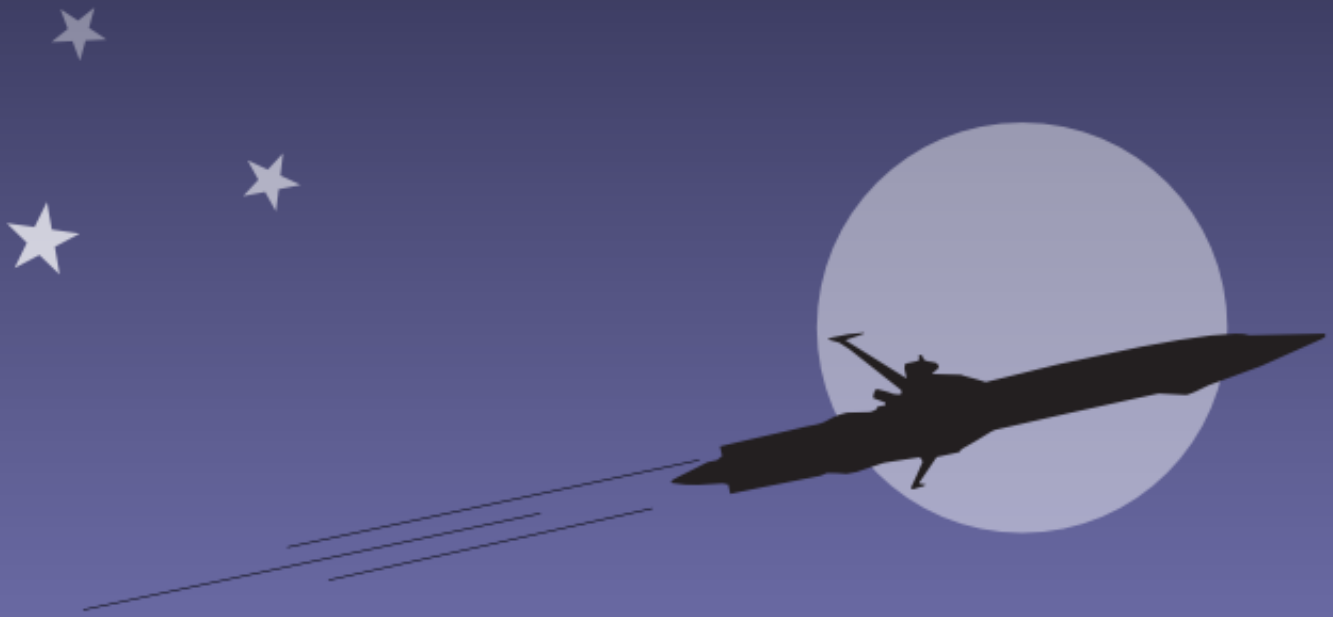
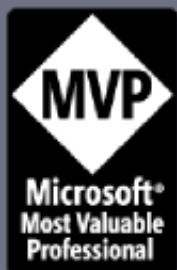


Tome 9

Stratégies & Avenir



Olivier Dahan



Collection
ALL DOT BLOG

(c)2014 Olivier Dahan / e-naxos

e-n@Xos

Formation – Audit – Conseil – Développement
XAML (Windows Store, WPF, Silverlight, Windows Phone), C#
Cross-plateforme Windows / Android / iOS
UX Design



ALL DOT.BLOG

Tome 9

Stratégies et Avenir

Tout Dot.Blog par thème sous la forme de livres PDF gratuits !
Reproduction, utilisation et diffusion interdites sans l'autorisation de l'auteur



Olivier Dahan
odahan@gmail.com

Table des matières

Présentation du Tome 9 « Stratégies & Avenir »	6
Se préparer au "big shift" de la stratégie Microsoft	8
Our Strategy with Silverlight has shifted	8
Back to 80's.....	8
Le tout-PC-Microsoft	9
L'heure du basculement technologique	10
Le fantôme de l'"Everywhere"	10
La gloire de l'horizontalité	12
Jobs l'assassin	12
De l'horizontalité à la verticalité : le grand Shift	14
Realpolitik !.....	15
Mais la verticalité alors c'est quoi dans tout ça ?	17
Le cross-plateforme à la sauce Windows	17
(Re) back to the 80's.....	18
Trois blocs, trois visions du monde	19
Windows 8 : Avantage cohérence ?.....	19
Conclusion	20
HTML 5 : la tragique fin d'un buzz.....	20
Le Web est-il encore l'avenir de l'Homme ?.....	21
Internet for ever.....	21
L'explosion des form factors coulent le Web.....	22
Un monde natif.....	23
La mort du Web ?.....	23
Et alors ? Et Alors ?.....	24
Conclusion	25
L'avenir proche du développement : quels environnements pour quels produits ?	26
Plusieurs approches.....	27
Le pari de l'impasse	27
Ma vision des choses	28

6 solutions possibles.....	32
L'investissement.....	32
Le tooling	33
Découpage des solutions.....	33
And the winner is.....	35
ROI : comment le prendre en compte ?.....	35
Conclusion	36
Le Web 3.0 = Pas de Web ?.....	37
Le natif remplace le Web.....	38
Internet != Web	38
HTML 5, le buzz qui fait flop	38
Un avenir segmenté	39
La mort d'Apple ?	39
La mort de Google et Facebook ?.....	40
Le Web 3.0 ne sera pas, HTML 5 non plus donc.....	41
Conclusion	41
Mobiles : les apps natives flinguent le web !.....	42
HTML5 La norme qui n'existera jamais (Officiel) !.....	44
Des raisons techniques	44
L'éclatement et la mort de la norme, c'est maintenant !.....	44
WHATWG ? Kesako ?	44
Pourquoi le divorce ?	45
BOOMMM !.....	45
Le tour de la presse	45
Conclusion	46
Silverlight 5 : Hommage à un oublié.....	46
9 Décembre 2011	47
Une histoire longue qui n'est pas terminée	47
Conclusion	48

Ne jetez pas WinRT par les fenêtres : offrez-lui en ! (la solution pour un WinRT adapté au PC).....	50
Rendez le "S" à Windows !.....	51
Les perspectives de Windows 8.1.....	52
Le menu démarrer, un problème qui se règle.....	52
Régler le problème du double bureau et du fullscreen WinRT.....	53
Conclusion.....	56
Microsoft annonce une refonte de Windows 8 historique.....	57
Une refonte, un "rework" dans le texte.....	57
Un virage à 180° ?.....	57
Quel contenu, quelle ampleur ?.....	58
Une annonce par la bande.....	58
Conclusion.....	59
WPF : Et si on en reparlait sérieusement !.....	59
WPF L'éternel oublié mais l'inoubliable aussi.....	59
Windows un socle fissuré ?.....	61
Fissuré mais majoritaire pour longtemps !.....	61
Windows va durer, mais quel Windows ?.....	62
L'avenir : WPF, en toute logique.....	63
Passéisme, WinRT-bashing ou réalisme ?.....	63
Conclusion.....	64
WPF : Retour sur une véritable avancée.....	65
UX et UI.....	65
Des exemples.....	66
Conclusion.....	75
BYOD, CYOD, MAM, MDM, MIM vive les unités mobiles !.....	76
Introduction.....	76
Les points clés d'une intégration en entreprise.....	77
Conclusion.....	85
Avertissements.....	87

E-Naxos87

Présentation du Tome 9 « Stratégies & Avenir »

Ce tome 9 est un recueil de billets publiés ces dernières années. Sa spécificité est d'aborder des sujets qui ne sont pas directement techniques comme les tomes précédents.

Ici il ne s'agit pas de comprendre telle ou telle autre option d'un langage ou d'une plateforme mais de saisir les tendances du marché, les raisons de certaines situations mais aussi de prévoir comment ces tendances vont évoluer et comment s'y préparer pour mieux gérer l'avenir.

Comprendre le « big shift » de Microsoft, mettre en place une stratégie d'investissement sous telle ou telle autre plateforme mobile, saisir les enjeux, éliminer les fausses solutions sont tout aussi essentiel que de connaître et bien utiliser `async/await` de C# ...

A quoi bon disposer d'outils de développement puissants, de plateformes techniques rutilantes si on s'est trompé de cheval en misant sur le tocard qui arrivera bon dernier ? Peu importe la qualité intrinsèque d'une plateforme, elle peut s'avérer être un mauvais choix lorsqu'il s'agit de rentabiliser son apprentissage. Nous ne travaillons pas pour la gloire, nos choix conditionnent notre qualité de vie, celle de notre famille. On dépasse ici les débats purement techniques.

Et ces réflexions concernent aussi bien le développeur (sur quel outil se former pour ne pas être largué l'année prochaine ?) que le DSI (quelle plateforme choisir ou éviter pour conserver mon poste...) ou le dirigeant (minimiser mes investissements et maximiser mes gains).

A la fois spéculatif et basement matérialiste, « Stratégies et Avenir » regroupe tous ces articles qui ont traité d'une façon ou d'une autre ce vaste problème qu'est de choisir les bonnes directions et d'éviter autant que faire se peut les mauvaises.

Mais les articles publiés ici sont parfois anciens.

Quel intérêt de les publier aujourd'hui dans ce recueil ?

Il y en a plusieurs.

Le premier est de regrouper en un seul endroit l'ensemble de ces textes qui restent pertinents. Synthétiser, rendre cohérent des textes épars, c'est aussi leur redonner une cohérence, offrir un nouveau niveau de lecture. Tous ensemble, liés en un seul livre, ils se donnent sens les uns les autres dans un jeu de ping-pong où l'écho de chaque billet peut s'entendre dans tous les autres.

Le second intérêt consiste en cette confrontation toujours risquée entre prédiction et réalité avérée. Oser prendre le risque de mettre en perspective ses visions d'avenir à l'aune de la réalité effective, du passé est un moyen de démontrer un autre type d'expertise qui dépasse la simple technique. Mon métier est aussi celui de Conseil,

vérifier que mes visions étaient justes m'apparait aussi naturel que de goûter la spécialité d'un pâtissier, le seul moyen de savoir si je reviendrais dans sa boulangerie ou n'y mettrais plus les pieds... Ce qui est naturel pour un pâtissier doit l'être aussi pour un expert en IT.

Enfin, l'intérêt ultime, celui qui donne un vrai sens pratique pour le lecteur à cette publication c'est de pouvoir revisiter simplement dans un livre facile à consulter tout un pan de ce qui constitue malgré tout l'histoire récente de notre profession. Car c'est par la connaissance de l'histoire et les enseignements qu'on en tire qu'on peut avoir un pouvoir réel sur l'avenir.

Ainsi, beaucoup de choses exprimées dans ces billets permettent à la fois de mieux comprendre pourquoi le marché est ce qu'il est aujourd'hui et comment s'en sortir au mieux. Nombre de conseils sont, à mon sens, toujours valables.

Dès lors, publier ce recueil s'est imposé comme un complément intéressant aux 8 autres tomes beaucoup plus techniques déjà parus. A lui seul il n'aurait pas forcément autant de valeur, mais dans le sens où il résume l'histoire de ce qui nous a mené à l'état actuel du marché il se place un peu comme une sorte de « making-of » de mes autres ouvrages techniques auxquels il donne un sens, une justification.

Ce tome 9 est aussi le seul de toute la série ALL.DOT.BLOG à ne pas avoir été réécrit. Ce serait tricher que de « mettre à jour » des visions exprimées dans le passé. Prévoir l'avenir après coup est forcément une escroquerie. Ici les textes originaux ont été respectés, sauf coquilles ou autres corrections mineures qui s'imposaient.

De même, à la différence des autres tomes où les billets devenus chapitres ne sont pas datés car ce qu'ils véhiculent est intemporel tant que les techniques dont ils parlent existent, j'ai souhaité ici indiquer en sous-titre de chaque chapitre sa date de parution accompagnée d'une brève introduction. Certaines analyses ont d'autant plus de poids lorsqu'on sait que je les ai écrites plusieurs années avant que les choses ne deviennent évidentes à tout un chacun...

Ce tome est titré « Stratégies & Avenir ». C'est-à-dire qu'il s'appuie sur l'entrelacement ambigu entre stratégies (comment agir en fonction d'une réalité avérée) et avenir (prédiction du marché, anticiper ses choix) qui est tel que forcément ces deux aspects se trouvent regroupés en un seul tome.

Je vous souhaite une bonne lecture et surtout du plaisir. Celui de revisiter le passé proche pour mieux comprendre le présent et le futur, de réfléchir aux alternatives qui se posaient hier, vérifier leur à propos, tout cela dans le but de mieux gérer encore le futur qui inlassablement se tient toujours devant nous, fuyant inexorablement d'un pas quand nous en faisons un vers lui... Condamnés à courir après cet objectif fuyant, bien connaître ce qui a créé le présent est pour nous tous une arme de grande valeur pour affronter demain et ses doutes...

