elle, contient les objets ayant une longue durée de vie. Il existe deux autres zones mémoire que sont la "Permanent Generation" qui contient la définition des classes de l'application et le "Code Cache" qui est l'espace dans lequel la JVM stocke et manipule le code optimisé [Fig.2].

Tout au long de l'exécution d'une application, la JVM maintient des compteurs qui lui permettent de détecter le code souvent exécuté. C'est sur ce code que les optimisations vont être les plus utiles et c'est donc sur celui-ci que la JVM travaille (Just-In-Time Optimisation). Le cycle d'exécution d'un fragment de code commence donc toujours en mode interprété. Lorsqu'il est détecté par le profiler, il est ensuite passé à l'optimiseur avec des données complémentaires sur son exécution (nombre d'appels, code appelant, ...). Une fois que

l'optimiseur a appliqué des transformations sur ce code, il stocke le résultat dans le "Code Cache" [Fig.3].

> Le compilateur javac optimise-t-il le code ?

Pour éviter toute optimisation inutile, le compilateur javac n'effectue presque aucune opération sur le code source Java. Prenons l'exemple d'un "Hello world !" dans lequel on a ajouté du code mort.

Si l'on compile cette classe et que l'on analyse le bytecode produit, on constate que la méthode empty() a été compilée en l'état, qu'elle est toujours appelée depuis la méthode main(), et que la méthode dead() et la variable bar sont également présentes dans le bytecode. Ce comportement est normal : la quasi-totalité des optimisations sur le code sont effectuées par Hotspot, durant l'exécution de l'application, si et seulement si le code est souvent utilisé. Dans le cas présent, la méthode dead() ne sera jamais supprimée car elle n'est jamais utilisée, sa suppression serait donc une perte de temps. La boucle d'appels de la méthode empty() sera, quant à elle, bien supprimée après un certain nombre d'itérations.

> Temps de chauffe

Au démarrage d'une application, l'intégralité du bytecode est interprétée par la JVM. La JVM entre alors dans une phase de chauffe, durant laquelle elle optimise les fragments de code fréquemment utilisés jusqu'à ce qu'elle atteigne un point d'équilibre permettant à l'application de fonctionner à vitesse maximale.

Pour bien se rendre compte de ce temps de chauffe, on peut exécuter la classe NoOptimisation en mode interprété uniquement (-Xint), en mode mixte (-Xmixed, le mode par défaut) et en mode compilé (-Xcomp) uniquement en comparant les temps d'exécution.

On voit que le mode interprété est plus long que le mode mixte, ce qui est logique. Cependant, on remarque aussi que le mode compilé est considérablement plus long que le mode mixte. Ceci est normal :

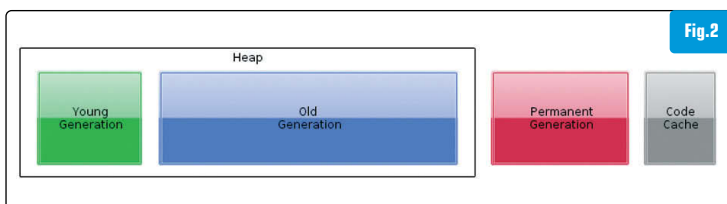


Fig.2



dans ce mode, la moindre méthode invoquée se retrouve compilée, même si elle n'est appelée qu'une seule fois, ce qui est inutile.

Le mode mixte de la JVM est donc le mode qui permet d'obtenir les meilleures performances le plus rapidement possible.

> Compilation du code

La JVM peut compiler les fragments de code "hot" en code natif. Cette compilation se fait alors en utilisant tout le jeu d'instructions du processeur sur lequel l'application s'exécute. Dans ce cas de figure, le code compilé est stocké dans le Code Cache et tous les pointeurs vers ce code sont réécrits pour référencer la version optimisée. S'il s'agit d'une compilation d'un bloc de code particulier dans une méthode, par exemple une boucle, on parle alors de "On Stack Replacement (OSR)", la JVM modifie la pile d'instructions de la méthode pour que le prochain GOTO mène sur le code compilé et non plus sur le code interprété.

On peut voir la JVM à l'oeuvre en activant les logs de compilation (-XX:+PrintCompilation).

Le format utilisé est le suivant : d'abord, un timestamp indique à quel moment (timestamp) la compilation a eu lieu. Ici, les compilations ont eu lieu à 61ms et 71ms.

Ensuite, un nombre entier indique l'id de compilation associé à l'optimisation. Il existe deux compteurs distincts, le premier pour les compilations de méthodes et le second pour les compilations OSR, c'est pourquoi les deux premières lignes portent le même id.

Le champ suivant est une série de caractères donnant des indications sur la nature de la compilation.

- b La compilation a bloqué l'exécution du programme;
- * La compilation a donné lieu à un wrapper en code natif;
- % La compilation a donné lieu à un OSR;
- ! La méthode compilée peut lever des exceptions;
- s La méthode compilée est synchronisée;
- n La méthode est déclarée comme du code natif.

On peut donc lire que la méthode `empty()` a été compilée à 61ms et qu'elle représente un seul octet de code. On voit également qu'un bloc de la méthode `main` a été compilé et remplacé à la volée (OSR) à 61ms. Enfin, la troisième ligne nous indique que la JVM a détecté à 71ms qu'une de ses optimisations a produit un code incorrect. Elle a désoptimisé ce code et est revenue en mode interprété. Le bloc optimisé a été marqué comme not-entrant, ce qui veut dire qu'il ne peut plus être exécuté. Comme il s'agit d'un retour en arrière, la JVM utilise l'id de compilation de l'optimisation annulée. Ceci explique pourquoi la troisième ligne a l'id 1 : elle concerne l'optimisation #1 faite à la ligne 2.

> Fusion et remodelage de code

L'une des optimisations effectuée par la JVM consiste à remplacer l'appel d'une méthode par le corps de cette méthode (inlining). Dans notre cas, l'appel de la méthode `empty()` peut être fusionné dans la méthode `main()`. Et c'est exactement ce que fait la JVM. On peut s'en rendre compte en affichant les logs d'inlining (-XX:+UnlockDiagnosticVMOptions -XX:+PrintInlining).

```
$ java -client -XX:+UnlockDiagnosticVMOptions -XX:+PrintInlining
NoOptimisation
                                @ 8   NoOptimisation::empty (1 bytes)
    inline (hot)
Hello world
```

La méthode `empty` de la classe `NoOptimisation` a été fusionnée car la JVM a détecté qu'elle était souvent appelée.

La taille des méthodes est un facteur très impactant pour la fusion de code. Si une méthode dépasse la taille autorisée (-XX:MaxInlineSize=), elle ne sera pas fusionnée dans d'autres blocs de code. À l'opposé, si une méthode est considérée comme triviale, très courte (-XX:MaxTrivialSize=), elle sera presque immédiatement fusionnée dans le code appelant. Typiquement, les getters et setters sont toujours très rapidement fusionnés.

La fusion de code par la JVM permet de rendre de nouvelles optimisations possibles. Reprenons l'exemple de code présenté par Brian Goetz dans la conférence "Towards A Universal VM" (Devoxx 2009).

```
public interface Holder<T> {
    T get();
}

public class MyHolder<T> implements Holder<T> {
    private final T content;

    public MyHolder(T someContent) {
        content = someContent;
    }

    @Override public T get() {
        return content;
    }
}

public String getFrom(Holder<String> h) {
    if(h == null)
        throw new IllegalArgumentException("h cannot be null");
    else
        return h.get();
}

public String doSomething() {
    MyHolder<String> holder = new MyHolder<>("Hello World");
    return getFrom(holder);
}
```

Analysons comment la méthode `doSomething()` va être optimisée. Dans un premier temps, la JVM va fusionner le code de la méthode `getFrom()` dans la méthode `doSomething()` car la méthode `getFrom()` est courte. La méthode `doSomething` va donc devenir :

```
public String doSomething() {
    MyHolder<String> holder = new MyHolder<>("Hello World");
    if(holder == null)
        throw new IllegalArgumentException("h cannot be null");
    else
        return holder.get();
}
```

Grâce à cette fusion, on constate que la branche `if(holder == null)` ne pourra jamais être exécutée car la variable `holder` est toujours initialisée. Cette branche est devenue du code mort, la JVM va donc la supprimer et aboutir au code suivant :


```
public String doSomething() {
    MyHolder<String> holder = new MyHolder<>("Hello World");
    return holder.get();
}
```

La méthode `getFrom()` référençait l'interface `Holder`, ce qui obligeait la JVM à effectuer l'invocation de la méthode `get()` à travers l'interface. À présent la JVM sait qu'elle a affaire à un type concret et peut donc remplacer le code de la méthode `get()` par son contenu, ce qui donne le pseudo code suivant :

```
public String doSomething() {
    MyHolder<String> holder = new MyHolder<>("Hello World");
    return holder.content;
}
```

Enfin, à travers l'analyse avancée du bytecode (Escape Analysis), la JVM va remarquer que la référence `someContent` est conservée en l'état dans la variable `content`, que cette dernière est une référence finale et qu'elle n'est jamais modifiée. Elle va donc pouvoir remplacer l'appel au constructeur de `MyHolder` et arriver au code suivant :

```
public String doSomething() {
    return "Hello world";
}
```

> Suppression des locks inutiles

La JVM procède aussi à un nettoyage du code synchronisé. Lorsqu'un fragment de code implique une section critique (bloc synchronisé) mais qu'il n'est utilisé que par un seul thread, la Hotspot peut modifier la nature du verrou utilisé, voire le supprimer complètement. Dans cet exemple se cachent de nombreux locks. En effet, la classe `StringBuilder` est synchronisée pour pouvoir être partagée entre plusieurs threads. Dans notre cas, c'est inutile et il aurait été préférable d'utiliser la classe `StringBuilder` qui n'est pas synchronisée. Si l'on exécute le programme avec l'élimination de locks (`-XX:+EliminateLocks`), on constate que la JVM a bien supprimé les entrées en section critiques et leurs sorties pour optimiser le programme.

Note : le paramètre `-XX:+PrintEliminateLocks` n'est disponible que sur une version de développement d'OpenJDK.

> Les deux modes de la JVM

La JVM dispose de deux modes de fonctionnement adaptés à des applications bien différentes.

Le premier mode est le mode "client", aussi nommé "C1". Il est utilisé pour les applications desktop, ayant une faible durée de vie (comptée en heures). Dans ce mode, la JVM démarre plus rapidement mais n'effectue que peu d'optimisations sur le code et a pour objectif de ne pas utiliser beaucoup de ressources matérielles. Le mode client est utilisé pour les applications desktop (NetBeans, IntelliJ, SQL Developer, ...).

Le deuxième mode est le mode "server", aussi appelé "C2" ou "opto". Il est utilisé sur les serveurs pour lesquels la durée de vie est bien plus longue (comptée en jours). Dans ce mode, la JVM démarre plus lentement compte tenu des optimisations effectuées pour atteindre le point d'équilibre. La limitation sur les ressources matérielles du

mode client n'existe plus ici : toutes les optimisations possibles sont activées. Le mode serveur est activé sur tous les serveurs d'applications (Tomcat, Glassfish, JBoss).

Pour aller plus loin : quelques pièges à éviter

Piège 1 : vouloir tout compiler

On pourrait avoir envie de compiler toute une application, puisque, par définition, le code natif est plus rapide que le code interprété. Cependant, c'est une mauvaise idée.

Que ce soit par le paramètre `-Xcomp` ou le paramètre `-XX:CompileThreshold=1`, toutes les méthodes seront compilées avant d'être exécutées et même fusionnées. Problème : lorsque le Code Cache est plein, la JVM peut tenter, en dernier recours, d'éliminer du code natif qui n'est plus utile (`-XX:+UseCodeCacheFlushing`). Si toutefois cette tentative se solde par un échec, la JVM cesse d'optimiser le code. Les threads de compilation sont arrêtés et ne peuvent pas être redémarrés. La JVM fonctionne alors sur ce qu'elle a pu optimiser jusque-là.

Piège 2 : forcer des paramètres de la JVM sans certitude

Lorsque l'on fait un tuning de la JVM, il faut s'assurer que celui-ci améliore réellement la situation et pouvoir expliquer ses choix. Il faut donc sans cesse mesurer les performances de son application pour vérifier qu'à chaque nouvelle version, le tuning apporte toujours un gain de performances.

Appliquer un ensemble de paramètres à une JVM sans avoir prouvé leur utilité revient à appliquer des solutions à un problème qui n'est pas diagnostiqué.

Piège 3 : optimiser prématurément son code

On l'a vu, la JVM modifie profondément le code qu'elle exécute. On tombe parfois sur des méthodes rendues illisibles pour des raisons de performance. Or à chaque nouvelle mise à jour, la JVM se dote d'améliorations qui rendent ces astuces obsolètes, voire nuisibles.