Se préparer au “big shift” de la stratégie Microsoft

Parution : 30 novembre 2011.

Écrit il y a plus de deux ans, cet article permet de comprendre l'état actuel du marché, l'avènement des applications natives, des smartphones et des tablettes, la mort de Silverlight et surtout pourquoi Microsoft a choisi la stratégie Windows 8 / WinRT. L'anticipation de ce qui allait se passer éclaire toujours le présent et permet de se projeter dans l'avenir pour faire des choix judicieux... Un article que j'écrirais autrement peut-être aujourd'hui mais qui dirait sur le fond les mêmes choses. Il éclaire d'une façon limpide ce qui reste pour beaucoup incompréhensible...

Tout le monde se souvient de cette petite phrase lâchée par Bob Muglia avant l'été (2011) “*Our strategy with Silverlight has shifted*”, ce qu'on traduirait par “notre stratégie à propos de Silverlight s'est déplacée”. Clair et nébuleux à la fois. La panique a envahi depuis le monde Silverlight, malgré la V5 qui va sortir très prochainement, la question devient “y-aura-t-il une V6 ?”. Mais je crois sincèrement que les vraies questions sont ailleurs. Microsoft a “shifté” sa stratégie, globalement, pas seulement autour de SL. Quelques éléments de réflexion pour mieux comprendre l'avenir.

Our Strategy with Silverlight has shifted

Bien entendu cela semble bien ciblé, c'est Silverlight dont il est question et de rien d'autre. *Mais est-ce si clair ?* D'ailleurs Bob Muglia a quitté Microsoft depuis cette annonce. Non pas que je crois que les deux événements soient liés aussi directement, mais tout simplement parce que Microsoft a “shifté” *toute sa stratégie*. Muglia a été poussé vers la sortie au profit de Steven Sinofsky. Le titre exact de leur poste est différent, mais leur rôle de leader a bien été échangé. Et ce ne sont pas les seuls modifications, d'autres jeux de chaises musicales ont eu lieu chez Microsoft.

Tout ce remue-ménage ne peut avoir pour seul point de départ et seul centre d'intérêt Silverlight, même si nous avons tous, nous les amoureux de SL, la tentation de tout ramener à ce fantastique plugin.

Non, *c'est un changement de stratégie en profondeur*. Décodons-le.

Back to 80's

Dans les années 80, les développeurs et éditeurs de logiciels se sont trouvés dans une situation très embarrassante : Quelle plateforme choisir pour développer un nouveau logiciel ? D'un côté nous avons Apple, tout auréolé de sa capacité à produire des machines comme l'Apple II ou le Mac, et de l'autre l'arrivée fracassante de l'IBM PC. Faire confiance à ceux qui savaient produire des micros depuis un moment ou faire confiance au poids gigantesque de l'image de marque IBM qui représentait tout le sérieux de la grosse informatique à cette époque.

Pour beaucoup de lecteurs il s'agit d'une époque tellement lointaine que certains n'étaient peut-être pas encore nés. Mais je vous assure que pour un éditeur de logiciel faire le choix Apple ou IBM à cette époque a été un vrai cauchemar. Choisir l'un et risquer de tout perdre si on se trompait, choisir les deux et doubler les coûts de développement et ne plus avoir assez de ressources pour lancer le produit...

Ne plaisantez pas sur ce passage d'ancien combattant... vous allez y être plongé de gré ou de force et d'une façon encore pire que vos aînés !

Mais pour comprendre suivons le fil de l'histoire.

Le tout-PC-Microsoft

Ce cauchemar a été certainement l'un des moteurs de la grande bascule qui a eu lieu, le marché, notamment des entreprises, basculant vers le PC, chaque bastion gagné faisant de plus en plus pencher la balance en faveur de cette plateforme et entraînant encore plus vite, par effet boule de neige, son adoption par d'autres entreprises dans le monde.

Ce fut l'avènement des PC. Et donc celui de MSDOS, suivi après par Windows.

Heureux les informaticiens ayant connu cette glorieuse époque ! 100% du marché de la micro-informatique était fait de PC, et 95% de ces PC marchaient sous MSDOS puis Windows.

Le cauchemar avait enfin pris fin.

Plus de question à se poser concernant le choix de la plateforme pour développer. C'était pour des PC, et donc pour de l'OS Microsoft.

Époque bénie entre toutes où nous avons préféré être captifs d'un seul éditeur, d'une seule architecture, car en retour nous avons gagné la stabilité, l'abaissement des risques lié au choix de la plateforme, l'abaissement des coûts de production (une seule version d'un soft au lieu de plusieurs pour plusieurs OS).

Les progrès immenses de la micro-informatique des années 90 et 2000 n'a qu'un seul moteur principal : l'uniformisation et la garantie de stabilité du parc mondial lié à la domination de Microsoft et de la plateforme hardware PC.

Il ne faudra jamais oublier ce point essentiel. **Car cet âge d'or est aujourd'hui révolu** par la volonté d'un fou haineux. Vous comprendrez plus loin...

Comme le prouvait dernièrement une enquête, pour la majorité des gens, c'est Microsoft qui a révolutionné la micro-informatique et non pas Apple, car Microsoft a permis la banalisation de l'outil informatique. Et le fait que tout le monde puisse avoir

accès à cette technologie compte bien plus que la poignée de bobos qui pouvaient se payer un Mac pour décorer leur bureau (médecins, gosses de riches...).

La domination de Microsoft et de la plateforme PC a été l'origine de l'explosion de la micro dans le monde.

Il y a bien eu la naissance d'un sentiment anti "M\$" comme on l'écrivait. Le sentiment d'être prisonnier de la volonté d'un seul éditeur. Les procès en position dominante, et le jeu, il est vrai, parfois trouble de Microsoft n'hésitant pas à utiliser des pratiques assez peu "friendly" avec la concurrence. Mais après tout, le modèle libéral américain c'est ça : écraser le plus de concurrents pour devenir le meilleur. C'est une guerre.

Sauf que ce rêve du libéralisme n'est qu'un leurre : lorsqu'il y a un vrai bon élève qui applique correctement toutes les ficelles de ce jeu sans règle, on lui tape dessus. Preuve que le libéralisme ne peut fonctionner que si le rêve ne reste qu'un rêve et qu'aucune entreprise ne gagne la partie... C'est à cette époque que j'ai pris conscience que le libéralisme était une idéologie aussi morte que l'était celle du communisme à la soviétique. Mais c'est une autre histoire !

L'heure du basculement technologique

Mais tout à une fin.

Si Microsoft a su rester leader sur la plateforme PC, c'est la plateforme PC qui ne représente plus 100% de la micro-informatique ! Cruel coup du sort...

Microsoft est devenu le roi d'un royaume qui se rétrécit.

Non pas qu'il se vende moins de PC dans le monde, il s'en vend toujours plus chaque année. Non, mais l'émergence des Smartphones et des tablettes fait, qu'en nombre d'unités en circulation un rééquilibrage s'effectue. Car ces petites machines sont de la micro-informatique avec des OS et des applications. On a dépassé le cadre des téléphones avec petits écrans mais totalement fermés et propriétaires.

Sans tenter de faire dans la divination, on peut admettre que les 100% de la micro-informatique, espace hier occupé entièrement par les PC, va, au fil du temps tendre vers un partitionnement de type 1/3. Un tiers de PC, un tiers de Smartphones, un tiers de tablettes. Le prix de ces deux derniers étant largement inférieur à celui d'une grosse machine de bureau, on peut même s'attendre à ce que, toujours en nombre d'unités, tablettes et Smartphones réunies dépassent les 66%. Le marché du PC classique (bureau, portable) restant toujours aussi énorme en unités vendues, mais ne comptant plus que pour une 20aine de pourcent dans le marché global de la micro-informatique.

Microsoft est toujours le roi de son marché et de sa plateforme, c'est cette dernière qui ne représentera plus à terme que 20% du marché !

Le fantasme de l'"Everywhere"

Bien que devenu totalement marginal durant ces 20 années de domination des PC, le Mac d'Apple a su conservé un marché de niche, entretenu de façon un peu fanatique par le culte de la marque, le culte de la personne comme celle de Jobs.

Et bien que tous les développeurs PC aient bénéficié de la domination de Microsoft pour payer leur loyer et remplir leur écuelle de soupe, beaucoup ont conservé comme une espèce de blessure narcissique, fantasmant toujours sur la plateforme Mac.

De là est né le fantasme de l'“Everywhere”.

Les informaticiens gardaient ce rêve irrationnel d'un moyen d'écrire un seul logiciel qui marcherait aussi bien sur PC que sur Mac, même si ce dernier ne représentait plus rien (2% environ) du marché. Les restes de cette blessures d'avoir été obligé de choisir les PC et leur DOS très moche alors que leur cœur les poussaient à vénérer le Mac ? Très certainement.

Dans le même temps, un autre phénomène incroyable autant qu'était imprévisible le succès des livres au temps de Gutenberg, le web explosait et modifiait en profondeur notre rapport à l'information. Et les limitations de Html, incapable de suivre la demande (vidéo, sons, animations...) ont forcé l'émergence d'un modèle particulier, celui des plugins.

Or le web est accessible par toutes les machines, PC et Mac inclus. Les deux phénomènes ont alors fait jaillir une étincelle dans l'esprit des informaticiens comme deux fils électriques qui se touchent : concrétiser, 20 ans après, leur fantasme ! En écrivant un plugin qui fonctionne sur PC et sur Mac, on pouvait dès lors écrire des logiciels totalement portables entre les deux univers sans avoir à faire de choix !

On touche tout l'irrationnel de ce choix puisque le Mac ne compte plus en part de marché depuis longtemps. C'est un fantasme, une vieille blessure, une sorte d'affront enfin lavé. C'est tout. *Technologiquement cela n'avait pas de sens dès le départ.*

Flash de Adobe s'est engouffré dans cette brèche fantasmagorique et finalement sans grand intérêt technique.

Mais l'histoire est pleine de rebondissements passionnants ! Flash a réveillé d'autres inspirations : celles de logiciels graphiques, animés, sonorisés. Flash a apporté la modernité au développement, bien plus que Apple n'a pu le faire en 30 ans. Flash a démontré qu'il était possible d'écrire des sites Web sous forme de vraies applications lookées, fonctionnelles, sachant gérer des données et pouvant rendre des services dignes d'une “vraie” application. D'autres technologies ont poussé dans la même direction (Ajax, Asp.NET...) toutes avec leurs points forts, mais aucune avec ce qui finalement avait fait le succès en partie d'Apple : le look.

Silverlight, quelques années après, a suivi cette voie. Celle du fantasme de l'Everywhere, WPF/E, WPF Everywhere.

La gloire de l'horizontalité

Ces quelques dernières années qui ont connu la montée en puissance de Flash puis celle de Silverlight ont été celles d'un grand rêve uniquement motivé par un vieux fantasme, une vieille frustration : écrire un logiciel une fois qui puisse tourner sur PC et sur le divin Mac adulé en secret par tous les PCistes...

Flash partout, WPF everywhere.

Le cross-plateforme universel était né. Un seul plugin porté sur PC et Mac, un seul logiciel à écrire qui fonctionne de la même façon partout.

Le plugin traversant *horizontalement* l'espace des OS permettant à un logiciel d'atteindre ce vieux rêve de la portabilité PC/Mac.

On notera au passage que tout cela n'avait dès le départ aucun sens véritable, le Mac ne pesait rien sur le marché, et il ne pèse toujours rien aujourd'hui. Toute cette débauche d'effets spéciaux n'a été motivée que par un vieux fantasme, une vieille frustration des PCistes.

Mais une fois encore l'histoire allait connaître un rebondissement incroyable, transformant un rêve inutile en un besoin réel !

Le marché des tablettes et des Smartphones émergeant simultanément à tout cela a donné un autre sens bien plus concret à ce fantasme, un autre terrain de jeu pour exprimer avec encore plus de vigueur ce vieux rêve de l'Everywhere : Si le saint plugin était porté sur toutes ces nouvelles machines, on tenait enfin l'universalité tant désirée !

On y était presque.

On a caressé un instant ce rêve.

Nos yeux ont brillé de voir ce fol espoir se concrétiser doucement sous nos yeux.

Mais comme certains équipages Apollo qui n'ont fait que tourner autour de la Lune sans s'y poser, revenant sur terre avec le goût amer d'être allé si loin sans avoir touché le sol de notre satellite, un assassin a brisé notre rêve.