Un bytecode optimisé ne ressemble pas au source Java dont il est issu. Il vaut mieux écrire un code maintenable, éviter les optimisations prématurées et laisser la JVM s'occuper de rendre le code plus véloce.

> Conclusion

Lorsque l'on paramètre une JVM, il faut toujours appliquer la règle "Diagnostiquer avant de soigner", c'est-à-dire prouver l'existence d'un problème et prouver qu'il sera résolu par un paramétrage manuel avant d'appliquer ce dernier. Il faut également vérifier après chaque mise à jour de l'application ou de la JVM que ce paramétrage profite toujours au projet. Au démarrage d'un projet, il est préférable de laisser les réglages par défaut tant que ces derniers n'ont pas montré leurs limites.

La JVM est une plateforme très évoluée qui ne prend des décisions que lorsqu'elles sont justifiées par un contexte d'exécution. À chaque mise à jour, elle devient de plus en plus performante et c'est ce qui en fait une plateforme incontournable aujourd'hui.

Pierre Laporte - Formateur "Java Performance Tuning" et consultant chez Xebia. <http://blog.xebia.fr>

Construire des IHM dynamiques sur Android avec les Fragments

Depuis son apparition, la plateforme Android n'a cessé d'évoluer. Conçue à l'origine pour des smartphones de moins de 4 pouces, elle se doit aujourd'hui de répondre aussi bien aux besoins de smartphones dotés d'écrans toujours plus grands qu'à des tablettes allant jusqu'à 10 pouces sans oublier pour autant les Google TV.

Apparus avec Honeycomb, version d'Android exclusivement dédiée aux tablettes, les fragments sont désormais pleinement accessibles au sein de l'OS depuis la version 4 et peuvent également être utilisés sur d'anciennes moutures de la plateforme via la bibliothèque de support. Un fragment représente un comportement ou une portion d'interface utilisateur se trouvant au sein d'une activité ayant pour but d'aider à organiser les composants graphiques de manière réutilisable entre activités. De fait, un fragment pourra être réutilisé sur différents layouts. Le cycle de vie d'un fragment est étroitement lié à celui d'une activité du fait du lien fort les reliant. Ainsi, un fragment est instancié uniquement lorsqu'il est rattaché à une activité. Devenant alors actif, il n'est détruit que lorsqu'il est détaché de son activité par un retrait de l'interface ou bien suite à sa destruction. Tout au long du cycle de vie de l'activité parente, des méthodes sont appelées permettant l'interaction au sein des fragments qui la composent [Fig.1]. Modélisés au travers de la classe `Fragment`, les fragments bénéficient d'une API dédiée. Au sein d'une activité, leur gestion se fait grâce à un manager récupérable par un appel à `getFragmentManager()`. Ce dernier permet notamment de :

- Récupérer les fragments attachés à une activité via un appel à `findFragmentById()` ou `findFragmentByTag()` pour les fragments représentant seulement un comportement.
- Retirer des fragments de la pile back via `popBackStack()` ce qui permet de simuler l'appui sur la touche back.
- Créer des transactions encapsulant les opérations réalisées lors de la manipulation de fragments.

Créées suite à un appel à la méthode `beginTransaction()` du manager de fragments, les transactions sont utilisables au travers de la classe `FragmentTransaction`. Parmi les méthodes les plus utiles, on citera `add` et `remove` qui, comme leur nom l'indique, permettent l'ajout et le retrait d'un fragment mais également la méthode `replace` qui permet de remplacer le contenu d'un conteneur par un fragment donné. Enfin, la méthode `addToBackStack()` permet d'ajouter la

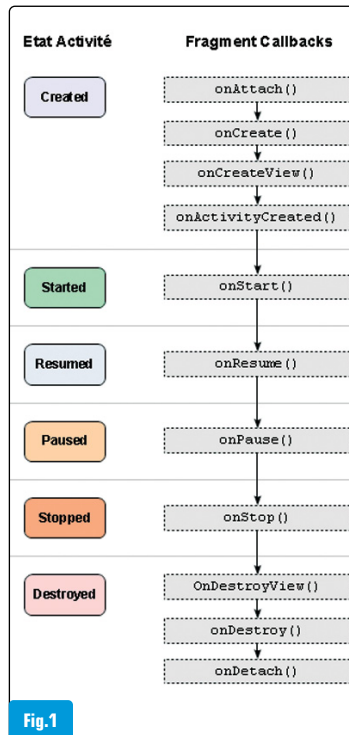


Fig.1

Lien entre une Activité et le cycle de vie d'un Fragment

transaction courante à la pile de la touche back alors que la méthode `commit()` applique l'ensemble des changements de la transaction courante. Pour terminer cette partie théorique sur les fragments, il est important de dire un mot sur les mécanismes de communication entre une activité et ses fragments. Au sein d'une activité, cela s'avère aisé puisqu'il suffit de récupérer le fragment désiré via la méthode `findFragmentById()` du `FragmentManager` et d'appeler ses méthodes publiques. Dans l'autre sens, c'est-à-dire permettre à un fragment de communiquer avec son activité de rattachement, la méthode `getActivity()` est mise à disposition au sein d'un fragment donnant accès à l'activité associée. Néanmoins, dans un souci de réutilisabilité, la bonne pratique va consister à définir une interface, au sein du fragment, proposant les méthodes de communication utiles et qui sera implémentée par l'activité parente. Enfin, lors de l'appel au callback `onAttach()` du fragment, il conviendra de vérifier que l'activité implémente bien cette interface et d'en garder une référence. C'est cette dernière qui sera utilisée pour communiquer avec l'activité, grâce aux méthodes exposées par l'interface.

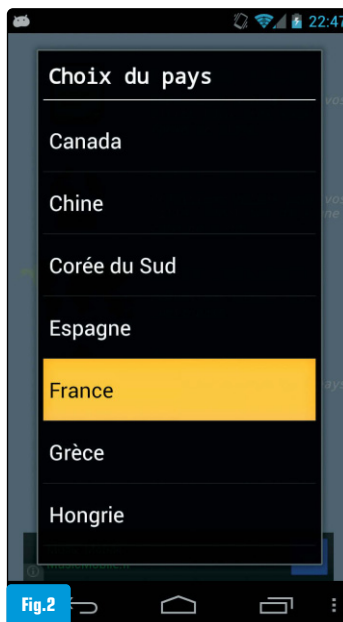


Fig.2

Activité listant les pays

> Etude de cas : FastMeteo

Les fondements théoriques des fragments présentés, nous allons maintenant passer à la mise en pratique via l'étude d'un cas réel : l'application FastMeteo, disponible sur le Play Store à l'adresse suivante : <https://play.google.com/store/apps/details?id=com.app.fastmeteo>. Développée dans le cadre du dossier Android de Programmez ! 145, l'application FastMeteo fournit la météo pour la ville courante de l'utilisateur mais également pour la plupart des villes du monde. Outre cette fonctionnalité de base, l'application propose également des cartes météo par pays.

Concrètement, cette fonctionnalité se divise en 2 activités : une première listant l'ensemble des pays pour lesquels une carte météo est disponible [Fig.2] et une seconde affichant la carte météo du pays sélectionné sur la liste précédente. Conçue à l'origine pour tourner sur des smartphones Android en version 2.2, l'application FastMeteo avait de fait

suivi une approche classique avec la création de 2 activités reliées par une action utilisateur sur la première au sein de laquelle le pays était sélectionné. Si ce fonctionnement était pleinement satisfaisant à l'époque et demeure juste acceptable aujourd'hui sur des smartphones aux écrans plus larges, il est simplement inutilisable sur des tablettes dont les écrans oscillent entre 7 et 10 pouces. En effet, la liste de pays s'affiche sur la totalité de l'écran, ce qui est une perte d'espace non négligeable et le fait de devoir revenir sur l'activité précédente pour changer de carte est plutôt désagréable pour l'utilisateur.

> Objectif

Afin d'adapter l'application aux tablettes, nous allons devoir partager l'écran en mode paysage afin de pouvoir y afficher sur le côté droit la liste des pays possédant une carte météo et sur le côté gauche, la carte météo sélectionnée. En outre, cette mise en place devra se faire en préservant le fonctionnement actuel pour l'orientation en mode portrait afin de ne pas dégrader le fonctionnement de l'application à ses 2500 utilisateurs actifs. Enfin, le partage de l'écran en mode paysage devra être appliqué uniquement pour des smartphones récents avec une taille d'écran d'au moins 4 pouces.

> Solution

Pour répondre à cette problématique, nous allons nous appuyer sur la puissance de l'API Fragment. Celle-ci va nous permettre d'adapter FastMeteo afin de la doter d'une IHM dynamique et modulaire répondant à la fois aux besoins des smartphones et des tablettes sans bouleverser l'architecture actuelle de l'application. Notre solution va consister à implémenter les comportements de nos 2 activités actuelles au sein de 2 fragments, nommés respectivement CountriesFragment et MapFragment. En mode portrait, notre application s'appuiera toujours sur 2 activités : MeteoActivity qui encapsulera le fragment CountriesFragment et MapActivity encapsulant le fragment MapFragment. En revanche, en mode paysage l'application utilisera seulement l'activité MeteoActivity englobant nos 2 fragments [Fig.3]. Le comportement dynamique de la solution se situant au niveau du CountriesFragment et plus précisément de la liste des pays qu'il affiche. En effet, lors du clic sur un pays, l'application suivant le mode d'affichage sélectionné décidera soit de mettre à jour le MapFragment avec la carte du pays courant soit de démarrer l'activité MapActivity qui aura la responsabilité de charger le MapFragment.

> Création des fragments

La mise en place de notre solution commence par la création des fragments qui vont constituer la partie fonctionnelle de l'application. Les activités que nous créerons par la suite serviront principalement à afficher les fragments et leur passer d'éventuels paramètres reçus depuis l'Intent. Le fragment CountriesFragment va lister l'ensemble des pays possédant une carte météo. Pour ce faire, il va étendre la classe ListFragment dont le rôle est similaire à celui de la classe ListActivity pour les activités. Elle possède notamment la méthode onListItemClick() qui nous servira pour récupérer le clic sur les éléments de la liste. Le code du fragment est le suivant :

```
/**
 * Fragment affichant la liste des pays ayant une carte météo.
 */
public class CountriesFragment extends ListFragment {
```

```
private boolean mDualPane;
private int mCurCheckPosition = 0;

@Override
public void onActivityCreated(Bundle savedInstanceState) {
    super.onActivityCreated(savedInstanceState);

    // Chargement des pays
    setListAdapter(new ArrayAdapter<String>(getActivity(), android.
R.layout.simple_list_item_activated_1,
        ResourcesCountries.getCountries(getResources())));

    // Vérifie que le frame embarquant le map fragment est présent
    dans l'UI
    View mapFrame = getActivity().findViewById(R.id.map);
    mDualPane = mapFrame != null && mapFrame.getVisibility() ==
View.VISIBLE;

    if (savedInstanceState != null) {
        mCurCheckPosition = savedInstanceState.getInt("curChoice", 0);
    }

    if (mDualPane) {
        // En mode partage d'écran, on surligne le pays sélectionné
        getListView().setChoiceMode(ListView.CHOICE_MODE_SINGLE);
        displayMap(mCurCheckPosition);
    }
}

@Override
public void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
    outState.putInt("curChoice", mCurCheckPosition);
}

@Override
```