La bulle de rêve éclate alors. Le retour à la réalité est d'autant plus cruel que nous y étions presque !

On comprend mieux la suite...

Jobs l'assassin

Loin du mythe du grand visionnaire que sa mort ne fait que renforcer, Jobs a assassiné plusieurs rêves pour sa seule gloire, sa fortune personnelle et celle de ses actionnaires.

Assassiner la marque Apple

C'est un peu hors du sujet, mais Jobs a commis deux assassinats pour satisfaire sa petite personne, et si le second nous intéresse plus, le premier sera certainement celui que les historiens du futur retiendront...

Au lieu de redresser Apple et d'envahir le monde de Mac, d'autoriser peut-être (et enfin) les compatibles Mac, au lieu de vendre de l'OSX, au lieu de faire de Apple ce que cette marque aurait pu devenir sans toutes les erreurs du passé, non, Jobs a préféré délibérément sucer la substantifique moelle de l'aura Apple pour vendre de la musique en ligne et copier le concept de baladeur de Sony. Plus tard il copiera l'idée du Smartphone de longue date représentée par Microsoft Windows Mobile, il pillera sans vergogne l'idée de la tablette qui existait déjà dans le monde Windows, sans parler de l'Origami, ce concept présenté par Microsoft avant qu'Apple ne fasse l'iPad...

D'une marque essentiellement élitiste avec toute l'aura qui allait autour (qu'on aime ou pas, la polémique de l'élitisme Apple faisant partie de l'image de marque), Steve Jobs s'est contenté de faire du grand public en bradant cette image de marque unique qu'était celle de Apple.

Aujourd'hui l'effet se tasse, Android mange sérieusement la suprématie de l'iPhone. Jobs a tué l'unicité de la marque. Il s'en est mis plein les poches, mais il a fait d'une marque d'exception une marque banale qui finira par se diluer dans la concurrence qui n'a pas forcément les deux pieds dans le même sabot...

Assassiner le rêve de l'Everywhere

Mais Steve Jobs a commis un deuxième assassinat. Il a tué le rêve de la portabilité, il a saccagé le fantasme de l'Everywhere.

En refusant, animé par une haine totalement psychotique, la présence de Flash sur les iPhone et iPads, alors même qu'il l'acceptait sur les Macs, Jobs a tué les plug-ins horizontaux.

Fin du rêve. Retour à la réalité. Back to 80's.

Puisque tout le fantasme derrière les plug-ins comme Flash et Silverlight était l'universalité, la portabilité absolue, et puisque tout un pan de la microinformatique devenait inaccessible aux plug-ins, c'est toute la justification même de ces plug-ins qui se trouvait sapée à la base. Déracinée de toute justification.

En toute logique puisque le monde Apple serait pour toujours inaccessible, les plug-ins ne pouvaient plus prétendre être la solution de portabilité pour le futur.

Jobs l'assassin venait de briser un rêve qui était presque accompli et nous renvoyait 30 ans en arrière.

Par filouterie, usant d'une stratégie machiavélique classique, le même Jobs mettait en lumière Html 5. Le buzz autour de cette version de Html allait finir de semer la confusion dans les rangs des éditeurs de plugins, Adobe et Microsoft en tête.

Il faut bien prendre conscience que ce n'est pas Html 5 qui aura tué Flash ou Silverlight, Html 5 n'aura été que l'instrument manipulé par un fou haineux, un type qui voulait la peau de Flash, qui vomissait Flash pour de sombres raisons. Un type considéré d'ailleurs comme caractériel et instable. Steve Jobs.

De l'horizontalité à la verticalité : le grand Shift

Devant ce mur infranchissable que Jobs a dressé aux portes des iPhones et des iPads, qui pouvait lutter ? Qui pouvait prétendre forcer la serrure ?

Personne.

Les plugins ne pouvaient plus prétendre à l'universalité et c'est ainsi qu'ils disparurent...

Adobe et Microsoft l'ayant compris et sachant que cette bataille était terminée à deux doigts d'être gagnée ont alors "shifté" leurs stratégies.

Html 5 et ses quelques gadgets faisant le buzz, à quoi bon expliquer à un client que cela ne réglerait pas tous les problèmes que cela ne serait pas la panacée ? ...

Un proverbe arabe dit "on ne fait pas boire l'âne qui n'a pas soif". Quelle sagesse...

Pour diriger une entreprise depuis longtemps je peux vous assurer que même animé des meilleurs sentiments, contre un buzz énorme vous ne pouvez rien. Perdre son temps pour expliquer à un client des détails qu'il ne veut pas entendre (et que souvent il ne peut pas comprendre) est la meilleure façon de perdre la vente... Nous ne sommes pas dans le monde des Bisounours, il faut payer le loyer et donc vendre ce que les clients veulent acheter, même s'ils se trompent, même si on le sait. Les plus honnêtes comme j'essaye de l'être perdent des clients en refusant de les envoyer sur des voies aventureuses, mais je sais que chaque client que je perds ainsi sera récupéré à chaque fois par quelqu'un qui n'aura pas ma conscience et qui lui pourra payer les traites de sa BMW... Des fois ça donne à réfléchir je peux vous le dire, même si personnellement cela ne change rien à l'honneur que je tiens à garder quand je conseille un client. J'ai souvent eu raison, des clients sont parfois revenus avec courage me l'avouer, mais entretemps d'autres s'étaient mis l'argent de leur budget dans les poches... Déontologiquement je reste vierge, mais quand il s'agit de sortir le chéquier de l'entreprise pour payer les charges on se dit que ça fait une belle jambe !

L'honneur, la vérité, toutes ces valeurs ne se mangent pas en salade et ne payent pas le loyer.

Microsoft, comme Adobe, ne sont pas des enfants de chœurs, ce sont des règles qu'ils connaissent parfaitement. Et plutôt que de se perdre à expliquer que rien ne peut remplacer Flash ou Silverlight, ils ont abdicé.

Peut-on leur en vouloir de faire preuve de réalisme ? Difficile. Même si certains amoureux de l'un ou de l'autre de ces plugins se sentent aujourd'hui abandonnés, sacrifiés, voire trahis.

La lucidité impose de se rendre compte que l'assassinat commis par Steve Jobs est irrémédiable. Ses conséquences sont immenses, comme tout assassinat. Mais il faut vivre avec et s'adapter à un monde qui change.

Avant la mort d'une personne tout reste possible, se dire des choses, construire un futur, s'aimer, se détester. Après la mort de cette personne tout se fige en l'état. Et pour ceux qui survivent un monde différent émerge, un monde qui ne sera jamais plus "comme avant", mais un monde plein de vie, de potentiel et d'espoir aussi.

La mort d'un rêve c'est la même chose. Une fois le deuil accepté, il faut penser à la reconstruction. A nouveau sourire, à nouveau se projeter dans le futur, à nouveau espérer, à nouveau conquérir d'autres espaces.

C'est pourquoi notamment Microsoft a décidé de faire un grand "shift".

Le fantasme de l'horizontalité par le biais des plugins est mort. Rien ne sert de pleurer. Nous connaissons le nom de l'assassin, il est mort lui aussi, passons à autre chose.

Realpolitik !

C'est donc en toute logique et en toute lucidité que Microsoft a déplacé sa stratégie. Et il faut saluer ce réalisme.

C'est de la Realpolitik comme diraient nos voisins teutons.

La realpolitik (de l'allemand : politique réaliste) désigne « la politique étrangère fondée sur le calcul des forces et l'intérêt national »

Microsoft a donc choisi, comme Adobe, de laisser tomber le rêve de l'horizontalité. Aujourd'hui si vous voulez développer une application cross-plateforme, traversant horizontalement tous les OS, l'écho de la voix de Jobs vient vous dire d'outre-tombe "Faites du Html 5 !".

Tout le monde sait bien que Html n'a jamais été implémenté de façon identique par les browsers, tout le monde sait que Html 5 en faisant exploser le nombre de balises et leurs interprétations possibles, il n'y vraiment aucune chance qu'une application soit réellement portable facilement d'un browser à l'autre. Encore moins qu'avant.

Tout le monde a conscience que Html 5 n'arrivera pas demain à la cheville de ce que savent faire Flash et Silverlight depuis hier.

Tout le monde sait que pour en arriver au niveau de Silverlight 5 il faudra certainement 10 ans à Html et 20 ans aux browsers pour qu'enfin ils implémentent un noyau commun stable permettant un niveau de portabilité aussi transparent que ne l'offrent Silverlight depuis des années.

Non content d'avoir tué un rêve, Jobs a fait perdre 20 ans de progrès à l'informatique.

Mais cela tout le monde n'est pas prêt à l'entendre et encore moins à le penser. Le buzz Html 5 est encore trop fort. Normal, on parle de quelque chose de virtuel pour l'instant.

Mais Html 5 dans la bouche des informaticiens et des médias c'est comme les filles et les motos dans la bouche des garçons de 15 ans, ils n'ont touché ni l'une ni l'autre mais ne parlent que de ça ... Il faudra attendre que tout ce petit monde murisse, que les premières désillusions éclatent au grand jour pour qu'enfin cesse le diktat du tout Html 5.

D'ici là, Microsoft, avec raison, aura "shifté" !

Trop compliqué d'expliquer. Et puis oser parler franchement des faiblesses de Html 5 aujourd'hui c'est forcément devenir une cible de tous les fanatiques de la planète, passer pour un aigri, un jaloux, pire un hasbeen s'accrochant au passé.

Bien entendu Microsoft ne tient pas à être classé dans ces catégories !

C'est pourquoi ils agissent sur deux axes en même temps :

- Clamer haut et fort "vive Html 5 pour le cross-plateforme" !
- Préparer avec vigueur l'avènement de la verticalité avec Windows 8

Clamer haut et fort les mérites des Html 5 pour le cross-plateforme est une stratégie très intelligente. Elle en fait hurler certains mais parce qu'ils n'ont pas compris que les dés étaient jetés. Jobs a tué le rêve du plugin universel, c'est fait, c'est fini, il faut se rentrer cela dans le crâne.

Donc clamer les vertus de Html 5 pour l'horizontalité n'est pas une trahison, juste de la lucidité. Certes Html 5 n'est pas encore clairement définie, et cette norme ne le sera peut-être jamais d'ailleurs (certains veulent arrêter de numéroté les versions de Html pour en faire un projet en perpétuel évolution), certes les outils sont à la traine, certes les browsers n'implémenteront jamais la "norme" de la même façon, certes, certes... Mais il n'y a rien d'autre !!! Le cross-plateforme passera forcément par Html 5. C'est comme ça. Rien ne sert de se lamenter.

Toute la question est en réalité de savoir si le cross-plateforme, ce vieux fantasme, n'est tout simplement pas qu'un simple fantasme sans réel fondement... Le Web est "cross-plateforme" depuis toujours. Html 5 ne changera donc rien à cela. Et quand aux applications, à supposer que Html 5 le permettent, imaginez-vous un logiciel

médical, un système de montage vidéo professionnel ou autre logiciel de ce type écrit en Html et en JavaScript ? Mais les désillusions viendront bien assez vite, les rêveurs se réveilleront un jour...

Mais la verticalité alors c'est quoi dans tout ça ?

Windows 8 : la verticalité

Le rêve "on est tous des frères et nos programmes marcheront partout" est cuit pour les plugins.

Microsoft n'a aucun intérêt réel dans Html qui ne lui appartient pas. Internet Explorer est diffusé gratuitement, aucun bénéfice à en attendre. Les outils de développements existent tous en versions gratuites très performantes. Tout cela coute de l'argent à Microsoft, pas mal, et ne rapporte et ne rapportera jamais rien. Il faut en avoir conscience.

Si Microsoft clame les vertus de Html 5 pour le cross-plateforme c'est qu'il n'y a plus d'autres choix pour ce besoin particulier. Ils se contentent donc de brosser les buzzeurs dans le sens du poil pour faire djeuns et dans le vent, ce qui est toujours bon pour l'image, cela ne coute rien...

En revanche Microsoft prépare sa riposte d'une autre façon : **la verticalité**.

L'éditeur a déjà sorti, très tardivement, Windows Phone 7, mais cela n'a pas bien marché. Le renouveau avec Nokia s'annonce déjà plus prometteur. Mais WP7 est un OS spécifique. Qui n'a aucune liaison ni portabilité avec Windows.

De plus Microsoft est absent du monde des tablettes, un marché très porteur.

Ils auraient pu faire comme Apple ou Google et réutiliser l'OS des téléphones pour proposer rapidement des tablettes.

Mais le lien avec Windows aurait, une fois encore, été perdu.

Et on sait que les revenus de Microsoft se font sur Windows et sur Office.

Le grand "shift", la big stratégie qui se cache derrière tout cela devient clair : de l'horizontalité (qui ne rapportait rien, Silverlight étant gratuit) on passe à la verticalité : un seul et même OS du PC de bureau au Smartphone en passant par les tablettes, Windows.

Le cross-plateforme à la sauce Windows

Ce n'est plus un plugin qui ne rapporte rien qui sert à faire un programme unique portable sur toutes les machines, c'est l'OS lui-même qui est portable sur les trois types de machines : PC, Tablettes, Smartphone.

Elle est là l'astuce.

Au lieu de traverser horizontalement les différents OS pour faire marcher des applications sur tablettes ou Smartphones, Microsoft nous propose un OS unifié qui fonctionne sur tous ces supports.

De fait, les applications WinRT seront cross-plateformes : elles seront utilisables sur PC, Tablettes et Smartphones.

Et pour ceux qui veulent à tout prix que leur appli fonctionne sur un Mac ou sous Linux, c'est à dire des marchés totalement ridicules en termes d'acheteurs potentiels, ils n'auront qu'à utiliser Html 5 comme le grand gourou Jobs leur a dit... Et bonne chance les gars !

(Re) back to the 80's

Tout cela est bien joli. On comprend mieux ce qui se passe depuis quelques mois. On comprend mieux le pourquoi du comment et, je l'espère, vous commencez à y voir plus clair sur l'avenir de Silverlight.

Car ce dernier a un avenir : il va être le seul moyen de déployer une application sous Windows 8 via le Web sans passer par le market place, ses tests, ses redevances et ses 30% piqués à chaque vente...

C'est pour cela que Silverlight 5 comporte de nombreuses nouveautés bizarrement utilisables uniquement sous Windows.

On se détache progressivement du rêve horizontal et des 2% de part de marché que représentent les Mac... (Celles de l'iPhone sont à jamais inaccessibles, rappelez-vous...).

Et quand WinRT sera totalement accepté, et que Microsoft aura peut-être ouvert une porte évitant de passer par un système payant (le market place), *il est fort probable que Silverlight tire définitivement sa révérence*. Mais cela est spéculation. Nous sommes obligés d'attendre pour voir. Personne ne peut lire le futur.

Ce dont nous sommes certains c'est que *WinRT va devenir l'outil cross-plateforme de Microsoft*, via un Windows 8 qui tournera sur toutes les plateformes justement. Pas celles des concurrents, Microsoft s'en fiche, comme s'en fichent Google et Apple.

C'est un repli sur soi-même, un retour aux fondamentaux : vendre sa marque et tenter de conquérir le monde avec elle.

L'avenir se présente ainsi pire que dans les années 80 où le délicat choix de développer pour Mac ou PC se posait.

L'avenir que nous a offert Steve Jobs est celui de trois blocs. Le retour de la guerre froide, comme si demain l'URSS refaisait surface et qu'on demande à chaque pays dans le monde de choisir son camp. Ami ou ennemi ?

Le lègue de Jobs est un monde déchiré, atomisé, l'anéantissement d'un rêve d'universalité.

Demain vont se dresser trois blocs : Microsoft, Apple et Google.

Chaque développeur devra choisir son camp. Ses alliés et ses ennemis.

Si tu développes pour Google, ne crois pas pouvoir porter ton appli sur iOS ! Et si tu travailles sous WinRT ne crois pas un instant pouvoir faire tourner tes softs sur Android. Il faut choisir camarade !

Le faux baba cool Jobs aura créé un monde de haine et de division, en tout cas il aura participé à rendre ce monde plus haineux et plus divisé qu'il ne l'était avant ses géniales interventions dans le monde de la micro-informatique...

Trois blocs, trois visions du monde

- Le bloc Apple, la dictature très fermée au monde extérieur, te propose un OS pour les tablettes et les téléphones, et un autre pour les Mac. A toi de voir. Les outils de développements ne sont pas terribles, les règles sont très dures, tu ne seras jamais riche, l'Apple Store lui le sera, tu travailleras à la gloire de la marque.
- Le bloc Google, grand sourire aux lèvres massue planquée dans le dos, te propose le rêve du cloud, pas d'OS, toutes tes données confidentielles dans leurs serveurs à eux. C'est gratuit, mais tu es leur chose, ils scannent tes mails et demain ton agenda, ta borne wifi, pour mieux te gaver de pub. Aies confiansssssssss... Comme disait le serpent du Livre de la Jungle. Ou celui qui pervertit Eve.
- Le bloc Microsoft te propose un OS, Windows 8, qui marche sur toutes les machines. WinRT qui accepte que tu ne sois pas un fan de Xaml et C# et qui te permet de développer en natif même avec des saletés comme Html et JavaScript, si ce n'est pas de la grandeur d'âme ça ! Bien sûr il faudra payer ta dime au market place. Il faut bien que tout le monde vive...

Voici les trois blocs entre lesquels demain il va te falloir choisir camarade !

Windows 8 : Avantage cohérence ?

Lorsqu'on y regarde de plus près, Microsoft répond très tardivement à tous ces bouleversements que sont les tablettes et les Smartphones, l'évolution du Web.

C'est vrai qu'ils se sont laissés débordés, ils l'avouent d'ailleurs.

Mais finalement leur solution semble la plus cohérente et la plus novatrice.

Android par exemple ne propose qu'une pâle copie de Windows pour son OS tablette et téléphone : des icônes, un bureau avec des icônes. C'est rassurant, ça ressemble à du vieux Windows.

Mais Microsoft avec Windows Phone 7, Windows 8 et Metro Style est le premier éditeur à proposer quelque chose d'autre, de plus novateur. Enfin on sort du système d'icônes et de la copie du bureau Windows qui date de ... tellement longtemps.

Avec Windows 8 sur PC et Tablettes, et bientôt assurément sur téléphones, Microsoft proposera un tout cohérent, sur les toutes les plateformes.

Bien sûr rien ne sera tout rose. Par exemple WinRT ne saurait être utilisé par de nombreux logiciels qui ne peuvent se satisfaire d'un mode sandboxé, et d'autres part nombres d'éditeurs refuseront d'abandonner du jour au lendemain 30% de leurs revenus au market place. C'est pourquoi Silverlight 5, ou même WPF 4.5 *qui utilisent le bureau "classique" de Windows 8 auront de beaux jours devant eux.*

Mais pour écrire un logiciel pour PC, Tablette et vraisemblablement pour téléphone, WinRT va vite s'imposer. Réexploitant habilement tout ce qu'il y a de bon dans .NET, à commencer par les API et les langages (Xaml, C#, VB...), WinRT permettra à une application d'être présente dans le monde entier sur tous les supports.

Oui mais ceux qui fonctionneront sous Windows.

Conclusion

La guerre des trois blocs ne fait que commencer !

Petit soldat développeur, Directeur informatique, Chef de projet, il va te falloir faire un choix. Windows, Google ou Apple.

Oublie tout de suite Html 5, tu gagneras du temps.

Pense sérieusement à de vraies applications. Quelles bases installées et quelles bases futures seront les plus porteuses pour tes créations ?

Tu vois, tu commences à ressentir la grande angoisse des anciens, celle dont je te parlais en introduction, celle du choix qui s'offrait à nous dans les années 80, mais comme tout s'améliore avec le temps, tes choix à toi seront encore plus difficile à faire !

Merci Steve Jobs !

HTML 5 : la tragique fin d'un buzz

Publication : 14 avril 2012

L'avènement du natif, la fin d'un Internet 100% Web grignoté par ces dernières sur unités mobiles, la montée du Cloud, plus d'internet et moins de Web, cet article explique pourquoi HTML 5 ne sera jamais le miracle annoncé.

HTML 5, l'un des plus gros buzz sur du vaporware des vingt dernières années... Ce buzz prend fin de façon tragique. Il faut rapidement s'en rendre compte pour prendre les bonnes décisions pour le futur de vos développements !

Le Web est-il encore l'avenir de l'Homme ?

Le Web est partout, il fait partie de nos vies, il a envahi nos cerveaux, nos écrans, et même celui des petits enfants (malgré l'interdiction de s'inscrire en dessous de 13 ans, Facebook compte beaucoup d'enfant de moins de 6 ans qui ont leur page !).

Le Web ce n'est qu'un outil de l'Internet, ce fantastique réseau qui, comme tout progrès de cette taille, amène autant d'avancées extraordinaires que de mésusages dangereux.

Mais si Internet est là, et pour longtemps, en va-t-il de même pour le Web ?

Mais n'entend-t-on pas dire tout le temps "Le Web est devenu un support et un transporteur d'information essentiel tant au business qu'à la vie privée." ?

Non ! C'est une erreur... **Internet est ce vecteur, le Web n'est qu'une application spécifique de l'Internet !**

Le Web sera-t-il présent dans l'avenir ? Oui, mais plus à 100%.

Le Web se programme, il a ses normes (beaucoup), ses modes, ses courants de pensées, ses dogmes. Et il a HTML comme base. Et cela ne changera pas de sitôt.

Ce qui va changer c'est que le Web ne va plus être 100% de l'utilisation d'Internet, comme le PC cesse de représenter 100% du marché des logiciels.

La version 5 de HTML est bien trop en avance sur les browsers, ces derniers implémentent les choses de façon trop anarchique, les outils de développement ne sont pas à la hauteur, etc. Mais la norme 5 de HTML s'imposera au fil du temps, comme ce fut le cas pour les 4 précédentes.

Mais voilà, si le Web restera le Web, ce sont les machines pour s'y connecter qui ne sont plus les mêmes ! Et cela change TOUT.

C'est Internet l'avenir de l'Homme, pas le Web...

Internet for ever

On le voit clairement, Internet en tant qu'infrastructure mondiale de communication libre fait aujourd'hui partie de nos "besoins vitaux", qu'il s'agisse de vendre de la

musique ou bien de préparer une révolution au Maghreb... Des plus riches aux plus opprimés, des plus bavards pour le plaisir de pérorer sur FB aux plus muselés derrière les pare-feu chinois, Internet est une "plomberie" devenue indissociable de l'avenir des entreprises comme de celui des peuples et des individus.

Le problème pour imaginer l'avenir c'est souvent qu'on associe trop facilement ce qui est séparé uniquement parce que l'habitude a créé un lien artificiel entre les choses.

C'est le cas du Web et d'Internet.

Internet la plomberie, le Web l'interface. Une simple application d'Internet. La plus voyante jusqu'à hier. Mais pas demain...

Internet et Web ne sont pas liés. *Le second n'est qu'une façon parmi d'autres d'utiliser le premier et pas la seule.*

Les browsers ne sont que des applications spéciales proposant une interface pour utiliser Internet d'une certaine façon.

Mais aujourd'hui Internet peut être utilisé directement par des applications sans passer par un Browser ! Des millions d'ordinateurs dans le monde à chaque seconde échantent des données en passant par Internet mais en ignorant totalement les browsers !

Bref : Confondre l'avenir d'Internet avec l'avenir du Web est une erreur grossière...

L'explosion des form factors coulent le Web

S'il est facile d'accéder au Web sur un PC ou un Mac, car double-cliquer sur l'icône d'un browser est aussi simple que de le faire sur n'importe quelle application, l'accès au Web sur des tablettes ou des Smartphones est déjà moins aisé. Tout utilisateur d'unité mobile l'a constaté.

Sur les Smartphones c'est la taille de l'écran qui gêne le plus. Et le nombre de manipulations pour arriver sur une page Web donnée, taper l'adresse avec un clavier tactile qui couvre l'écran, etc...

Sur les tablettes on retrouve les mêmes problèmes, celui de la taille trop petite de l'écran en moins malgré tout.

Mais le problème n'est pas là. Il est dans l'UX et les possibilités offertes.

Lancer une application native sur un Smartphone ou une tablette offre un confort et des possibilités bien supérieurs à toute visite d'une page Web !

Contrôler finement les deux caméras, le détecteur de présence, la boussole, le GPS, accéder aux données du téléphone, automatiser les tâches, tout cela ne peut se faire

correctement qu'en natif, indépendamment du fait qu'une application compilée restera toujours plus réactive que du JavaScript interprété sur une machine donnée.

Le royaume du natif qui nait sous nos yeux relègue le fantasma d'universalité et de portabilité de HTML 5 à sa nature même de fantasma.

Un monde natif

Smartphones, tablettes, et bien entendu depuis toujours les PC et les Mac, offrent une UX très largement supérieure lorsqu'ils sont programmés en natif que n'importe quelle page Web, même en ajoutant le Canvas à HTML.

La preuve ?