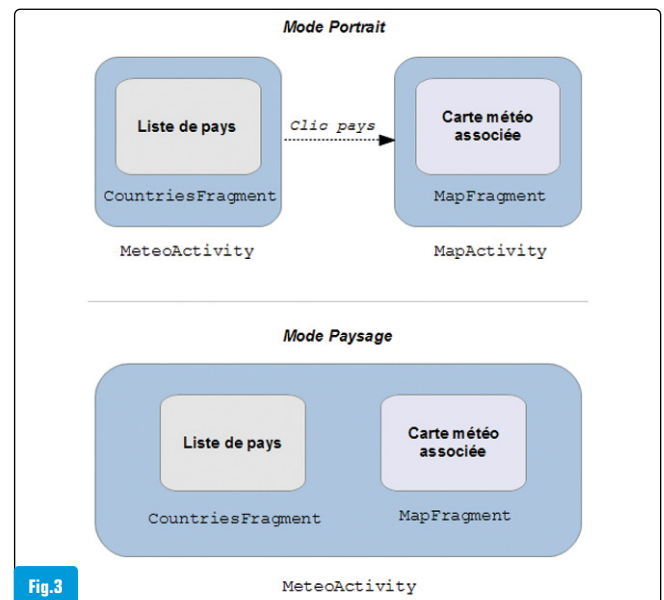


Fig.3

Solution dynamique retenue


```

public void onListItemClick(ListView l, View v, int position,
long id) {
    displayMap(position);
}

/**
 * Méthode affichant la carte météo.
 */
private void displayMap(int index) {
    mCurCheckPosition = index;

    if (mDualPane) {
        // En mode partage d'écran, on affiche la carte au sein du map
        // fragment
        getListView().setItemChecked(index, true);

        // On vérifie si le fragment n'est pas déjà affiché ou s'il
        // n'affiche pas le bon pays ==> auquel cas on le remplace.
        MapFragment mapFragment = (MapFragment) getFragmentManager
        ().findFragmentById(R.id.map);

        if (mapFragment == null || mapFragment.getShownIndex() != index) {
            mapFragment = MapFragment.newInstance(index);

            // Transaction pour remplacer le fragment existant par
            le nouveau
            FragmentTransaction ft = getFragmentManager().beginTransaction();
            ft.replace(R.id.map, mapFragment);
            ft.setTransition(FragmentTransaction.TRANSIT_FRAGMENT
            _FADE);
            ft.commit();
        } else {
            // Mode portrait ==> on lance l'activité Map Activity
            Intent intent = new Intent();
            intent.setClass(getActivity(), MapActivity.class);
            intent.putExtra("index", index);
            startActivity(intent);
        }
    }
}

```

Le fragment MapFragment va quant à lui être utilisé pour afficher la carte météo du pays sélectionné. La hiérarchie de vues du fragment est créée à l'appel de la méthode onCreateView() en inflatant le layout map_country qui était précédemment utilisé par l'application. Le code du fragment a l'allure suivante :

```

/**
 * Fragment secondaire affichant la carte météo d'un pays sélectionné.
 */
public class MapFragment extends Fragment implements Country
MeteoPublisher<MapFragment> {

    private MapCountryView mapCountryView;
    private ProgressBar progressBar;

```

```

/**
 * Création d'un map fragment initialisé avec le pays sélectionné.
 */
public static MapFragment newInstance(int index) {
    MapFragment mapFragment = new MapFragment();
    Bundle args = new Bundle();
    args.putInt("index", index);
    mapFragment.setArguments(args);
    return mapFragment;
}

public int getShownIndex() {
    return getArguments().getInt("index", 0);
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
    if (container == null) {
        // si container est null, la frame contenant le fragment
        // n'existe pas, il n'y a donc pas d'intérêt à créer la hiérarchie
        // de vues.
        return null;
    }

    // création de la vue depuis le fichier layout map_country actuel.
    View view = inflater.inflate(R.layout.map_country, container,
false);
    mapCountryView = (MapCountryView) view.findViewById(R.id.
mapCountry);
    progressBar = (ProgressBar) view.findViewById(R.id.progressBar);
    return view;
}

@Override
public void onDestroyView() {
    super.onDestroyView();

    if (mapCountryView != null) {
        mapCountryView.cancelLoad();
    }
}

@Override
public void onResume() {
    super.onResume();

    if (mapCountryView != null && progressBar != null) {
        int shownIndex = getShownIndex();
        // on récupère les informations du pays associé à shown
Index
        Integer[] mapObjects = ResourcesCountries.getCountryObject
(shownIndex);
        getActivity().setTitle(mapObjects[2]);
        mapCountryView.setCountryDrawable(mapObjects[0]);
        mapCountryView.setCities(getResources().getStringArray(map
Objects[1]));
        mapCountryView.setCountryMeteoPublisher(this);
    }
}

```

```
// Chargement de la météo pour les villes de la carte
mapCountryView.loadMeteoForCities();
}
}

public ProgressBar getProgressBar() {
    return progressBar;
}

public void displayMeteo(Meteo meteo) {
    Bundle args = new Bundle();
    args.putSerializable(Util.METEO_PARAM, meteo);
    // ouverture d'une dialogue lors d'un clic sur la météo d'une
    ville sur la carte
    getActivity().showDialog(MeteoActivity.DIALOG_CITY_ID, args);
}
}
```

> Création des activités

Le contenu de l'activité MeteoActivity est simplifié à l'extrême avec pour seule responsabilité le chargement du layout meteo_layout :

```
public class MeteoActivity extends DetailsMeteoActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.map_meteo_layout);
    }
}
```

En mode portrait, le layout map_meteo_layout se contente d'ajouter le fragment CountriesFragment en son sein :

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res
/android"
    android:layout_width="match_parent" android:layout_height=
"match_parent">
    <fragment class="com.app.fastmeteo.fragments.CountriesFragment"
        android:id="@+id/countries"
        android:layout_width="match_parent" android:layout_height=
"match_parent" />
</FrameLayout>
```

En revanche, la définition du layout en mode paysage va différer avec l'ajout de ce même fragment et le partage de l'écran horizontalement avec un FrameLayout qui servira à accueillir le fragment MapFragment. Placé au sein du dossier res/layout-land, son contenu est le suivant :

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res
/android"
    android:orientation="horizontal"
    android:layout_width="match_parent" android:layout_height=
"match_parent">
    <fragment class="com.app.fastmeteo.fragments.CountriesFragment"
        android:id="@+id/countries" android:layout_weight="30"
        android:layout_width="0px" android:layout_height="match
_parent" />
    <FrameLayout android:id="@+id/map" android:layout_weight="70"
        android:layout_width="0px" android:layout_height="match
_parent">
```

```
        android:background="?android:attr/detailsElementBackground" />
</LinearLayout>
```

Pour limiter l'utilisation de ce layout en mode portrait aux smart-phones ayant un écran de plus de 4 pouces, il faudrait rajouter un qualifient ciblant la taille de l'écran, ce qui donnerait le nommage suivant : layout-sw330dp-land. La seconde activité nommée MapActivity, utilisée uniquement en mode portrait, va servir à afficher le fragment MapFragment en son sein de la sorte :

```
/**
 * Activité secondaire utilisée pour afficher le map fragment quand
 * l'écran n'est pas assez large pour afficher tout au sein d'une seule
 * activité.
 */
public class MapActivity extends DetailsMeteoActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        if (getResources().getConfiguration().orientation == Configuration.
            ORIENTATION_LANDSCAPE) {
            // écran en mode paysage ==> pas besoin de cette activité.
            finish();
            return;
        }

        if (savedInstanceState == null) {
            // Ajout du map fragment au contenu de l'activité
            MapFragment mapFragment = new MapFragment();
            mapFragment.setArguments(getIntent().getExtras());
            getFragmentManager().beginTransaction().add(android.R.id.
                content, mapFragment).commit();
        }
    }
}
```

Un petit mot sur la classe abstraite DetailsMeteoActivity dont le rôle n'est pas réellement lié au travail réalisé pour adapter l'IHM de FastMeteo aux écrans plus larges mais qui sert à factoriser le code servant à afficher un dialogue détaillant la météo d'une ville cliquée sur la carte météo affichée.

> Résultat

La mise en place d'une IHM dynamique n'aura que très peu touché à l'architecture existante de FastMeteo puisqu'une grande partie du code existant aura pu être réutilisée, que ce soit dans les fragments ou les activités créés.

> Conclusion

L'apparition des fragments avec Android 3.0 et leur mise à disposition pour tous types de périphériques à partir de la version 4 aura mis en exergue la nécessité de repenser les approches existantes lors de la création d'IHM. Là où hier encore, on pouvait créer des activités sans se soucier de la manière dont leur contenu occuperait l'espace, il est essentiel à présent de réaliser des IHM modulaires dans le but de gagner en réactivité mais également en flexibilité avec une occupation de l'espace mieux pensée.

Sylvain Saurel – Ingénieur d'Etudes Java / JEE
sylvain.saurel@gmail.com

Comment combiner Amazon EC2 et S3 pour un service web ?

Amazon Web Services (AWS) propose des services de stockage en ligne et de serveurs virtualisés. Depuis, l'offre d'AWS s'est étoffée avec une gamme complète de services d'application et d'infrastructure pour répondre à l'ensemble des besoins allant du site web à l'application métier critique, du calcul haute performance au plan de reprise d'activité – pour ne citer que quelques cas d'usage.

Dans cet article, nous vous proposons de découvrir comment utiliser les services Amazon EC2 et Amazon S3 pour héberger des sites web évolutifs et hautement disponibles. Notez que le service de stockage Amazon S3 a dépassé les 1 300 milliards d'objets stockés, sur lesquels jusqu'à 830 000 requêtes sont exécutées chaque seconde.

L'utilisation des solutions Amazon Web Services est simplifiée pour permettre un accès facilité. L'ouverture d'un compte AWS ne nécessite que quelques minutes et se fait directement en ligne sur le site : aws.amazon.com. L'ensemble des services est accessible en self-service après avoir entré son numéro de carte bancaire. Amazon Web Services propose un niveau d'utilisation gratuite au cours de la première année. Ce niveau couvre une large gamme de services. Les entreprises pourront passer à la facturation électronique pour une gestion simplifiée.

> Une infrastructure globale

AWS est une plateforme informatique à haute fiabilité, redimensionnable et à bas coût dans le Cloud qui alimente des centaines de milliers d'entreprises dans 190 pays. Elle est organisée en 8 régions dans le monde dans lesquelles il est possible de déployer instantanément des ressources Cloud : 3 en Amérique du Nord, 3 en Asie, 1 au Brésil et enfin 1 en Europe (Irlande). Nous utiliserons cette dernière – dénommée eu-west-1 – dans la suite de l'article.

Chaque région comprend plusieurs « zones de disponibilité » (aussi appelé AZ – Availability Zone dans la console). Ces zones peuvent être vues comme des datacenters ou des groupes de datacenters distincts, reliés par un réseau à haut débit et à faible latence, et isolées de manière à ce qu'un événement sur une zone n'impacte pas les autres. La région Europe propose ainsi 3 zones nommées a/b/c. [Fig.1]

> Des centaines d'AMI pour EC2

Nous allons commencer par le service EC2 (Elastic Compute Cloud) qui propose à la demande des serveurs virtualisés dans plus de 16 tailles et de configurations différentes. Ces serveurs supportent une large gamme d'OS : Windows Server 2003/2008/2012, ainsi que de nombreuses distributions Linux : Ubuntu, Suse, Redhat, CentOS... La totalité des services AWS disposent d'API qui les rend tota-

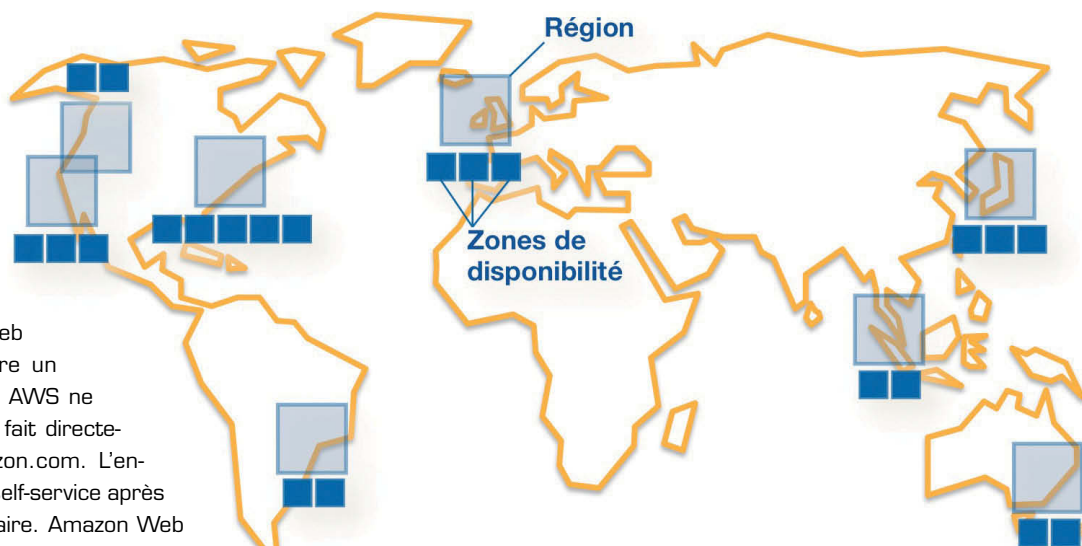


Fig.1 Régions et Zones AWS dans le monde

lement programmables et scriptables au travers de nombreux SDK (Java, .Net, PHP, Ruby, Python...). Dans la suite de l'article nous utiliserons le SDK Python 'boto' (<https://github.com/boto/boto.org>).

AWS dispose d'une bibliothèque des centaines de machines prêtes à l'emploi sous forme d'images AMI (Amazon Machine Image). On y trouve des images d'OS nus fournis et certifiés par les principaux éditeurs, ainsi que des piles applicatives prêtes à l'emploi. Une place de marché la complète avec des applications payantes à l'heure ou au mois des (aws.amazon.com/marketplace).

A titre d'exemple, nous démarrerons un serveur sur la base d'une image LAMP gratuite proposée par BitNami. Cette image porte la référence « ami-ab9490df » sur la région Europe. Nous la démarrons dans la zone « eu-west-1a » sur une machine de taille « m1.small » (1 cœur, 1,7 Go de mémoire).

> Sécurité et authentification

Afin d'accéder de manière sécurisée aux API d'AWS, on utilise un couple de clefs AWS_ACCESS_KEY et AWS_SECRET_KEY. En cas de compromission, ces clefs peuvent être révoquées et ré-générées.

Afin d'assurer la sécurité réseau des serveurs, un firewall virtuel obligatoire – appelé « Security Group » – est présent devant chaque serveur. Il est fermé par défaut et chaque port utilisé doit être ouvert explicitement. Dans notre exemple nous ouvrirons les ports TCP SSH(22) et HTTP(80).

Pour se connecter aux serveurs virtuels, une paire de clefs SSH est également nécessaire. La clef publique sera déployée automatiquement sur tous les serveurs. En revanche la clef privée sera gardée bien précieusement sur le poste client : c'est l'unique Sésame qui permet d'établir la connexion. Pour générer ces clefs, deux options sont possibles. La première option consiste à générer localement la paire de clefs avec l'outil « ssh-keygen » et d'importer la clef publique via une API EC2. La seconde est de faire générer la clef par AWS et de télécharger en HTTPS la clef privée. Une fois téléchargée, la clef privée est effacée des serveurs d'Amazon et seule la clef publique est conservée. C'est ce que nous illustrerons ici avec la commande « create_key_pair ».

Enfin AWS assure une sécurité des bâtiment, du réseau et des hyperviseurs comme en témoigne les certifications ISO 27001, SOC-1 & SOC-2, PCI DSS niveau 1. Le modèle de sécurité est dit « partagé » dans la mesure où AWS assure la sécurité des couches basses. La sécurisation des machines virtuelles reste à la charge des clients qui sont libres d'installer les patchs OS quand ils le souhaitent.

> Haute disponibilité

Il existe tout un éventail d'outils pour gérer la haute disponibilité des applications. Dans notre exemple, nous utiliserons la fonction d'Elastic IP. Nous commençons par réserver une adresse IP Internet statique. Cette adresse sera à notre disposition indéfiniment tant que nous ne la relâchons pas explicitement. Ensuite la commande « associate_address » permet d'associer cette adresse à n'importe quelle instance EC2 en quelques secondes.

On peut ainsi simplement créer un système actif/passif avec deux instances EC2. On associe l'adresse EIP à l'instance active. Pendant ce temps l'instance passive surveille cette première et en cas de défaillance, elle préemptera l'adresse IP avec un nouvel appel à la commande « associate_address » pour récupérer le trafic en quelques secondes. Mieux, la seconde instance pourra être déployée dans une zone de disponibilité distincte – par exemple eu-west-1b – afin de se protéger d'une rare mais possible interruption de service sur un datacenter.

Il existe d'autres moyens d'assurer une haute disponibilité des applications comme les services « auto-scaling » et « elastic load-balancer », cela pourra faire l'objet d'un autre article.

> Jouons un peu

Une fois le script exécuté avec succès, notre serveur est opérationnel et l'adresse IP de connexion disponible. On pourra s'y connecter en SSH et via l'interface web.

```
$ ssh -i demo.pem bitnami@54.246.112.44
The authenticity of host '54.246.112.44 (54.246.112.44)' can't
be established.
RSA key fingerprint is eb:cd:7d:71:8a:b8:be:2c:fc:1b:25:21:79:
3e:e8:44.
Are you sure you want to continue connecting (yes/no)? yes

Welcome to Ubuntu 12.04 LTS (GNU/Linux 3.2.0-26-virtual x86_64)

[...]
```

```
*** Welcome to the BitNami LAMPStack 5.3.14-1 ***
*** BitNami Wiki: http://wiki.bitnami.org/ ***
*** BitNami Forums: http://answers.bitnami.org/ ***
To run a command as administrator (user «root»), use «sudo <command>».
See «man sudo_root» for details.

$
```

[Fig.2]

Vous aurez remarqué que l'instance a pris environ deux minutes pour démarrer. S'il s'avère que la machine utilisée est trop ou pas assez performante, il est possible de changer sa taille par un simple arrêt-relance. Il est ainsi très facile de retailer à la hausse un serveur en cas de fort usage, puis à la baisse, par exemple la nuit et le week-end. La facturation se fait à l'heure d'utilisation. Par exemple notre serveur m1.small coûte \$0,065 par heure, soit environ 5 centimes d'euros de l'heure. Une configuration de taille double « m1.medium » à 3,7 Go de mémoire coûtera le double, soit environ 10 centimes d'euros de l'heure.

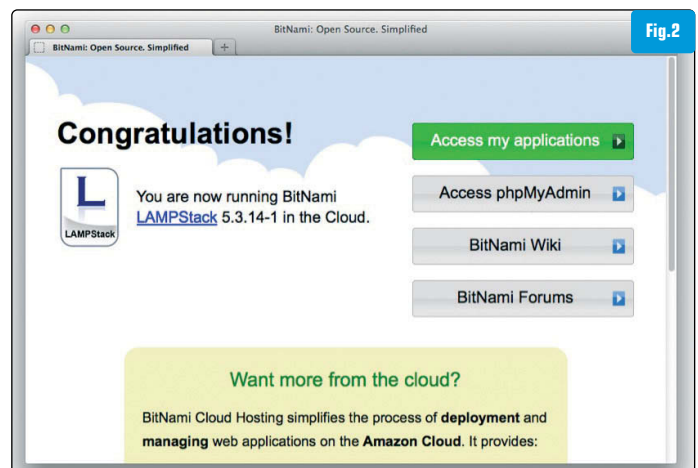
C'est bien là une des caractéristiques uniques du Cloud : vous pouvez à tout moment stopper une machine et arrêter de payer pour celle-ci, ou bien à tout moment changer la taille de vos machines – sans vous soucier de gérer la capacité de votre datacenter, c'est Amazon qui s'en charge.

C'est une propriété très appréciée pour les environnements de production comme de développement où vous avez à votre disposition des ressources jetables lors des phases d'intégration continue, de tests de non régression ou de tests en charge.

> Une mémoire d'éléphant

Pour compléter l'offre EC2, le service S3 (Simple Storage Service) permet de stocker des objets en ligne avec une facturation à l'usage (à partir de \$0,095/Go/mois). Les objets peuvent être répartis dans plusieurs « buckets », sachant que chaque bucket a une capacité illimitée tant en nombres d'objets qu'en taille totale. S3 est conçu pour une très haute durabilité de 99,999999999%. Cela signifie que les objets sont stockés de manière redondante sur plusieurs disques et dans plusieurs datacenters (zones de disponibilité) afin de les mettre à l'abri de pannes matérielles.

A noter que pour des raisons de sécurité et de protection des données, les objets déposés au sein d'une région (par exemple en Irlande) ne sont jamais déplacés ou copiés dans une autre région. Dit autrement, AWS garantit que les données stockées en Irlande ne



Connexion en SSH à l'instance LAMP

quittent jamais le sol irlandais. Seuls les clients AWS peuvent explicitement déplacer ces données de région à région – à un tarif avantageux. Nous allons maintenant exploiter une fonction particulière de S3 qui permet de l'utiliser comme serveur web statique. Par l'activation de l'option « Enable web hosting » et la configuration de règles de sécurité idoines, un bucket devient un serveur web public de contenus statiques. A noter qu'il est facile de définir finement les « mime-types » des objets ou de mettre en place des redirections HTTP. Il existe de nombreux outils pour uploader des fichiers dans S3 ou de synchroniser un répertoire local avec un bucket – par exemple l'utilitaire open-source s3cmd.

Dans l'exemple suivant nous créerons un bucket nommé 'static.mon-application.fr' dans la région Irlande.

> CloudFront : un accélérateur combinant EC2 et S3

Nous avons précédemment déployé un serveur LAMP : il est chargé de servir les requêtes HTTP dynamiques. En parallèle nous avons déposé les objets statiques (js, css, images, vidéos...) dans S3. Il ne reste plus qu'à combiner ces deux composants derrière le même nom de domaine.

Pour cela nous allons utiliser le service de CDN (Content Delivery Network) d'Amazon nommé CloudFront. Pour simplifier, il s'agit d'un réseau de proxies/caches répartis dans différents endroits du monde (il y a par exemple 2 points de présence à Paris). Ce CDN va agir comme un accélérateur en réduisant la latence pour le site web. Il est facturé uniquement à l'usage sans minimum ni engagement.

A noter que le coût du trafic entre EC2/S3 dans la région Europe et CloudFront est très réduit, ce qui rend l'utilisation de CloudFront très attractive.

Nous créons donc une distribution CloudFront en déclarant deux origines : d'une part notre instance LAMP EC2, d'autre part notre bucket S3. Au sein de CloudFront nous définissons des comportements (behaviors) permettant de router toutes les requêtes vers certaines formes d'ULR (*.jpg, *.css, *.js...) vers S3, et le reste des requêtes vers notre instance EC2 [Fig.3].

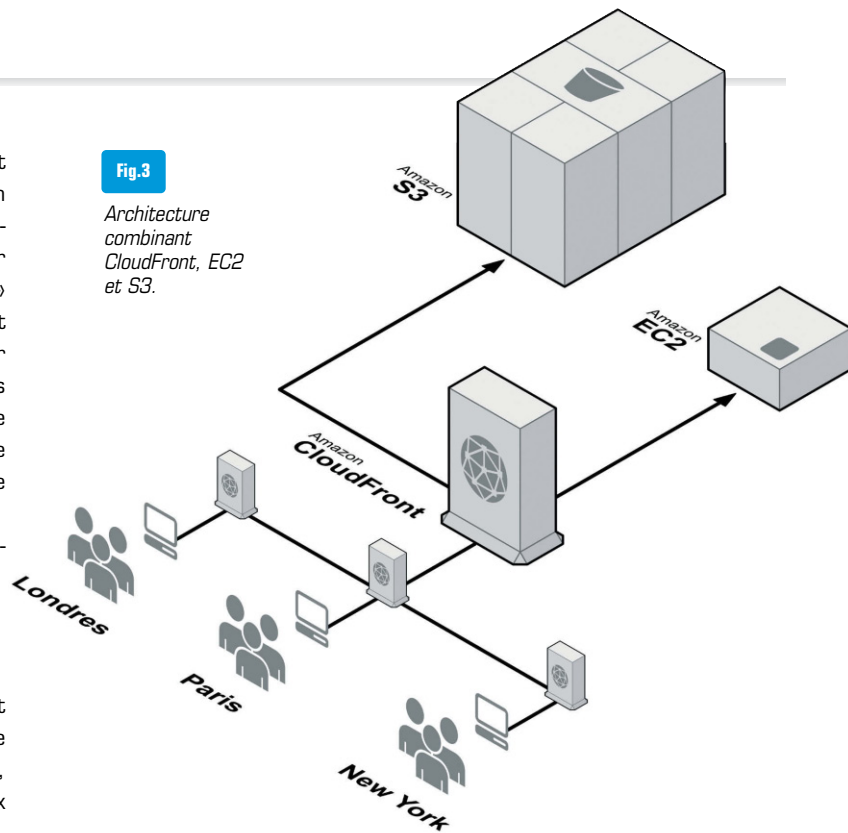
Enfin CloudFront nous propose une URL de type xxx.cloudfront.net qu'il suffira de déclarer dans son DNS cible au moyen d'un simple CNAME. Toutes les requêtes seront immédiatement accélérées par le CDN, où que se trouvent les internautes dans le monde.

> Conclusion

Nous avons fait un rapide tour d'horizon des services de base d'AWS pour les applications Web. Il existe de nombreux autres services disponibles comme des bases de données relationnelle ou NoSQL, un load-balancer, un moteur de recherche... A bientôt.