Même les rois du Web comme Google proposent sur leur OS Android des **applications natives** pour accéder à Gmail ou à Google+ applications typiquement Web s'il en est !

Facebook propose une application native pour Android ou iOS !

Intermarché, Super U, Cdiscount, et bientôt tous les grands acteurs de la consommation **offrent dès aujourd'hui des applications natives iOS et Android en place et lieu d'aller visiter leur site !** Quitte parfois à ce que l'application ne soit qu'une copie du site, **ce qui montre la précipitation de certains de ces projets et l'urgence ressentie de changer de cap...**

Faisant écho à d'autres de mes billets parus ces derniers temps, **le monde qui se profile est un monde d'applications natives et non plus de pages web.** Refuser cette évidence c'est s'embourber dans un Web qui n'est plus porteur d'avenir.

La mort du Web ?

Bien sûr que non !

Le Web ne mourra pas, pas plus que l'arrivée des Smartphones et des tablettes ne tuera le monde du desktop. Il faudrait être illuminé et ne rien comprendre à ce qui se joue pour penser une chose pareille.

Mais, car il y a un "mais", les PC ne représenteront plus 100% du marché du logiciel comme ce fut le cas dans ces 20 dernières années.

Et dans cette même voie et pour les mêmes raisons (en plus de celles invoquées plus haut), **le Web ne représentera plus 100% de l'utilisation d'Internet.**

Le Web n'est pas mort, HTML 5 non plus. Il y aura un HTML 6, un 22 aussi pourquoi pas. **Mais ce Web ne pourra plus prétendre être le seul moyen d'accéder à de l'information vivante sur Internet.**

Une grande part de ce qui était véhiculé par le Web au travers des browsers sera demain utilisé par des applications natives au travers d'Internet et du Cloud sur des unités mobiles.

Et alors ? Et Alors ?

Hé hé ... Zorro n'est pas arrivé, n'en déplaise à Henri Salvador... Aucun sauveur, aucun messie des temps nouveaux pour protéger le pauvre développeur et l'empêcher d'être noyé dans le Styx des choix cornéliens. Aucun salut n'est à attendre.

Bien au contraire. *Les blocs en présence vont tout faire pour maintenir la pression et assurer leur suprématie sur le marché.* Steve Jobs l'a fait, ils ont tous été obligés de suivre...

Jobs l'avait bien compris dans son refus de Flash. Personnage détestable et pas aussi génial que ça mais à qui il faut malgré tout reconnaître une sacrée avance sur les autres lorsqu'il s'agissait de parler du futur et de protéger la valeur de ses actions chez Apple...

Car en refusant Flash, donc en tuant Silverlight aussi mais sans le nommer, en prônant HTML 5 comme "universel", Jobs savait très bien pour en être l'instigateur que seules les applications natives auraient un avenir sur iOS et qu'il lançait tout le monde sur une mauvaise piste histoire uniquement de se débarrasser du danger que représentaient Flash et Silverlight ! Ne pas le croire serait du coup le prendre pour un crétin ce qui serait très éloigné de la réalité tout de même...

En tuant le rêve d'applications cross-plateforme comme Flash ou Silverlight, Jobs a lancé le monde entier sur **une fausse piste, celle de HTML qu'il savait lui-même condamnée...** Mais tout le monde est tombé dans le panneau...

En évitant l'avènement d'applications cross-plateformes passant par Internet sous Flash ou Silverlight, Jobs s'assurait qu'aucune "vraie" application ne pourrait passer sur son matériel en évitant le market place !

Les moutons qui ont suivi le buzz HTML 5 ont été envoyés au mur, ils ont été utilisés pour tuer Flash et Silverlight, seuls véritables moyens d'offrir la qualité d'UX du natif tout en échappant au market place des applications natives...

Pauvres fous, trimbalés par un manipulateur de talent... Comme un seul homme ils se sont jetés sur le buzz, ils ont piétiné les seules technologies qui étaient en mesure, enfin, et pour la première fois en micro-informatique, d'apporter le rêve de la portabilité !

Jobs est mort. Mais sa manipulation a fonctionné.

Aujourd'hui, celui qui vise l'universalité, le cross-plateforme pour ces applications n'a plus d'options.

Le monde du natif a pris le dessus, et pour longtemps.

Une part du Web est morte, pour toujours. Le Web vecteur de l'universalité au travers d'applications Flash ou Silverlight est mort. HTML ne viendra pas sauver le monde de ce constat pour toutes les raisons expliquées ici.

Reste trois blocs proposant trois plateformes incompatibles sur lesquelles les applications natives sont privilégiées et où le Web est voué à ne plus être qu'un citoyen de seconde classe. Le Web, donc les browsers, pas Internet !

Alors, sur ce Web, développer en HTML 5 ou en 3 n'a plus guère d'importance... Si vous n'offrez pas une application native, si vous vous acharnez à vouloir à tout prix développer en HTML 5, en dehors des PC et des Mac, vous ne toucherez personne sur les autres form factors...

C'est en cela que la grande époque du Web est révolue. Celle du natif vient de renaître. **Internet est plus puissant et plus vivant que jamais**, mais ces applications spéciales permettant de s'en servir d'une façon spéciale que sont les Browsers servis par Html ne sont plus des outils exclusifs du futur.

La modernité n'est plus à HTML ni aux browsers, la modernité est dans le natif.

Conclusion

Apple, Google, et demain Microsoft, nous proposerons trois visions du monde. Il faudra choisir son camp car croire qu'un éditeur de logiciel ou qu'un développeur pourra être compétent dans les trois mondes est une folie.

A ce jour Apple conserve une vision très réactionnaire des choses : d'un côté le Mac, de l'autre le mobile. S'ils ne se réveillent pas demain ils chuteront comme ils ont chuté il y a 20 ans. Apple, l'éternel serpent de mer de la micro, qui chute à chaque fois pour les mêmes raisons : son manque d'ouverture, son égoïsme congénital, son culte de la personne qui trinque quand le gourou disparaît... Le gourou a pu revenir une fois car il n'avait été qu'évincé la première fois. Maintenant il est mort, et son retour est une affaire à laisser entre les mains des spiritistes...

Google n'est pas un spécialiste du software et encore moins des OS, Android n'est qu'un relookage rapide de Linux (même origine que Ubuntu). Google s'est attaqué avec un talent incroyable aux Smartphones devenant un vrai concurrent d'Apple. Ils ont créé la surprise là où personne ne s'attendait à les voir et encore moins à réussir. Ils ont étendu ce succès aux tablettes. Le rapprochement Android / Ubuntu qui s'opère en ce moment doit être le signe que Google nourrit peut-être un appétit pour les plateformes desktop. Attention danger ! Mais nous n'y sommes pas.

Microsoft, d'ici octobre prochain, va nous proposer une vision cohérente, un OS qui fonctionne sur tous les form factors, avec les mêmes compétences, réutilisant même celles déjà engrangées dans l'entreprise avec C#, .NET, WPF, Silverlight. Un OS et une plateforme de développement très au-dessus de tout ce qui existe chez les

concurrents. Visual Studio, Expression Blend, C#, et même le très commercial couple Html5/js. Il y en a pour tous les goûts, toutes les compétences. Mais avec une seule équipe, une seule formation.

Alors certes Microsoft va arriver bien tardivement sur le marché des mobiles. Mais ce seront les seuls à proposer une vision cohérente du "natif" avec WinRT sur toutes les plateformes.

Je ne connais pas un DSI qui ne serait pas séduit par cette offre qui soulage des lourdes charges que représenterait la constitution d'équipes de développement mixtes, accumulant des talents Windows, Android et iOS.

La mode du BYOD est un leurre. Car 80% des tablettes actuelles sont des iPad. Quelle économie pour l'entreprise d'éviter l'achat de quelques tablettes si cela doit être compensé par l'embauche d'une équipe spécialisée sous iOS en plus des équipes Windows déjà en place ?

Le coût d'une formation adaptée ou ne serait-ce que d'un seul salarié permanent est plus lourd sur une année que l'achat de dizaines, centaines de tablettes Windows 8 ARM...

Demain, Internet ne sera plus 100% du Web, demain le natif sera maître sur les mobiles, demain Microsoft proposera une solution unifiée, il est temps de se préparer à ces changements et d'oublier le buzz HTML 5 qui vous a fait perdre tant de temps.

Car demain, c'est aujourd'hui.

L'avenir proche du développement : quels environnements pour quels produits ?

Publication : 30 mars 2012

Comment choisir les plateformes de développement sur lesquelles investir ? Une approche rationnelle et personnalisée pour aider à faire les bons choix. Cet article sous ce prétexte pratique permet de faire le tour des possibilités, des OS, des outils, de la formation, bref de faire un tour d'horizon des plateformes. Toujours d'actualité !

Septembre et son BUILD sont déjà loin, mais ses effets réels ne se feront sentir que dans quelques mois encore... Entre temps, hors de la sphère Microsoft il s'est passé, il se passe, et il se passera beaucoup de choses ! Pour le développeur cette période de calme, de transition à la fois dictée par les annonces des uns et des autres et par le ralentissement de la crise autant que celui qui entoure toute élection présidentielle doit être saisie comme une chance. L'opportunité de réfléchir au futur et de s'y préparer.

Plusieurs approches

Le marché se divise aujourd'hui selon plusieurs plans de découpe qui sont autant de clivages amenant à une réflexion différente :

- La dichotomie Grand Public / Entreprise
- La segmentation des OS (iOS / Android / Windows / Ubuntu ?)
- La diversité des form factors (Smartphones / tablettes / portables / PC)
- Les standards de fait ou à venir (Html / Java / .NET / WinRT)
- Les outils (leur disponibilités, leur puissance, les langages proposés...)

Le développeur doit se positionner face à ces grands choix car, disons-le tout de suite, jamais il ne sera possible de briller et de proposer une carte de visite d'égale compétence dans tous ces domaines à la fois !

Choisir est quelque chose de difficile. Pas tellement parce qu'on doit sélectionner la « meilleure solution », mais surtout et plutôt parce qu'on doit « éliminer » des solutions à qui on ne prête pas les mêmes qualités, en se trompant peut-être.

C'est le problème de faire des « impasses » à un examen. Le principe est simple, assez logique, si je tombe sur l'un des sujets que j'ai bien travaillé je suis sûr d'avoir une bonne note. Mais en contrepartie je dois sélectionner quels sujets je ne travaillerai pas... et sur lesquels je sècherai totalement si je tombe dessus. Toute la difficulté se situe dans ce pari que j'appellerai ici « **le pari de l'impasse** ».

Le pari de l'impasse

Nous ne parlons donc plus de « choix » de la « meilleure solution » tout cela est trop subjectif. Nous sommes forcément plus influencés par ce que nous aimons que par ce qui nous laisse indifférent... Difficile dès de lors de tenter une analyse *a priori* si on ne se borne pas par des principes plus objectifs.

Alors je vous propose d'utiliser plusieurs techniques dont un raisonnement qui se base sur ce qui vous laisse indifférent plutôt que de vous enflammer sur ce qui suscite en vous passion et exaltation.

Le « pari de l'impasse » consiste donc nous plus à choisir ce que nous pensons être le meilleur pour l'avenir mais à éliminer ce qui nous semble le moins bon.

Par exemple, plaçons nous sur le premier plan de clivage évoqué plus haut : la dichotomie Grand Public / Entreprise.

Il faut faire un choix. C'est certain. Vous dirigerez-vous vers l'un ou l'autre ? Le choix des outils, des supports, de l'OS, etc., beaucoup de choses vont en découler.

Si vous raisonnez par choix positif, peut-être votre enthousiasme pour les jeux ou pour l'IPad vous incitera-t-il à vous diriger vers le Grand Public.

Si vous raisonnez par la logique du « pari de l'impasse » la question devient toute autre : *lequel des deux mondes puis-je éliminer en minimisant mes risques à moyens et longs termes ?*

Le Grand Public, on le voit aux statistiques de vente de tous les market places du monde, se limite à leur Top 50 : des jeux, encore des jeux, toujours des jeux, et les pages jaunes... Les premières applications payantes un peu intelligentes arrivent avec des scores bien lointains dont la rentabilité apparaît négative, le tout maintenu dans un flou artistique par les market places qui veulent des tas d'applications et qui ne veulent donc pas que nous sachions la vérité sur les pertes des créateurs de logiciels...

S'orienter vers le Grand Public c'est donc développer des jeux avant tout. Je ne parle pas de la 100.001ème version de la lampe torche. Laissons cela aux ados qui apprennent à développer en Html / JS...