Fig.3

Architecture combinant CloudFront, EC2 et S3.



```
# -----
# connexion à EC2 sur la région Europe (Irlande)
conn = boto.ec2.connect_to_region("eu-west-1", aws_access_key
_id=AWS_ACCESS_KEY, aws_secret_access_key=AWS_SECRET_KEY)

# création d'une paire de clef SSH dans le répertoire courant
conn.create_key_pair(KEY_NAME).save(«.»)
# les fichiers .pem doivent être en mode 400 pour le client SSH
os.chmod(«./s.pem» % KEY_NAME, 0400)

# création d'un Security Group (firewall) et ouverture des ports
TCP SSH(22) et HTTP(80)
sg = conn.create_security_group(SEcurity_GROUP, «lamp: ssh et web»)
sg.authorize('tcp', 22, 22, '0.0.0.0/0')
sg.authorize('tcp', 80, 80, '0.0.0.0/0')

# démarrage de l'instance LAMP bitnami
reservation = conn.run_instances('ami-ab9490df', key_name=KEY
_NAME, security_groups=[SECURITY_GROUP,], instance_type='t1.micro')
instance = reservation.instances[0]
# attente du démarrage de l'instance
while instance.update() != 'running':
    time.sleep(5)

# réservation d'une adresse IP statique
ip = conn.allocate_address()
# association de l'adresse avec l'instance
conn.associate_address(instance.id, ip.public_ip)

print «IP = %s» % ip.public_ip
```



Stephan Hadinger
Senior Manager, Solutions Architecture chez Amazon Web Services

```
#!/usr/bin/python
# -*- coding: latin-1 -*-
import os, time, boto.ec2;

AWS_ACCESS_KEY=»XXXX»
AWS_SECRET_KEY=»YYYY»
KEY_NAME=»lamp-key»
SECURITY_GROUP=»lamp-sg»
```

Marmalade Quick : Développement rapide de jeux 2D multiplateformes

Pour ceux qui ne connaissent pas Marmalade, je vais tenter de faire un bref résumé : Marmalade est un environnement de développement spécifique permettant la réalisation d'applications et de jeux pour plusieurs plateformes (iOS, Android, Windows, etc.), tout en ayant un seul et unique code source.

Contrairement à une approche orientée web tel que PhoneGap, Marmalade se veut natif. Cependant, n'étant pas open source, il n'est pas possible de comprendre réellement son fonctionnement. Le langage principal de Marmalade est le C++, compilé et transposé pour fonctionner sur plusieurs plateformes.

Pourquoi utiliser le C++ ?

Simplement parce que Marmalade est principalement prévu pour migrer des jeux «Desktop» vers un environnement mobile. La plupart des jeux sur PC ou Mac sont développés en C++. Il est plus compliqué à maîtriser mais bien plus puissant que tout autre langage s'appuyant sur une machine virtuelle... Cet article n'a cependant pas pour but d'expliquer le développement d'applications ou de jeux avec Marmalade, mais se consacre uniquement à Marmalade Quick : un RAD(*) dédié au développement de jeu 2D.

(*) Rapid Application Development

QU'EST-CE QUE MARMALADE QUICK

C'est un environnement de développement accéléré permettant le développement de jeux 2D très rapidement, mais plus limité qu'un développement classique.

Marmalade s'appuie sur OpenQuick (ou «Quick»), qui s'appuie lui-même sur certaines technologies existantes pour faciliter le développement :

- Cocos2D-X
- Box2D
- SQLite 3
- JSON

De plus, Marmalade a décidé de proposer un autre langage pour Quick afin de permettre aux développeurs d'utiliser un langage plus simple (et plus connu des joueurs) : Lua.

> Lua : puissant, simple et léger

Lua a été développé par trois développeurs brésiliens issus du groupe de recherche **TeCGraf**, de l'université pontificale catholique de Rio de Janeiro :

- Luiz Henrique de Figueiredo
- Roberto Ierusalimsky
- Waldemar Celes

Lua est un langage de script multiparadigme : procédural, fonctionnel et orienté objet à prototype.

Principalement utilisé en tant que langage de script dans les jeux vidéo (World of Warcraft, Angry Birds, L.A. Noire, etc.), il a aussi été porté pour développer sur console de jeu : PlayStation Portable, Nintendo DS, PlayStation 2, etc.

Il est entièrement développé en ANSI C, ce qui lui permet d'être exploitable sur énormément de plateformes : Windows, Systèmes UNIX, périphérique mobile, mainframe, etc.

> Une touche d'open source associée au SDK Marmalade

OpenQuick est entièrement open source <https://github.com/marmalade/OpenQuick>. Il s'appuie sur un moteur de jeu 2D robuste et multiplateforme : Cocos2D-X

Marmalade Quick s'appuie sur OpenQuick, et apporte certaines fonctionnalités supplémentaires provenant du SDK Marmalade : Navigateur intégré, In-App multiplateforme, Facebook, Géolocalisation, connexion réseau, etc.

INSTALLATION DE MARMALADE QUICK

Pour pouvoir utiliser Marmalade Quick, il est nécessaire d'installer le SDK de Marmalade, disponible à cette adresse : <https://www.madewithmarmalade.com/downloads>.

Le SDK est disponible via plusieurs types de licences payantes, mais une licence d'évalua-

tion est disponible pendant 30 jours. Marmalade Quick est gratuit, et ne nécessite qu'une licence d'évaluation de Marmalade pour être utilisé.

Sous OS X, le SDK s'installe à la racine du disque dur de votre système : `/Developer/Marmalade/`.

Une fois installé, téléchargez le SDK de Marmalade Quick, disponible à cette adresse: <https://www.madewithmarmalade.com/marmalade-sdk/features/create-faster-marmalade-quick>.

Ce SDK est au format zip, et doit être extrait dans le dossier d'installation du SDK de Marmalade: `/Developer/Marmalade/6.2/` (Marmalade étant en version 6.2 à la date de l'écriture de l'article).

Ensuite allez dans le dossier **Applications** du SDK de Marmalade, vous devriez trouver une application du nom de **quickLaunchPad**. Si ce n'est pas le cas, recommencez l'opération décrite ci-dessus [Fig.1].

L'application **quickLaunchPad** permet la création de projet Marmalade Quick préconfiguré.

Elle permet aussi de gérer la finalité de notre projet : la publication du jeu. Elle permet de générer des exécutables compatibles avec iOS, Android, BlackBerry, etc. Pour ce faire, il suffit de cliquer sur le bouton **Publish** de l'environnement souhaité.

UN ENVIRONNEMENT DE DÉVELOPPEMENT LUA POUR MARMALADE QUICK

Plusieurs environnements de développement dédiés à Lua existent. Cependant, un se démarque particulièrement en étant multiplateforme et en proposant l'intégration de Marmalade Quick : **ZeroBrane Studio**.

Disponible à cette adresse : <http://studio.zero-brane.com> et installable sur OSX, Windows et Linux via l'utilisation des formats dédiés (.dmg, .exe, etc.).

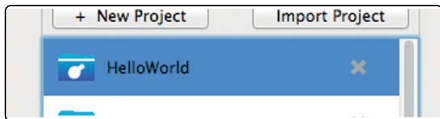
Il propose la coloration syntaxique, l'auto-complétion et le débogage à distance intégré. L'objectif des développeurs est avant tout de proposer un EDI complet et simple d'utilisation (un des développeurs a conçu l'EDI de sorte que son fils de 6 ans puisse l'utiliser facilement...) [Fig.2].

En plus d'intégrer le SDK Marmalade Quick, ZeroBrane Studio intègre aussi différents SDK de jeux : Löve 2D, Gideros, Moai et CoronaSDK.

EXEMPLE : HELLO WORLD

Dans un premier temps, ouvrez l'application **QuickLaunchPad** du SDK Marmalade Quick, puis cliquez sur le bouton '+ New Project'. Un onglet s'ouvre nécessitant la saisie du nom du projet et le répertoire de celui-ci [Fig.3].

Dans notre cas, nous allons donc saisir «HelloWorld» dans Project Name, puis cliquer sur le bouton 'Create Project'. Le nouveau projet apparaît dans la liste, à gauche.



Si vous sélectionnez le projet nouvellement créé dans la liste, vous aurez quatre nouveaux onglets : **Settings**, **Test**, **Publish** et **Advanced**.

Ces onglets permettent de définir certains paramètres de votre application et facilitent la génération des applications correspondant aux plateformes voulues (e.g. IPA pour iOS, exe pour Windows, etc.).

Ensuite, ouvrez ZeroBrane Studio et sélectionnez le fichier **main.lua** du projet créé par **QuickLaunchPad** : **File > Open > Sélectionner main.lua**.

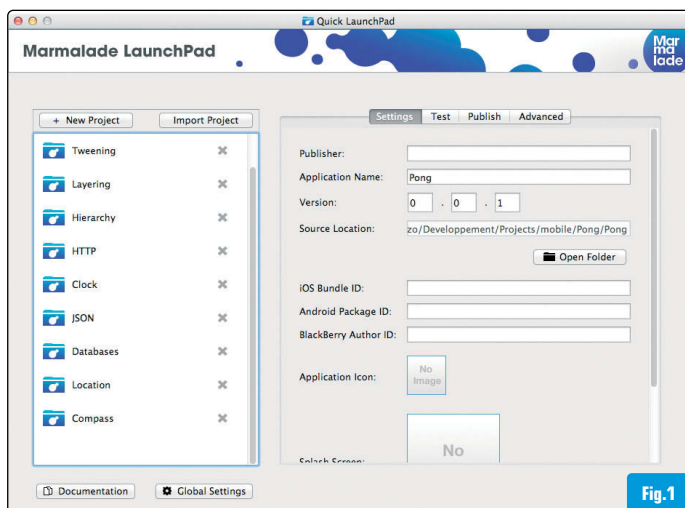
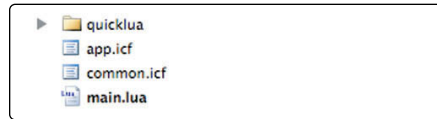


Fig.1

Le fichier s'ouvre dans l'EDI. Il faut ensuite indiquer à ZeroBrane Studio que le répertoire du projet est le même que celui du fichier courant : **Project > Project Directory > Set From Current File**.

L'onglet «Project» devrait changer pour faire apparaître la nouvelle arborescence du projet.



Le fichier **main.lua** contient déjà du code source, un «HelloWorld» *made in Marmalade* :

```
print(«This is my app!»)
```

Cliquez sur le bouton **Run** et le simulateur Marmalade va se lancer en affichant un écran noir.

Regardez bien la console de ZeroBrane Studio, et vous verrez apparaître «This is my app!». Remplacez cette chaîne de caractères par «Hello World !» et le Hello World est terminé !

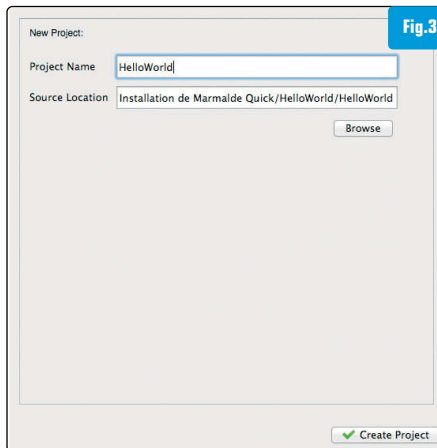


Fig.3

> Bonus : Afficher un rectangle

Dans le but de démontrer encore une fois la simplicité de Marmalade Quick, voici une ligne de code permettant d'afficher un carré blanc sur l'écran :

```
director:createRectangle(20, 20, 100, 100)
```

Cliquez sur **Run** et regardez le résultat, vous devriez voir un rectangle apparaître en bas à gauche de l'écran : [Fig.4]

Cette simplicité se retrouve également dans l'ensemble de l'API fournie par Marmalade Quick, disponible à cette adresse : <http://quick-docs.madewithmarmalade.com>

Vincent Saluzzo - Itelios
Ingénieur développement Logiciel
« Expert en développement iOS »



Fig.4

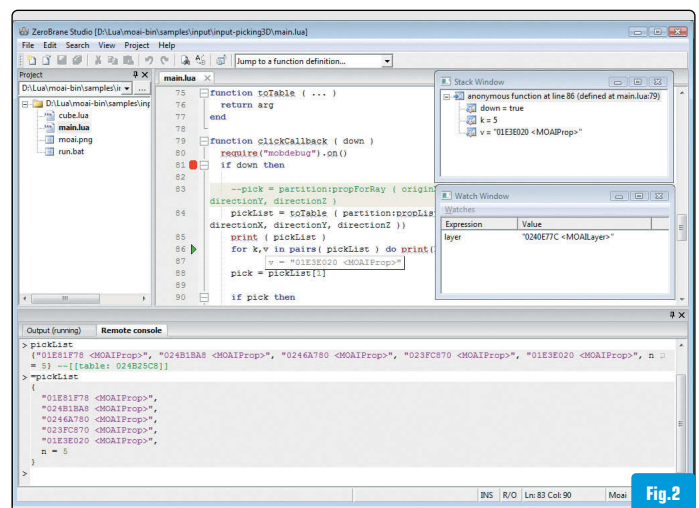


Fig.2

Débordement de tampon :

La plupart des failles de sécurité sont exploitables par une attaque dite de débordement de tampon. Comprenons ce mécanisme, afin de nous en prémunir lorsque nous écrivons du code.

Les attaques par débordement de tampon exploitent des vulnérabilités dans du code écrit a priori en C ou un autre langage natif. Un système d'exploitation écrit en C est donc potentiellement menacé. C'est le cas de Windows, bien sûr, mais aussi des autres systèmes comme Linux ou OS X. Lorsque vous faites une mise à jour pour combler une faille dont on vous dit qu'elle permettrait à un attaquant de prendre le contrôle de votre machine ou d'exécuter du code arbitraire, il y a gros à parier que cette faille peut être exploitée par une attaque par débordement de tampon, encore appelée *buffer overflow* ou *buffer overrun*. Pourquoi les failles de ce genre sont-elles aussi nombreuses ? Les causes sont a priori multiples. Nous les trouvons dans un code mal écrit par un programmeur pressé par le temps ou par son chef de projet. Ou par un programmeur peu attentif, voire inconscient du problème. Il arrive aussi que la faille soit mise en place par un programmeur malveillant, même si le cas est fort heureusement beaucoup plus rare. A travers cet article, nous allons nous efforcer de bien comprendre le fond du problème, ce qui nous aidera ensuite à être attentifs pour écrire un code beaucoup plus sûr.

1 UN PROGRAMME C (TRÈS) MAL ÉCRIT

Soit ce code C qui est un pousse-au-crime informatique :

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void TrouSecurite(const char* texte)
{
    char tampon[12];

    printf(«\nTrouSecurite - pile:\n%p\n%p\n%p\n%p\n»
    « <= adresse de retour\n\n»);

    // ce qu'il ne faut jamais faire...
    // copier une chaîne sans en vérifier la longueur.
    strcpy(tampon, texte);

    printf(«Chaine recue :\n%s\n\n», tampon);
    printf(«\nTrouSecurite - pile:\n%p\n%p\n%p\n%p\n%p\n»
    «<= adresse de retour\n\n»);
}

void Victime(void)
{
    printf(«Fonction 'Victime' executee ;-)\n»);
    exit(1);
}
```

```
int main(int argc, char* argv[])
{
    printf(«Adresse de la fonction a attaquer (Victime): %p\n»,
    Victime);

    if (argc != 2)
    {
        printf(«Il me faut un argument!\n»);
        return -1;
    }
    TrouSecurite(argv[1]);
    return 0;
}
```

Votre oeil exercé n'a pas manqué de remarquer que la fonction 'Victime' n'est jamais appelée dans le programme. Seule la fonction 'TrouSecurite' est appelée depuis *main*. Pourtant, nous allons montrer qu'en attaquant ce programme il est possible de provoquer l'exécution de la fonction Victime. Nous utiliserons pour cela un simple script Python, langage a priori inoffensif, et pourtant... Un Basic ou un Perl feraient d'ailleurs tout aussi bien l'affaire. Bien sûr Victime est elle même inoffensive dans notre exemple, mais cela peut être différent dans une application réelle. Et même si on n'invoque pas une fonction du programme il est possible de faire exécuter un code malicieux totalement externe à l'application si le cadre de pile est assez grand pour cela. Nous verrons ce que cela signifie un peu plus loin. La fonction TrouSecurite est vulnérable car la fonction C standard strcpy y est invoquée pour copier une chaîne dans un tampon nommé judicieusement 'Tampon' sans vérifier au préalable que la longueur de la chaîne n'excède pas celle du tampon. Penser que notre attaque n'est possible que parce qu'on dispose du code source et qu'on y voit en clair la faille n'est pas pertinent.

Ne jamais sous-estimer les pirates qui passent leurs journées à la recherche des failles, même sans disposer de code source. Pour ceux qui sont vraiment motivés, il est possible de détecter des failles par reverse engineering. Les pirates excellent à cet exercice, car ils sont motivés par des perspectives financières. Les hackers, lorsqu'ils participent à un concours doté de prix alléchants, font des merveilles également.

Le code d'exemple disponible sur notre site, a été compilé avec Visual Studio 2008. La démonstration reste valable avec n'importe quel compilateur (y compris gcc sous Linux), mais peut demander une petite adaptation du code Python pour fonctionner, même sous votre Visual si vous compilez vous-même. Dans tous les cas compilez le code avec le moins d'options possible. Pas d'informations de débogage (mode Release) et pas d'option d'optimisation ni de vérification

de buffer notamment. Cela correspond aux commutateurs /Od et /GS- sous Visual. Pour les besoins de la démonstration la présence de /GS- est vraiment importante sous Visual Studio, car elle désactive la vérification du cadre de pile. En dehors de cette démonstration, sa présence est en principe de bon aloi pour la raison inverse. Lorsque vous compilerez le code d'exemple sous Visual Studio, celui-ci émettra un avertissement vous mettant en garde sur le fait que l'emploi de strcpy n'est pas sûr. Le moins que l'on puisse dire est que le compilateur à raison :) L'idée, lorsque vous compilez le programme d'exemple, est d'avoir un code machine généré aussi académique que possible si l'on peut dire, et surtout facile à comprendre et à observer. Mais il serait faux de penser que la présence d'autres options rendrait automatiquement invulnérable aux attaques. Il n'en est rien. La démonstration de cet article fonctionnera en compilant à partir du projet Visual Studio 2008. Le script Python (crack.py) qui exploite la vulnérabilité est déposé dans le répertoire 'Release' du projet.

2 COMPRENDRE LE PRINCIPE DE LA PILE

Pour schématiser, un ordinateur c'est un microprocesseur entouré de mémoire. Dans le microprocesseur se trouve un certain nombre de registres qui sont eux aussi des cases mémoires dans lesquelles sont effectués des calculs et autres opérations sur les valeurs qu'elles contiennent. Le contenu des registres a souvent besoin d'être sauvegardé quelque part avant d'être récupéré et utilisé. Ce quelque part ne peut être n'importe où en mémoire, celle-ci étant en principe occupée par l'application. Ce quelque part est dans une zone attribuée au lancement de l'application par le système d'exploitation et nommée la pile. Dans le cas d'un langage compilé comme C, la pile sert aussi au passage des paramètres pour les appels de fonction ou encore de zone d'allocation des variables locales à une fonction. Le microprocesseur, et les programmes à travers lui, n'accèdent pas n'importe comment à la pile mais le font, dans le cas d'un processeur de PC, par le moyen de deux registres particuliers dits pointeurs de piles et nommés respectivement ESP et EBP pour le pointeur de pile général et pour le pointeur de pile local.

Commençons par voir comment le processeur place, on dit «pousse», une valeur sur la pile. Au départ le contenu de la pile est totalement aléatoire, mais nous supposons pour simplifier que la pile soit remplie d'octets nuls. Supposons que la zone de la pile commence à l'adresse 80000. Cette valeur est hexadécimale car nous sommes dans le domaine du code machine. Dire que la pile commence ici signifie que ESP, son pointeur, contient 80000. Représentons la pile :

```
7FFF4 00000000
7FFF8 00000000
7FFFC 00000000
80000 00000000<= ESP = 80000 pointe ici
```

Supposons que le processeur ait besoin de pousser le contenu du registre EAX sur la pile. EAX est un registre 32 bits. Supposons qu'il contienne la valeur FFFFFFFF. En assembleur l'instruction qui pousse EAX est :

```
push EAX
```

et son effet est le suivant :

```
7FFF4 00000000
7FFF8 00000000
7FFFC FFFFFFFF<= ESP = 7FFFC pointe maintenant ici
80000 00000000
```

ESP est d'abord décrémenté de la valeur 4 puisqu'on veut pousser 4 octets, puis la valeur est placée à l'endroit nouvellement pointé par ESP. Si le processeur veut récupérer le contenu précédemment poussé, il exécute l'instruction assembleur :

```
pop EAX
```

qui est l'inverse de la précédente. La valeur pointée par ESP est d'abord lue, EAX est chargé avec, puis le pointeur est incrémenté de 4 :

```
7FFF4 00000000
7FFF8 00000000
7FFFC FFFFFFFF
80000 00000000<= ESP = 80000 pointe de nouveau ici
```

Remarquons au passage que la valeur poussée reste dans la pile. Il y a des instructions assembleur qui travaillent avec la pile sans le dire. C'est le cas de l'instruction call qui sert à sauter dans un sous-programme, puis à reprendre l'exécution où il l'avait laissée après qu'une instruction ret (pour return) ait été rencontrée dans le sous-programme. Supposons qu'une instruction call 40000 se trouve à l'adresse 20000. Lorsque le processeur rencontre cette instruction, il commence par pousser, selon le processus expliqué ci-dessus, la valeur 20004 qui est l'adresse immédiatement derrière l'instruction call et qui est l'endroit où devra reprendre l'exécution, puis place 40000 dans le registre EIP, autrement dit, il saute à l'adresse 40000. Quand le ret du sous-programme est atteint le processus inverse est exécuté. L'adresse de retour, 20004 est dépilée, puis placée dans le registre EIP, donc le processeur saute à l'adresse 20004 et le programme reprend son exécution là où il en était. Et que se passe-t-il si un petit malin peut trafiquer l'adresse de retour sur la pile avant que ret ne soit rencontrée ? Le processeur sautera à cette adresse trafiquée et y exécutera le code machine qu'il y trouvera. Ce peut être l'adresse d'une fonction comme dans notre exemple, mais ce peut être aussi un programme méchant injecté dans le cadre de pile, voire à un autre endroit. Un moyen classique pour trafiquer une adresse de retour est l'attaque par débordement de tampon.

3 COMPRENDRE LE PROTOCOLE D'APPELS DES FONCTIONS C

Commençons par comprendre comment sont effectués en C les appels de fonctions, au niveau du code machine généré par le compilateur. Il existe plusieurs protocoles : cdecl, stdcall, pascall, etc. Sur le fond, le principe est le même. Nous supposons que c'est le protocole cdecl qui est utilisé dans cet article. Le mécanisme, généré par le compilateur pour chaque appel est toujours le même, pour un même protocole. Regardons d'abord de l'extérieur de la fonction appelée. Placez un point d'arrêt sur la ligne :

```
TrouSecurite(argv[1]);
```

du programme d'exemple et lancez l'exécution en mode débogage (F5 sous Visual Studio). Nous parlons bien ici de lancer le program-

me compilé en mode Release. Quand le programme s'arrête au point d'arrêt, ouvrez la fenêtre 'Disassembly' (commande Alt - 8) qui vous montrera le code assembleur du programme, tel que généré par le compilateur. Vous voyez alors : [Fig.1].

```
TrouSecurite(argv[1]);
0040109F mov     eax,dword ptr [argv]
004010A2 mov     ecx,dword ptr [eax+4]
004010A5 push    ecx
004010A6 call    TrouSecurite (401000h)
004010AB add     esp,4
```

Les deux premières lignes s'occupent de récupérer le pointeur sur la chaîne passée par la ligne de commande, autrement dit argv[1]. Ensuite ce pointeur est poussé sur la pile. Pourquoi ? Parce que c'est un paramètre de la fonction TrouSecurite et qu'en C les paramètres sont passés aux fonctions par l'intermédiaire de la pile. Puis le programme exécute le sous-programme, alias TrouSecurite via call comme expliqué plus haut. Au retour le programme restaure la pile. Pour cela il ne dépile pas la valeur poussée mais incrémente ESP de 4 (add esp,4), ce qui revient parfaitement au même, puis l'exécution se poursuit tranquillement à l'adresse 004010AE. Exécutez ce programme pas à pas (F11 sous Visual Studio) jusqu'à entrer dans TrouSecurite. vous voyez : [Fig.2].