Avez-vous les compétences, l'équipe (graphistes, musicien...) pour créer un jeu capable d'égaliser le succès d'Angry Birds ? Si oui, foncez ! Si non, le Grand Public c'est comme se lancer dans la chanson, si vous n'avez pas une structure financière lourde derrière vous, beaucoup s'y tentent, bien peu réussissent...

Reste donc l'Entreprise... Mais ce n'est qu'une ébauche de pensée, bien d'autres facteurs sont à prendre en compte, ce que je vous laisse faire.

Pouvez-vous ignorer le marché Grand Public ou celui de l'Entreprise ?

C'est tout l'intérêt du « pari de l'impasse » ! Apporter un angle de vue différent de celui par lequel on aurait raisonné habituellement. Forcer une réflexion différente.

Et ce questionnement doit être fait pour chaque catégorie, chaque clivage énoncé plus haut...

C'est un outil pour vous aider à vous situer, pas un mode pensée absolu qui me permettrait de vous imposer un choix. Je ne développerai donc pas plus cet aspect de la réflexion pour tous les points, cette réflexion vos appartient, *c'est le chemin que vous prendrez vous-mêmes qui vous éclairera au fur et à mesure sur la pertinence de vos choix.*

Ma vision des choses

Après avoir présenté les clivages essentiels, les décisions à prendre et une méthode pour tenter d'améliorer votre réflexion, je vais vous livrer ma propre vision des choses. Elle ne vaut que pour moi dans ses réponses, mais son chemin peut trouver résonance chez bien des lecteurs, donc les intéresser.

Les offres du marché

Je suis parti de l'ensemble des offres du marché, présentes et à venir dans un avenir proche (donc réaliste, il n'était pas question que je puisse prendre des décisions sur de la science-fiction...).

OS	Form factor
iOS	Tablette (iPad) Smartphone (iPhone)
Android	Tablettes (cibles multiples) Smartphones (idem)
Windows 8 Metro Style	Tablettes (Win8 for ARM) PC Smartphones (WP8)
Windows classic WPF	PC
Silverlight	Smartphones WP7 Web (quasi absence) Intranet
Ubuntu	Smartphones, Tablettes, PC

De nombreuses remarques peuvent être faites, et chacun pourra leur donner le poids qu'il juge, encore une fois je présente ici ma vision des choses, pas une vérité absolue.

Apple

La première remarque est qu'on ne peut pas ignorer le phénomène Apple, qu'on aime ou pas, qu'on trouve cela outrageusement monté en épingle ou non, le buzz existe depuis des années, ce n'est donc plus un buzz, c'est une réalité de terrain. Le market place de Apple est celui qui rapporte le plus.

Concernant Apple on doit aussi noter que l'iPad3 semble marquer le pas côté évolution, comme ce fut le cas de l'iPhone 4S. Jobs n'est plus là et cela se sent, Apple se banalise, la concurrence est arrivée à faire aussi bien voire mieux. Les parts de marché d'Apple semblent toujours orientées à la hausse mais nul ne peut prévoir l'ampleur du rééquilibrage certain avec Android et Windows 8 dans les 2 ans à venir. [Une prudence que je ne regrette pas avec le recul !]

Apple est en outre un marché fermé, une marque, avec une seule machine par catégorie, aucune concurrence sur iOS, des outils de développements pas terribles et

un style de gestion, de production (chinoise), d'autoritarisme, d'élitisme qu'on aime ou non.

Apple, bien qu'aillant ajouté deux ou trois choses sur son OS desktop ne cherche absolument pas à créer une véritable convergence entre les Mac et l'univers iPad/iPhone.

Android

Android est un phénomène plus intéressant. D'abord parce qu'il s'ouvre sur des tas de gammes de machines diverses offrant toutes des options plus ou moins personnalisées avec une vraie concurrence entre fabricants. Pour l'instant Android n'est pas présent sur le segment des PC mais on a pu voir quelques prototypes de machines Google. Plus intéressant encore est la convergence entre Android et Ubuntu. On sait, et ce n'est qu'un début, qu'environ 7000 lignes de code Android vont être intégrées à Ubuntu, et parallèlement Ubuntu sera installable sur les téléphones Android (à partir des double-cœurs) tout en restant compatible à 100% avec Android et ses fonctionnalités (agenda, téléphone...).

A terme il faut envisager la possibilité pour Ubuntu de s'approprier toutes les applications Android et devenir ainsi du jour au lendemain une alternative crédible pour les PC face au Mac et à Windows 8. Le tout en s'appuyant sur la bonne image de marque Android.

Windows 8

Windows 8, quant à lui, marque une évolution d'esprit majeure chez Microsoft, une forte envie d'unification et de cohérence. On sait que l'OS tournera sur PC, tablette mais aussi sur Smartphones avec WP8 qui reprend un noyau Windows en laissant tomber le noyau Windows Mobile. On sait que sous WP8 c'est WinRT qui deviendra la nouvelle plateforme de développement.

Windows 8 sera alors le seul OS à adresser tous les form factors de l'instant.

On peut bien entendu être critique sur le mélange Metro Style / Bureau classique, en faisant un OS à « deux têtes » incompatibles entre elles. Je suis le premier à trouver cela assez mal géré. De même que jusqu'à la dernière bêta l'utilisation de Windows 8 sur une machine desktop n'est vraiment pas très ergonomique, même avec un écran tactile. Mais tout cela peut encore s'arranger jusqu'à la finale (on a déjà pu voir des améliorations notables entre la première preview réservées aux développeurs et la bêta publique).

N'oublions pas non plus que Windows 8 est un OS charnière entre les anciens systèmes et Metro Style. Un peu comme Windows 95 à cheval entre le 16 et le 32 bit. Si Microsoft réussit son pari, Windows 9 n'aura plus de bureau classique (ou bien caché pour la compatibilité). *Mais on ne peut pas exclure que Windows 8 connaisse le sort de Vista, surtout en entreprise où Windows 7 serait alors le nouvel XP pour les dix*

ans à venir pendant que se développerait un marché grand public sous Metro Style (tout nouvelle machine sera fournie comme cela).

Silverlight et WPF

Je terminerai sur Silverlight, ce n'est pas un OS mais c'est une plateforme de développement pour laquelle j'ai toujours eu un penchant. Sous WP7 c'est même l'unique moyen de développer (en dehors de XNA qui sera arrêté). Officiellement, au moment où j'écris ce billet, Silverlight 5 est sorti depuis quelques temps sans rencontrer l'engouement des versions précédentes (certainement parce que les nouveautés sont très orientées OOB avec certificat ce qui rend tout cela bien peu utilisable). En outre rien ne dit qu'il n'y aura pas de version 6 mais rien ne dit non plus le contraire... *Toutefois, le passage de Windows Phones 8 à WinRT devrait sonner le glas de ce mini WPF portable...* Resterait donc le Web mais Silverlight n'a jamais beaucoup été utilisé pour le Web grand public. Ne subsiste plus que le monde de l'entreprise où Silverlight a su séduire notamment sur les Intranet où il offre confort, puissance, centralisation tout en évitant l'installation et la mise à jour lourde qu'impose WPF sur chaque machine du parc. Silverlight, dans le meilleur des cas semble donc s'orienté vers un marché de niche. Cela peut être rentable.

Hélas, ce marché de niche ne sera certainement pas suffisant pour que Microsoft maintienne encore très longtemps Silverlight qui aura été durement touché par la logique « no plugin » lancée par Microsoft, stratégie en réalité suiviste de la position de Steve Jobs qui détestait Flash et qui a fait tout son possible pour dénigrer cette solution qui comme Silverlight était pourtant cross-browsers, puissante et au point.

C'est injuste pour Silverlight qui est et reste une plateforme agréable et puissante, bien au-dessus de ce que Html 5 ne peut offrir. *Mais nous devons considérer que Silverlight est au développement ce que James Dean est au cinéma : une star fauchée en pleine gloire et pleine jeunesse.*

Après avoir incarné la survie de WPF, jamais vraiment adopté, Silverlight cèdera sa place à WinRT et plus proche de lui encore à WPF qui devient, par le jeu de tous ces changements, la seule solution moderne pour développer des logiciels natifs Windows non sandboxés (en dehors de l'espace WinRT donc) et échappant au contrôle planétaire du market place de Metro Style... Bon nombre d'éditeurs risquent d'être attirés par WPF pour ces simples raisons. Les éditeurs de logiciel et leurs actionnaires adopteront-ils WinRT et accepteront-ils de verser 30% de leurs ventes à Microsoft et de voire chuter leurs recettes et leurs actions ou bien s'accrocheront-ils au bureau classique de Windows, avec WPF ou d'autres solutions, l'avenir va être passionnant sur ce point !

Bref, Silverlight et WPF ont encore leur mot à dire. Etrange retour du sort, Silverlight tendait à devenir le survivant, ayant écrasé WPF. Mais le "suicide" (en fin le meurtre...) de Silverlight en pleine gloire par Microsoft change la donne. Et WPF pourrait devenir l'avenir des gros développements en bureau classique, donc plutôt en entreprise (ou

bien les logiciels de type musique tels Cubase, Ableton Live, ou encore les outils de dessins 2D ou 3D comme Illustrator, Maya...).

Dans ce qui suit je ne classe pas Silverlight et WPF non pas que je n'accorde plus de valeur à ces environnements (au contraire) mais tout simplement parce que j'envisage l'avenir et ce qu'il faut apprendre pour ne pas être largué. Silverlight et WPF sont déjà dans le champ de mon expertise, quoi que leur réserve l'avenir je suis prêt. Je vais donc m'intéresser plutôt à ce auquel je ne suis pas tout à fait préparé, comme beaucoup de lecteurs.

6 solutions possibles...

Me voici, comme vous, devant pas moins de six solutions possibles ! Puis-je prétendre être un expert dans les 6 à la fois ? Non. Puis-je passer mon temps à tester et jongler avec autant d'OS, de langages, de bibliothèques, de trucs et astuces à connaître pour chaque form factor ? Non plus.

J'en reviens donc à la stratégie de l'impasse... Quelles sont mes réelles préférences, qu'est-ce qui m'indiffère et comment rationaliser mon approche ?

Mes préférences sont claires, je préfère très largement .Net, C #, Xaml, Visual Studio et Expression Blend que tout le fatras d'outils et de langages qu'offrent la majorité des solutions listées.

Qu'est-ce qui m'indiffère le plus ? L'Apple-mania, Linux. Question de choix personnels. Rappelons que je vous livre ici mon cheminement de pensée car c'est ce cheminement qui peut vous aider, mes conclusions, mes « j'aime » et mes « j'aime pas » me sont strictement personnels, à vous de bâtir votre propre réflexion en partant de vos sensibilités !

Classer les choses en termes de préférences instinctives, je l'ai dit, ce n'est pas rationnel. Je vais essayer d'ajouter des points objectifs pour tenter un classement des solutions.

L'investissement

Finalement c'est un point essentiel. Faut-il le compter en premier ou en dernier, peu importe, en tout cas l'investissement humain et financier pour atteindre un niveau d'expertise respectable dans chaque solution qui s'offre à moi est un facteur crucial.

Je vais donc, en toute honnêteté, tenter d'attribuer un poids à l'investissement qu'il me faudrait faire pour atteindre l'expertise nécessaire dans chaque contexte.

L'investissement financier

Il est assez facile à comptabiliser et j'ai voulu prendre un barème simplifié. Je considère que pour une plateforme donnée, un form factor donné, il me faut au moins une machine « native » pour développer ou tester les logiciels. Par exemple si

je vise les tablettes Android il faut que j'en achète au moins une... se contenter de l'émulateur n'est pas professionnel, sous aucun environnement. Idem pour des OS comme Ubuntu, le faire marcher dans une machine virtuelle va bien pour un test, pas pour développer, il faut mobiliser une machine.

Je ne prends pas en compte ici les machines que j'e possède déjà, c'est un investissement qui serait à ventiler selon mes choix.

J'ai ainsi considéré le cout d'acquisition du matériel en hecto Euros, échelle qui se calle le mieux sur le prix moyen de tous les form factors (par exemple 8 pour une machine aux environs de 800 euros).

L'investissement humain

Il est difficile à quantifier en jours une formation sur des plateformes aussi diverses. Je préfère une approche plus globale. J'ai choisi le pourcentage de choses à connaître pour arriver à l'expertise. Par exemple sur Silverlight en Intranet ce pourcentage est proche de 0% (pour moi), pour du Silverlight sous WP7 je considère qu'il me manque encore 5% de connaissances pratiques pour être au top (investissement que je ne fournirais pas en raison de la fin de WP7 et de son mode de programmation particulier qui passera à WinRT sous WP8, mais c'est une autre histoire).