```
void TrouSecurite(const char* texte)
{
00401000 push    ebp
00401001 mov     ebp,esp
00401003 sub     esp,0Ch
    char tampon[12];
```

Les premières instructions construisent le cadre de pile de la fonction. L'opération consiste d'abord à pousser EBP, le pointeur de pile local. En effet, EBP gère à coup sûr le cadre de pile de la fonction appelante. Il faut donc le sauvegarder. Il sera restitué à la sortie de la fonction ainsi :

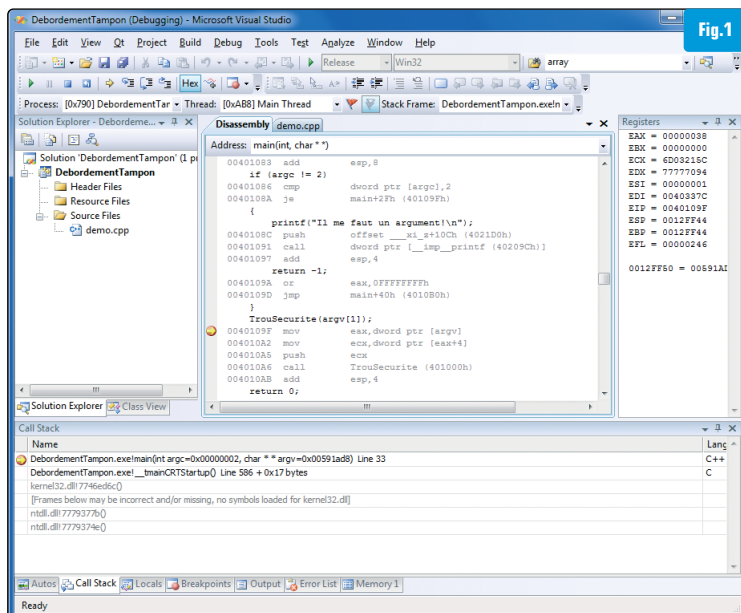
```
00401044 mov     esp,ebp
00401046 pop     ebp
00401047 ret
```

ou éventuellement :

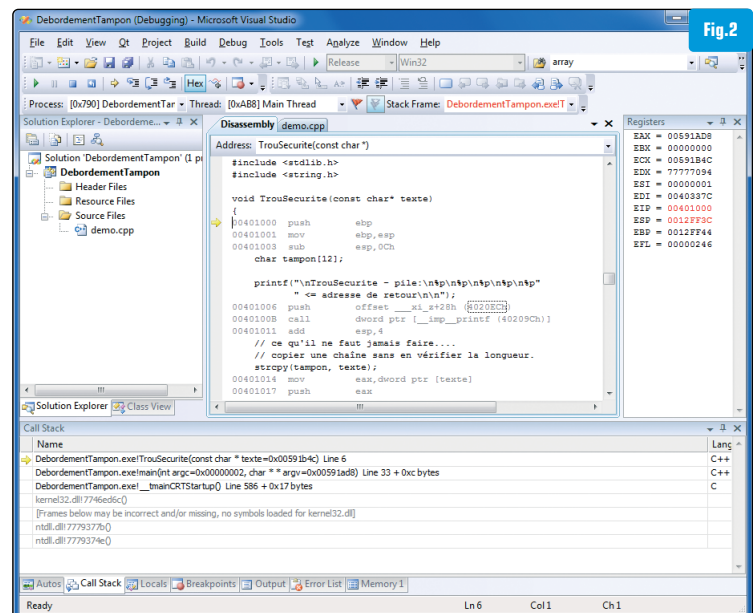
```
leave
ret
```

ce qui est la même chose. Reprenons la construction du cadre de pile. mov ebp,esp a pour effet de charger le pointeur de pile global actuel dans ebp., puis ESP est décrémenté de 0C (valeur décimale 12), autrement dit la taille du tampon destinée à recevoir la chaîne. Ce tampon est donc constitué simplement en décalant le pointeur de pile de 12 octets. Remarque: si la taille avait été 0a (valeur décimale 10), ESP aurait probablement été décrémenté de 0C malgré tout, soit une valeur multiple de 4. Maintenant EBP pointe sur le sommet de la pile, ESP sur le bas de la pile. La zone entre les deux est appelée le cadre de pile. Quand strcpy entrera en action, elle copiera la chaîne octet par octet à partir de l'adresse pointée par ESP, jusqu'à ce que le zéro de fin de chaîne soit rencontré. Si la chaîne est plus longue que 12 octets (zéro final inclus), sa copie sortira du cadre et détruira la pile. Si la copie place une nouvelle valeur au bon endroit, cette valeur sera l'adresse du point de reprise du programme à la sortie de la fonction.

Quand un cadre de pile est très grand (cas extrêmement fréquent) alors il est possible d'y placer un programme complet et on l'exécute comme expliqué. Les pires dégâts sur le système hôte sont alors à craindre... Quel est ce 'bon endroit' dont nous parlons ? Très simple. Le cadre de pile mesure 12 octets. Derrière se trouve ESP précédemment poussé. Nous en sommes donc à 16 octets. Immédiatement derrière se trouve l'adresse de retour à trafiquer. Si ici nous pouvons placer l'adresse de la fonction 'Victime', alors celle-ci sera exécutée, sans être appelée. Le programme d'exemple visualise le cadre de pile au moyen d'un usage original de printf qui est utilisée sans paramètres transmis, bien que nous lui demandions d'afficher 5 valeurs de pointeur. Dans un tel cas, printf prend les



Le mécanisme d'appel de notre fonction TrouSecurite tel que généré par le compilateur de Visual Studio.



Voici ce qui se passe au cœur de notre fonction TrouSecurite

valeurs dans la pile qu'elle considère comme ses paramètres. Les fonctions standard du C sont de véritables boîtes à surprises :-). Ensuite le programme visualise le cadre de pile après la copie de la chaîne reçue et qui est «Programmez!» (11 caractères) dans l'exemple tel que configuré dans le projet Visual Studio 20.8 donné sur notre site.

4 A L'ATTAQUE !!

Il est temps de casser notre programme. Avec ce brave Python il est possible de placer n'importe quel caractère dans une chaîne. Nous commençons par donner 16 octets qui sont là uniquement pour le remplissage du cadre de la pile attaquée. Puis nous donnons l'adresse de la fonction Victime. Comme le montre le programme d'exemple cette adresse est 00401050. Attention, les microprocesseurs PC travaillent en mode Little Endian ce qui explique que l'adresse est écrite à l'envers dans la chaîne Python. Nous poussons la malice jusqu'à laisser strcpy placer l'octet nul elle-même en tant qu'octet de fin de chaîne :-). Voici ce script Python ravageur :

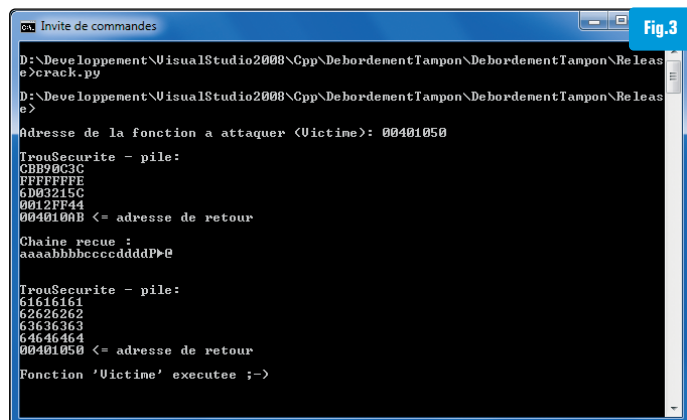
```
from os import execvp
from os.path import sep
from struct import pack

executable = 'DébordementTampon.exe'
pathexe = '.' + sep + executable

crackargs = [pathexe, 'aaaabbbbccccddd\x50\x10\x40']

execvp(crackargs[0], crackargs)
```

Comme le montre la figure, [Fig.3] Victime est bel et bien exécutée, car son adresse se retrouve comme par magie sur la pile, et c'est cette adresse dans laquelle sautera le microprocesseur, au retour de la fonction TrouSecurite.



Un programme C cassé par 4 lignes de Python :-)

5 SE PROTÉGER DES DÉBOREMENTS DE TAMPON

Ayant bien pris la mesure du problème, voyons comment nous prémunir des attaques par débordements tampon. Avant tout, vaccinons-nous contre le mauvais réflexe qui est de dire que si une zone de pile est trop petite et qu'elle risque d'être débordée, alors, il n'y a qu'à l'agrandir. Ainsi, dans TrouSecurite nous mettrions :

```
char tampon[2000];
```

Cette idée est complètement idiote. Mais ne rigolez pas !, on la voit de temps en temps de la part d'un programmeur pressé par le temps et qui a la tête dans le guidon. C'est vraiment stupide, car le problème du débordement de tampon est non seulement toujours là, mais nous avons tellement agrandi le cadre de pile que nous avons maintenant la place pour y mettre un programme complet. Bref, en agrandissant le tampon, nous avons considérablement agrandi la faille :-). Un programmeur rigoureux veillera à tester la longueur d'un argument de fonction devant être copié dans la pile **avant** l'opération de copie. On peut aussi adopter la politique de ne copier qu'une certaine longueur, par exemple avec la fonction `strncpy`. Ce qui présente l'inconvénient de tronquer éventuellement l'argument. (Gare au zéro terminal des chaînes dans ce cas !) Dans l'idéal, le mieux est d'allouer dynamiquement une zone mémoire en fonction de la taille de l'argument.

A ce titre, le premier réflexe devrait être de ne pas réinventer la roue. Le meilleur code est toujours celui que l'on n'écrit pas. On utilisera plutôt des bibliothèques qui ont fait leurs preuves. Là, même si vous adorez C, un peu de C++ avec sa bibliothèque standard et sa classe `string` sera sans doute bienvenu. Ensuite, ne faites jamais confiance aux développeurs de votre équipe. Ce n'est pas parce que votre code doit être appelé par une fonction que développe votre voisin de bureau qu'il n'y a aucun danger.

Au contraire, il convient d'être deux fois plus rigoureux. Votre voisin de bureau ne sait pas comment vous implémentez votre fonction, et il ne veut pas le savoir. Il a suffisamment à faire avec son propre code. Il invoque votre code peut-être n'importe comment, mais vous ne pouvez pas lui en vouloir.

C'est à vous d'écrire du code solide. Enfin, ne vous faites pas confiance. Un programmeur teste volontiers son code dans le cadre d'une utilisation normale, mais oublie fréquemment de le faire dans le cadre d'une utilisation débile. C'est un tort. Une dernière chose. Si vous ne pouvez pas faire confiance à votre voisin pour appeler votre code, vous pouvez lui faire confiance pour y voir immédiatement une faille s'il y jette un oeil. Les programmeurs ont l'esprit critique, c'est bien connu !

Prêtez-vous de bonne grâce au jeu parfois désagréable de la revue de code. Faites relire votre code, relisez celui des autres.

Frédéric Mazué - fmazue@programmez.com

L'information permanente

- 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

IntelliJ IDEA 12 : bug d'applications Web

L'Ultimate Edition de l'IDE dispose de beaucoup d'assistants pour développer pour le Web, d'un support pour des frameworks et de la majorité des serveurs disponibles sur le marché. J'ai trouvé un problème pour tester des applications Web sur des serveurs locaux (bug?). Ce problème existe, si on essaye des Servlets, ou le support JSF2, sur Tomcat, Google App Engine dev serveur, etc.

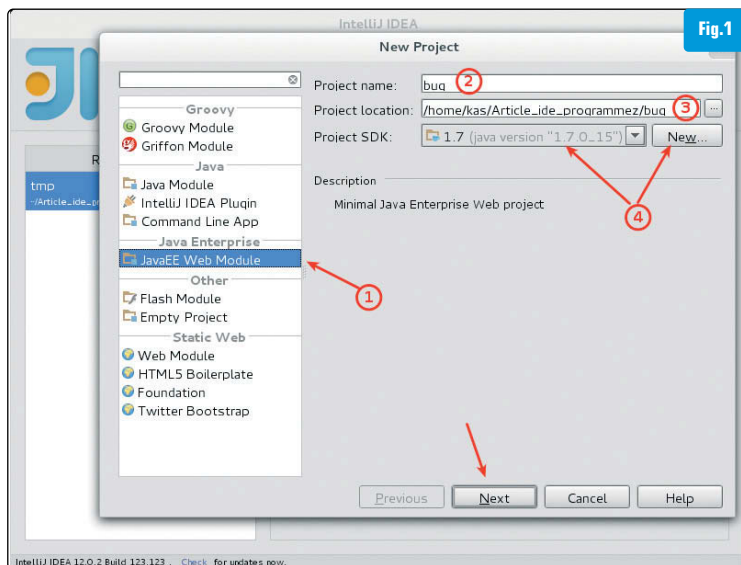
> Configuration

IntelliJ IDEA 12 • Java SE 7u15 (JDK) • Tomcat 7.0 • Fedora 17 (Linux 64 bits)

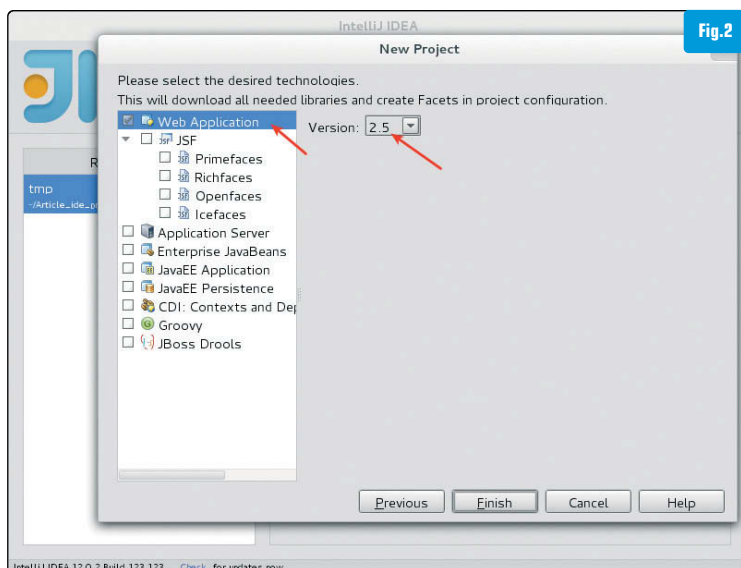
> Contexte

On commence par créer une application Web :

Create New Project dans l'écran d'accueil (ou File > New Project) Sélectionnez **JavaEE Web Module** (1). Donnez un nom au projet (2). Sélectionnez un dossier (3) et le JDK utilisé en cliquant sur **New...** (4). Continuez en cliquant sur **Next**. [Fig.1]. Cochez la case **Web Application** et sélectionnez sa version (par exemple 2.5) [Fig.2].



Java EE Web module



Cochez Web Application

Cochez aussi **Application Server** et sélectionnez **Tomcat 7.0** (Si c'est la première application utilisant Tomcat 7, il faut sélectionner le dossier d'installation du serveur). Maintenant, cliquez sur **Finish** pour créer le projet. *Ajoutez un petit Servlet :*