Le tooling

C'est bête mais je n'envisage pas, sauf sous la contrainte, de travailler avec des outils de développements mal fichus. C'est triste, mais quand on est habitué à Visual Studio on est un peu handicapé : tous les autres environnements paraissent moins bien. Malgré tout il faut bien attribuer un poids à ce tooling. Certains sont plus agréables que d'autres.

Il faut aussi intégrer la dimension de la qualité. On peut ne pas avoir un grand amour pour certains langages mais leur reconnaître une qualité correcte. Par exemple je n'ai jamais aimé Java, j'aimais beaucoup Delphi, j'ai adoré C#, mais Java et JavaScript ne m'ont jamais séduit. Cela se joue certainement à peu de chose pour Java tellement C# en est proche. Mais c'est comme ça, c'est mon sentiment. Je dois en tenir compte. Sans pour autant dévaloriser des langages comme Java qui sont parfaitement utilisables.

Comme je ne voulais pas écrire un livre sur cet aspect des choses, j'ai décidé d'attribué aussi un pourcentage. Un pour mes préférences, un pour les qualités intrinsèques des outils disponibles sur une plateforme donnée.

Découpage des solutions

Le découpage proposé plus haut n'est pas assez fin pour dégager certaines tendances. J'ai donc re-segmenté là où cela me semblait nécessaire.

J'ai supprimé Silverlight et WPF du tableau puisque j'ai déjà l'expertise, ce qui n'est pas forcément le cas de tout le monde (il s'agit bien de ma vision des choses et de mon cas personnel, à adapter donc !)

Solution/ notes	Invest. Machines	Cibles	Invest. Formation	Invest/Cible	Inversion (1/x)	Intérêt	Tooling	Total * 100
iOS	12	2	1,0	6,0	0,17	0,2	0,4	1,36
WinRT C#/Xaml	12	3	0,2	0,8	1,25	0,6	1,0	75
WinRT Html/JS	12	3	0,8	3,47	0,29	0,2	1,0	5,8
Android	7	2	0,9	3,15	0,32	0,5	0,4	6,4
Ubuntu	10	1	0,9	9	0,11	0,1	0,5	1,15
Ubuntu sur Android	17	3	0,9	5,11	0,19	0,3	0,5	2,85

L'investissement par cible est calculé comme suit :

(Investissement machines divisé par le nombre de cibles visées) multiplié par le pourcentage de l'effort de formation.

Ce sous total représente une *charge* (investissement humain et financier pondérés par le nombre de cibles offertes par la solution). Plus la charge est grande, plus le résultat est grand. C'est tout l'inverse d'une « préférence ». Il convient donc d'inverser ce résultat pour pouvoir continuer les calculs en chaînes. C'est-à-dire obtenir un nombre d'autant plus petit que la charge est grande. Sinon nous en arriverions à mettre en première place l'environnement le plus coûteux et offrant le moins de cibles !

Une simple inversion (1/x) est donc calculée pour poursuivre les calculs. D'où la présence de la colonne Inversion

Le total final est multiplié par 100 pour le rendre plus lisible.

Le coût des machines est attribué en fonction des prix moyens. Pour iOS par exemple il faut compter environ 1200 euros pour acheter un iPad3 et un iPhone 4S. Pour WinRT il faut sensiblement le même investissement (tablette ARM Win8 + téléphone WP8). Pour Ubuntu je compte le prix d'un PC correct mais sans plus (1000 euros) alors que je ne le compte pas pour Windows 8 tout simplement parce que je suis déjà équipé Windows et que si je devais faire du Linux demain je serai obligé de garder mes machines Windows dans tous les cas, donc d'investir. Alors que la majorité de mes machines Windows 7 passeront en Windows 8 sans surcoût matériel. Ce choix est discutable dans l'absolu, mais je rappelle une fois encore que je prends en compte ma situation et non pas celle de monsieur tout le monde. Pour un lecteur qui ne ferait que du Linux et qui n'aurait aucun équipement Windows la situation serait

inversée. Mais je crois que ma situation reflète correctement celle de la plupart des lecteurs de Dot.Blog malgré tout.

Le nombre de cibles est un simple constat. WinRT couvrira les PC, les tablettes et les Smartphones WP8, donc 3 cibles. iOS couvre deux cibles (iPad et iPhone). Android couvre aussi deux cibles (tablettes et Smartphones). Ubuntu ne couvre qu'une cible (le PC) en faisant abstraction des déclinaisons (PC de bureau, PC serveur...). Ce cas reste toutefois discutable car l'intégration prochaine d'un module Ubuntu dans Android pourrait ajouter deux cibles de plus à cette solution et donc lui faire gagner quelques points.

And the winner is...

En fait, il n'y a pas de gagnant à ce "jeu" là... Il s'agit pour l'essentiel de choisir entre la peste et le cholera en faisant le moins mauvais choix pour l'avenir...

Solution	Total
WinRT C#/Xaml	75
Android	6,4
WinRT Html/JS	5,8
Ubuntu + Android	2,85
iOS	1,36
Ubuntu	1,15

Ce tableau est classé en ordre décroissant, plus une solution se trouve en haut plus elle mérite de l'intérêt.

Reste un point essentiel : le ROI.

ROI : comment le prendre en compte ?

On peut lire tous les comparatifs sur les différents market places, on peut essayer de deviner les chiffres qui ne sont pas publiés... Cela risque d'être très long et pas forcément plus juste que les certitudes faciles à connaître...

Ainsi, entre Android et iOS on sait que le market place de Apple est plus rentable pour les développeurs. Dans quelle proportion ? Sur quelle(s) catégorie(s) de logiciels ? C'est plus difficile à savoir.

Premier constat, dans notre tableau final les deux solutions sont si éloignées que quelques pourcents de rentabilité en plus ou moins pour Apple ne changera visiblement pas l'ordre obtenu.

Concernant Ubuntu, il se noie aujourd'hui dans les diverses moutures de Linux. Faisant abstraction des serveurs Web, Linux avance mais est loin de concurrencer Windows. Là encore prendre en compte les pourcentages ne changerait rien dans l'ordre du tableau tellement l'écart entre les solutions WinRT et Ubuntu sont éloignées. Surtout qu'ici nous parlons ROI et non pas pénétration du marché. Le ROI sous Windows reste évidemment bien meilleur que sous Linux et sous Ubuntu en particulier.

Pour WinRT lui-même il est difficile de prévoir son succès et son adoption... c'est même un grand point d'interrogation. *Windows 8 pourrait connaître le sort de Vista...* sur PC. Mais cela n'empêchera certainement pas les tablettes Windows 8 ou les Smartphones WP8 de se vendre correctement. Et comme tous les PC récents seront bientôt livrés sous WinRT, cela fait tout de même quelques centaines de millions d'utilisateurs potentiels à venir. Donc un ROI qui devrait être correct.

On notera aussi qu'il s'agit ici de choisir un axe, une direction pour l'avenir. Ce qui est déjà connu aujourd'hui n'entre pas dans le tableau... Je suis déjà un expert en Silverlight et en WPF. Si Windows 8 faisait un flop et que Silverlight et WPF soient utilisés encore plus qu'aujourd'hui, je suis déjà bien placé pour gérer cette possibilité. Encore une fois c'est mon raisonnement se basant sur mes compétences et mes connaissances, à chaque lecteur d'adapter tout cela en fonction de son cas !

Conclusion

Dans mon cas personnel, mes calculs m'indiquent que **WinRT sous C# et Xaml** est le meilleur choix. Celui qui me demande le moins d'investissement en formation ou en matériel, celui qui touche le plus grand nombre de cibles. Mais aussi celui qui m'offre le meilleur tooling, et qui est le plus proche de ce que j'aime bien faire.

Mais étonnement, ce même tableau me dit qu'ensuite j'ai tout intérêt à m'intéresser à Android bien plus qu'à WinRT en Html / JS ! Et finalement j'avoue que cet « avis » qui ressort du tableau ne me semble pas si idiot. Je vais donc le suivre. Quitte à fournir un effort d'apprentissage mieux vaut s'ouvrir un nouveau marché et de nouveaux clients que de rester à tourner autour du pot WinRT.

De même iOS se trouve en queue de tableau avec Ubuntu. Cela ne m'étonne pas, je n'y connais rien et je n'ai pas envie d'en savoir plus sur le monde Apple. Je n'aime pas Apple depuis longtemps, car justement j'ai déjà été client Apple et développeur sous Apple. C'est en connaissance de cause que je fuis la pomme donc. Le tableau ne fait que traduire ma forte subjectivité malgré des efforts d'objectivation.

Mais à tout prendre, le tableau m'apprends qu'il est beaucoup plus important de s'intéresser à Ubuntu en couple avec Android. Et là encore, dans l'absolu je trouve le conseil assez sage... Un tel mariage redonne beaucoup d'intérêt à Ubuntu.

Empêtré comme vous tous dans ce grand moment de solitude autour de la mort, il faut bien le dire, du marché Silverlight (pas du produit, pas de la techno, mais du marché) et même si je pense qu'une fois que le buzz Html 5 sera passé les gens reviendront peut-être à Flash et Silverlight (parce que ça marche, c'est cross browser et que le tooling est au point), il est temps de prendre aussi compte ce qui va venir et qui va modifier notre métier.

La poussée incroyable en si peu de temps d'Android sur les tablettes et les Smartphones ne peut être balayée d'un retour de manche. Pas plus que la sortie de Windows 8 qui se veut le grand unificateur sur tous les form factors avec WinRT.

Apple et son alléchant marché peut en tenter plus d'un, avec raison.

Que faut-il faire ? Sur quoi se former aujourd'hui pour être dans le coup demain ? Sur quoi faire l'impasse sans le regretter ?

Bien entendu mes petits tableaux ne sont qu'une excuse pour parler de tout cela, faire remuer en vous toutes ces possibilités.

En rien les conclusions de mes tableaux ne sont à prendre à la lettre. Elles ne peuvent pas même s'appliquer à quelqu'un d'autre que moi, et bien entendu, moi-même je ne prends pas que cela en compte pour mes choix futurs...

L'idée était de vous faire voyager dans toutes ses solutions en offrant une sorte de méthode qui, même si elle n'est pas absolue, vous permettra de mettre en évidence un ordre, une priorité à laquelle vous ne seriez peut n'être pas parvenue juste en y pensant comme ça.

C'est comme toujours le voyage qui compte bien plus que le site touristique qu'on prend en photo à l'arrivée. C'est en cheminant qu'on avance, lapalissade s'il en est, mais les portes ouvertes sont souvent celles qu'on oublie le plus souvent de franchir !

Réfléchissez-bien, pour Dot.Blog vous commencez à vous faire une idée des prochaines directions...

Le Web 3.0 = Pas de Web ?

Publication : 10 mai 2012

Le Web 3.0 pourrait ne jamais exister, en tout cas pas sous la forme d'un Web 2.0 ou de ce qu'on connaît aujourd'hui. Les raisons sont diverses et je joue ici un jeu de spéculation teinté d'une pointe de provocation afin de pousser la réflexion personnelle du lecteur. Toujours d'actualité !

En attendant de pouvoir sérieusement vous parler de WinRT, je m'intéresse beaucoup en ce moment à l'avenir du développement, tant pour les développeurs que les éditeurs dans la période très troublée que nous vivons depuis deux ans. Depuis plusieurs mois je vous fais part de mes observations, de mes doutes, de mes conseils et j'espère vous aider par mes réflexions à vous forger votre propre opinion. Souvent en décalage avec la pensée unique de l'instant, mes prévisions choquent parfois certains lecteurs. Force est de constater que le Web, donc HTML, a des soucis à se faire et j'en veux pour preuve de nombreux événements qu'il faudrait être fou d'ignorer...

Le natif remplace le Web

Ça c'est une certitude, **l'avenir est aux applications natives** sur les mobiles et j'en ai déjà parlé. Je renvoie le lecteur à mon billet "[Mobiles : les apps flinguent le Web !](#)" pour éviter toute redite.

Internet != Web

J'en ai parlé aussi, le Web n'est qu'une façon particulière d'utiliser Internet, au travers de logiciels spécifiques, les browsers, et de normes plus ou moins bien établies (html, Js, Css, Flash, Silverlight...).

La fin du Web n'est donc pas la fin d'Internet, bien au contraire. Internet, cette plomberie électronique planétaire est au contraire de plus en plus sollicitée par les services web, les serveurs de données OData, JSon, les applications et les données dans le Cloud, etc.

Mais il s'avère que **cette sollicitation sera de moins en moins le fait des browsers et de plus en plus le fait d'applications natives**, principalement sur les unités mobiles.