```
package com.java_javafx.ka;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;

/**
 * Created with IntelliJ IDEA.
 * User: ka
 * Date: 3/1/13
 * Time: 3:08 PM
 * To change this template use File | Settings | File Templates.
 */
public class Test extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();
        out.println("test bug serveurs");
    }
}
```

> Mappez le servlet dans web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
         http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
         version="2.5">

    <servlet>
        <servlet-name>Test</servlet-name>
        <servlet-class>com.java_javafx.ka.Test</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>Test</servlet-name>
        <url-pattern>/test</url-pattern>
    </servlet-mapping>
</web-app>
```


> Configurez l'application

Avant de lancer l'application il faut passer par le menu **Run > Edit Configurations**, pour vérifier et modifier si besoin la configuration pour l'application sur le serveur [Fig.3]. On note de suite un avertissement «pas d'artefact pour le déploiement» et on vous propose de solutionner le problème en cliquant sur le bouton «Fix». **Ne cliquez surtout pas**. Ceci ne résout pas le problème je vous le confirme. Vérifiez malgré tout la configuration du serveur. Cliquez ensuite sur l'onglet Deployment [Fig.4]. Dans l'onglet Deployment, on nous affiche qu'il y a rien à déployer. Cliquez sur Cancel pour quitter la fenêtre.

> Solution

De toute façon, si vous essayez de lancer l'application comme telle, vous n'aurez rien dans le navigateur. Il est aussi possible que l'IDE vous affiche des messages d'erreurs bizarres sur la console. Il faut pour solutionner le problème, créer un Artefact attaché au projet (module) en cours. Dans le menu **File > sélectionnez Project Structure** [Fig.5]. Sélectionnez **Artifacts(1)**. L'artefact existant n'est

pas bon, c'est un artefact générique(2). Il nous faut un Artefact correspondant au module. Pour cette raison on commence par supprimer l'artefact en cliquant sur le signe «-» (3), [Fig.6]. Après avoir supprimé l'artefact créé par l'IDE, nous allons créer un nouvel artefact. Un clic sur le signe «+». Sélectionnez **Web Application : Exploded > sélectionnez From Modules**.

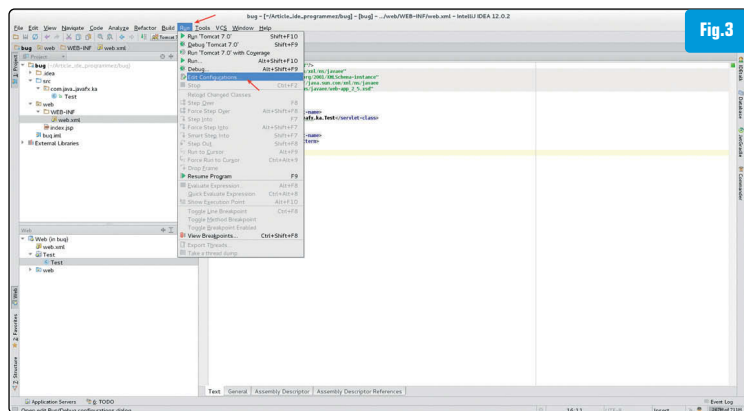
Le nom du projet est sélectionné > un clic sur **OK**. L'artefact est ajouté > un clic sur **OK** [Fig.7]. Pour valider l'artefact, on retourne dans le menu **Run > Edit Configurations**.

Sélectionnez l'onglet. **Deployment**.

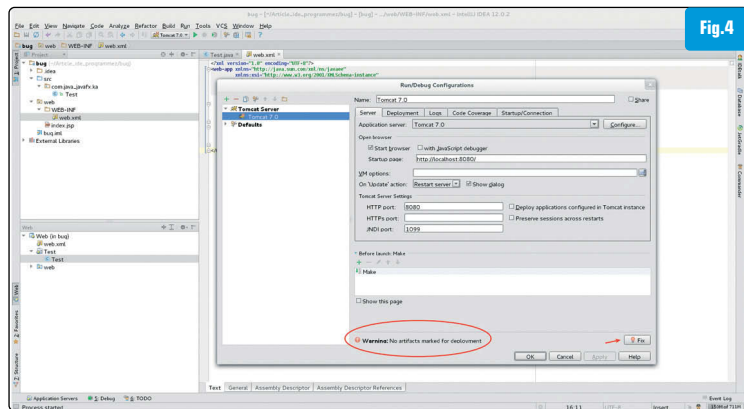
Cliquez maintenant sur le bouton «Fix» [Fig.8]. L'artefact est ajouté pour le déploiement. Pour lancer l'application le menu **Run > Run** ou un clic sur le bouton «Play». Maintenant le servlet est lancé et le serveur marche sans problème.

Ouvrez un navigateur d'internet (par exemple Firefox) et tapez l'URL suivant : <http://localhost:8080/test>.

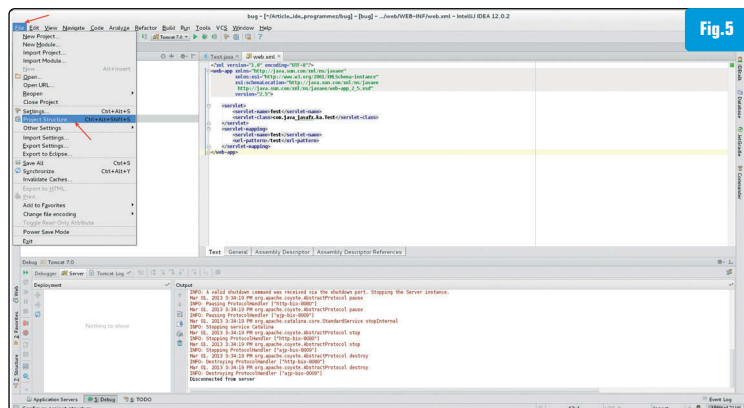
Kaesar Alnires - Développeur Java - Leader jug-Cergy
<http://www.java-javafx.com>



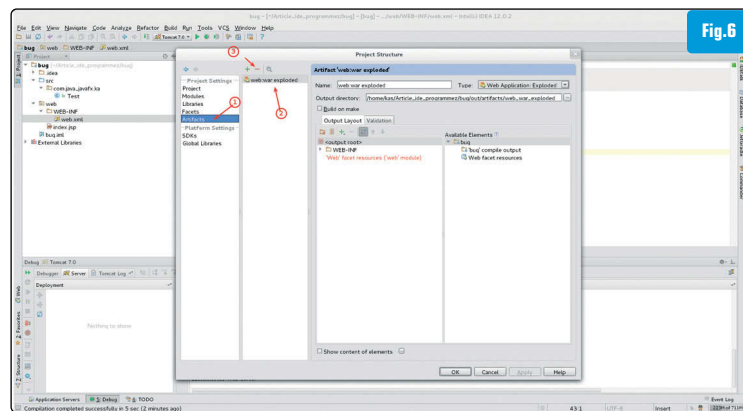
Le menu Edit Configurations



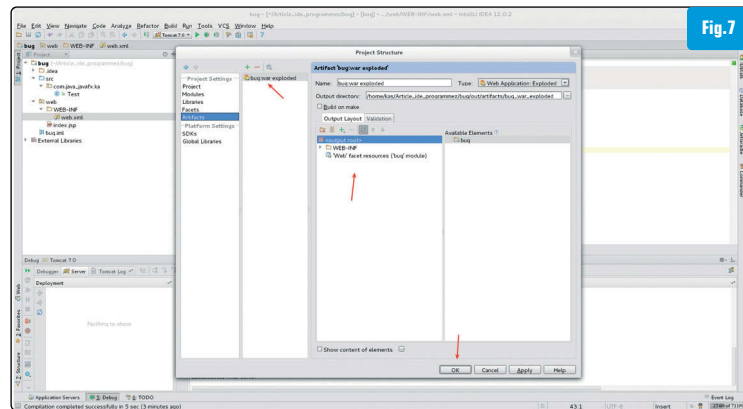
Un message d'avertissement



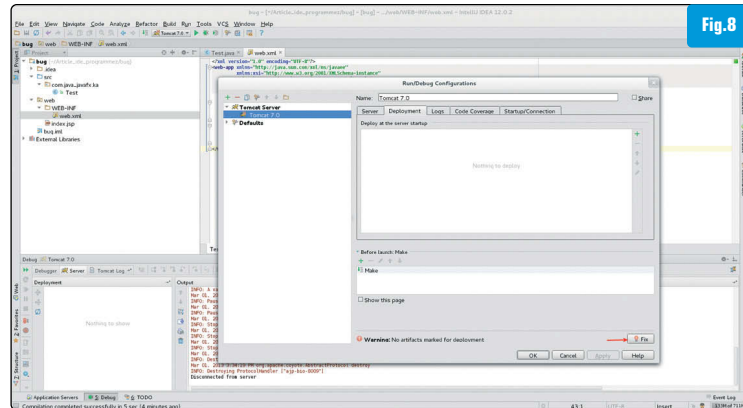
Project Structure



Supprimer l'artefact qui existe



Détails de l'artefact ajouté



Cliquer sur le bouton Fix

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.*



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 Boulangers 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

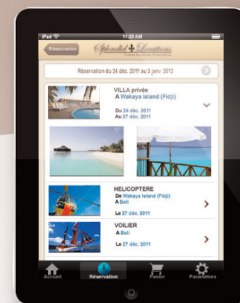
Adresse mail

☐ Je joins mon règlement par chèque à l'ordre de SOLUTIONS LOGICIELS ☐ Je souhaite régler à réception de facture

DÉVELOPPEZ 10 FOIS PLUS VITE

WINDEV®

ATELIER DE
GÉNIE LOGICIEL
PROFESSIONNEL



DÉVELOPPEMENT D'APPLI POUR TABLETTE*: AVEC WINDEV MOBILE 18 C'EST UN JEU D'ENFANT

* IPAD, ANDROID, WINDOWS 8 & RT



VOUS AVEZ UN EXISTANT WINDEV ? IL EST PORTABLE DIRECTEMENT. RETAILLEZ LES FENÊTRES, RECOMPILÉZ.

VOUS N'AVEZ PAS D'EXISTANT? DÉVELOPPEZ TRÈS TRÈS VITE VOS APPLICATIONS MOBILES AVEC WINDEV MOBILE 18, EN GÉNÉRAL QUELQUES JOURS SUFFISENT .

BASE DE DONNÉES SQL INCLUSE



« WINDEV 18 »

Atelier de développement
multi-plateformes: Windows 8,
RT, 7, Vista, XP..., Linux, Internet,
Android, iPhone, iPad...

WINDEV Mobile 18 compile les applications mobiles.
WINDEV 18 compile les applications Windows, Mac et Linux.
WEBDEV 18 crée les sites Internet et Intranet.

Votre code source est unique

**DEMANDEZ VOTRE
DOSSIER GRATUIT**

260 pages - 100 témoignages - DVD
Tél: 04.67.032.032 info@pcsoft.fr

www.pcsoft.fr

Fournisseur Officiel de la
Préparation Olympique



VERSION
EXPRESS
GRATUITE
Téléchargez-la !