HTML 5, le buzz qui fait flop

J'en ai parlé longuement dans le billet "[Html 5 : la tragique fin d'un buzz](#)". **Si le Web se fait manger par les applications mobiles, ses techniques sont immédiatement ringardisées en perdant toute possibilité d'être "universelles"**. Simple question de logique puisque les applications natives des mobiles ne sont pas en Html mais en C#, Objective-C ou Java.

Dès lors Html 5 ne peut prétendre atteindre l'objectif que le buzz nous serine depuis deux ans justement et qui en faisait l'unique intérêt : l'universalité.

L'avenir est aux applications mobiles, donc natives, **le Web tel qu'on le connaît va se recentrer sur un minimum commun pour rester vraiment universel et devenir une devanture pour les applications mobiles et celles des différents market places...**

Ce Web-là sera plutôt développé en ASP.NET avec du HTML 3 ou 4 et un coup d'Ajaxisation qu'en HTML 5, sable mouvant d'une norme jamais terminée et forcément implémentée différemment par les différents browsers et leurs différentes versions (Attention danger ! Coût d'écriture et de maintenance prohibitifs en vue pour ceux qui ne suivront pas ce sage conseil !).

Un avenir segmenté

Concernant les plateformes mobiles (ou non, car WinRT tournera aussi bien sur les PC que les mobiles) je vous avais proposé un petit panorama des diverses possibilités dans mon billet "[L'avenir proche du développement : quels environnements pour quels produits ?](#)". Je ne reviendrai pas non plus ici sur cet exposé puisqu'il vous suffit de suivre l'hyperlien pour avoir l'original complet !

Cette segmentation pèse lourd s'il faut tout développer pour 3 ou 4 OS différents avec des langages et des outils différents ! Je doute qu'un seul DSI et qu'un seul patron soit d'accord pour multiplier par 3 ou 4 le nombre d'informaticiens de ses équipes et par autant ses coûts de maintenance évolutive et correctrice ! Il faudra donc choisir et faire des impasses ...

Le recentrage vers des applications Cloud ou des serveurs centralisant 100% du code métier est la logique de la prudence : fabriquer des frontaux iOS, Android, WinRT ou autres n'est alors plus qu'affaire d'interfaces utilisateurs ce qui permet de contenir l'explosion des coûts de réalisation et de rester plus agile pour réagir aux différents rééquilibrages et bouleversement du marché à venir.

Car le marché si illisible pour l'instant va tendre vers un nouvel équilibre qu'il est bien difficile de prévoir. Il y a donc urgence à se mettre dans la position la plus favorable possible pour aborder une mer agitée et les tempêtes qui éclateront sans crier gare !

Professionnellement, deux approches sont seules à même de permettre la réalisation de tels serveurs centralisés : Java et .NET. Bien que Microsoft pousse WinRT, il est bien évident que .NET reste dans le monde Microsoft la seule technologie sérieuse pour ce type d'applications (on voit mal pour l'instant une application serveur écrite sous WinRT, sandboxée et tombstonée ! Autant qu'on voit assez mal une mise à jour rapide des machines sous Windows server vers du Metro Style...).

Si j'ai bien entendu un penchant largement connu, et assumé, pour .NET, ceux qui choisiront Java pour reformater leurs applications en mode full server ne seront pas dans l'erreur. Ici seule l'affinité qu'on a avec la plateforme compte, elles se valent globalement du point de vue technique. **Toutefois, la cohérence de .NET, sa grande proximité avec WinRT et l'ouverture sur le monde mobile que cela représente m'apparaît un choix logique minimisant les coûts de formation du personnel.**

La mort d'Apple ?

Dans un billet fort intéressant, George Colony, patron de [Forrester](#), un des poids lourds du consulting aux US, annonce la fin en pente douce de Apple pour dans deux ans, en comparant sa chute à celle de Sony (Cf. "[Apple = Sony](#)").

L'argumentation est intéressante et je vous en joins à lire cet article.

Pour ma part je pense que Apple va chuter aussi parce qu'ils utilisent les mêmes recettes que celles qu'ils ont appliqué à l'époque du Macintosh : des produits très en avance vendus dans un élitisme le plus total soutenu par une gouroutisation du dirigeant mais dans l'isolement le plus total. Les procès faits à ceux ayant tentés, même légalement, de fabriquer des compatibles Mac a eu pour seul résultat la chute d'Apple alors numéro 1 de la micro au profit de IBM et de tous les compatibles PC.

Apple rejoue exactement la même scène, commettant les mêmes erreurs. Une fois la concurrence à niveau, ce qui est le cas aujourd'hui (Samsung et ses machines Android est devenu ces derniers jours le 1er fabricant de téléphones devant Nokia par exemple, Windows Phone 8 n'aura rien à envier à iOS non plus), le modèle Apple subira les mêmes défaites que par le passé. Même histoire, rejouée de la même façon. Même issue.

Mais nous restons ici dans la spéculation ce qu'il ne faut jamais oublier. Disons que l'ensemble de tout ce qui est dit ici converge vers un même point nous permettant de préparer l'avenir. Des certitudes sur le futur, cela n'existe pas.

La mort de Google et Facebook ?

On pourrait en conclure facilement que la mort de Apple laisserait un boulevard à Google qui deviendrait à terme le seul géant capable d'imposer la donne, face à Microsoft dont, pour l'instant, nous ne pouvons raisonnablement parier ni sur le succès ni sur l'échec de sa stratégie Windows 8 (puisque tout cela reste à venir).

Il s'agit encore de spéculation qu'il ne faut pas prendre à la lettre. C'est la tendance de toutes ces spéculations qui prend un sens, pas une en particulier.

Qu'est-ce qui pourrait bien tuer des géants tels que Google ou Facebook à l'horizon 2017 ?

La réponse est simple : les entreprises du web mobile qui fonctionnent sur des modèles économiques plus rentables, tout simplement...

Ce n'est pas votre serviteur qui le dit, mais [Eric Jackson sur le site de l'agence Forbes...](#)

Et ils proposent bien entendu une argumentation solide pour avancer un tel bouleversement. Il s'agit de spéculation et chacun pourra donner à cette analyse le poids qu'il veut (d'autant qu'un autre contributeur chez Forbes, Haydn Shaughnessy, nuance l'analyse de Jackson selon un autre point de vue). Mais ce n'est pas sans raison que dans ce billet j'ai tenu à rappeler d'autres de mes billets, d'autres analyses de géants du consulting mondial qui, au fil de l'eau, tracent toutes un même lit pour

une même rivière : celle d'une mort du Web actuel, *une mort qui bien entendu ne sera pas un "silence radio", mais un changement de cap, un réajustement des objectifs, un recentrage des technologies, une modification des moyens en œuvre et une redéfinition des priorités.*

Je vous propose de lire directement l'article fort intéressant de Challenges (en français) qui résume le billet de Jackson :

["Pourquoi Facebook et Google pourraient disparaître en 2017".](#)

L'une des bases de cette analyse est qu'une génération chasse l'autre. Il y a eu la génération Netscape, Yahoo, AOL, etc entre 94 et 2001, celle du Web 1.0 qui est morte depuis longtemps. Entre 2002 et 2009 le Web 2, le fameux Web social a vu naître des géants comme Facebook et LinkedIn ainsi que les spécialistes des coupons comme Groupon.

Mais, je cite l'article, **"Mais depuis 2010, l'internaute va progressivement se changer en mobinaute, à l'image de la profusion de nouvelles applications [natives donc, OD] créées pour l'iPhone d'Apple ou les Smartphones fonctionnant sous Android."**

Si le Web 1.0 a introduit la notion de "portail" agrégeant les informations venant de nombreuses sources, si le Web 2.0 a ajouté la dimension "sociale", le Web 3 pourrait tout simplement ne pas exister... **"Car moins que la naissance d'un Web 3.0, cette révolution [des mobiles, OD] signe en fait la mort du Web traditionnel."** (sic)

Le Web 3.0 ne sera pas, HTML 5 non plus donc.

De fait, **c'est tout le Web qui flanche**, c'est son modèle et ses technologies, incapables d'apporter une nouveauté capable de contrer **la montée massive et inattendue des applications natives sur les unités mobiles.**

Si le Web n'est tout simplement plus l'avenir d'Internet, il serait bien naïf de prêter encore le moindre intérêt à HTML 5 vous en conviendrez aisément.

Un Web plus "utilitaire" naîtra certainement, vitrine des applications mobiles, point d'accroche pour la publicité institutionnelle, support des blogs, bref, un Web très sage, très "pratique", un Web "aiguillage", un répertoire géant, loin des applications soi-disant universelles tout en arabesques tracées en SVG sur un Canvas HTML 5...

Conclusion

Dans cette période floue, chacun aimerait avoir une boule de cristal pour prédire l'avenir. Faire des choix en ayant la certitude de ce que sera le futur serait tellement simple...

Hélas, pas plus aujourd'hui qu'hier, pas plus en informatique que dans notre vie privée il n'est possible de lire l'avenir.

Est-ce à dire que tous les possibles sont ouverts et que donc on ne sait rien ?

Bien sûr que non !

Un éclaireur qui amène une troupe vers une cible bien cachée dans une forêt est-il un devin ? Non. C'est **un professionnel qui sait lire les traces bien réelles**, parfois tellement ténues que seul lui peut les voir, mais il s'appuie sur **des faits, des indices matériels** et non sur la magie.

Au travers de ces divers billets je ne veux pas être un mage, un gourou ou une madame Irma, **je ne suis qu'un éclaireur qui lit les pistes**. Libre à vous de me suivre ou de tracer votre route, seul au milieu de cette épaisse forêt dangereuse sans tenir compte de mes conseils, c'est votre droit le plus strict !

Mais pour en savoir plus : Stay Tuned !

Mobiles : les apps natives flinguent le web !

Publication : 3 mai 2012

L'avènement des unités mobiles et de leurs applications natives font vaciller le Web au point qu'il devient aujourd'hui possible de quantifier entre 5 et 10% la chute de fréquentation du Web classique au profit des « Apps » natives des mobiles (Cloud, services Web...). Un article toujours d'actualité donc.

J'en parlais dans mon billet "[Html 5, la fin tragique d'un buzz](#)" qui a fait réagir beaucoup de monde. Jugé par certains comme une vision délirante du futur, la grande majorité des lecteurs adhéraient à ma prédiction. Celle-ci se confirme bel et bien, **les applications natives sont l'avenir du développement, et non plus Html 5...**

N'en déplaise aux grincheux ou à ceux qui trop "intoxiqués" par le buzz Html 5 se refusent à voir la réalité en face, j'ai maintenant des chiffres à vous proposer : **La fréquence moyenne des sites Internet a baissé de 5,5% sur un an, la faute aux applications mobiles.**

Bien sûr je vois les mêmes sceptiques se préparer à m'invectiver en doutant de mes chiffres.

C'est pourquoi que je n'en dirais pas plus, et je renvoie le lecteur à cet article édifiant de BusinessMobile.fr "[Le succès des applications mobiles plomberait le trafic des sites Web français](#)".

C'est court, clair, et cela confirme ce que je disais : Les applications natives des mobiles prennent le pas sur les sites Web, ce qui, de fait retire tout son caractère "universel" supposé à Html 5 puisque **les mobiles ne se programment pas en ce langage destiné au Web.**

Certains futés me feront remarquer que si, les mobiles qui fonctionneront sous Windows 8 pourront être programmés en html / js, et que donc c'est bien l'avenir.

Je n'entrerai pas dans ce qui m'apparaît être ici un concours de mauvaise foi : d'une part Windows 8 n'existe pas encore (fin d'année 2012) on ne peut donc pas encore parler de sa pénétration du marché et encore moins de quels outils de développement seront ou non adoptés sur cet OS, et d'autre part il convient de rappeler que Html/js sous WinRT se programme de façon très spécifique liée à Windows 8 et **ne permet en rien de développer des applications pour le Web. Développer sous WinRT c'est développer en "natif"** pour Windows 8 quel que soit le langage choisi. Comme sous .NET il existe un choix de langages, dont aujourd'hui JavaScript et html. Mais tout cela permet de faire du natif et non du web.

Bref, **l'avenir est au natif** et le problème sera de choisir quel natif... iOS, Android ou le prochain Windows 8 ?

En tout cas ne pas écouter la petite voix qui vous dit depuis longtemps que Html 5 n'est pas l'avenir universel que le buzz fait croire serait certainement une grosse erreur. L'avenir du Web lui-même sera toujours Html mais pour la portabilité des sites on choisira certainement plutôt ASP.NET avec du Html 3 ou 4 et une poignée de JavaScript que Html 5.

Même s'il reste impossible aujourd'hui de prévoir le succès de Windows 8 (qui cumule à mon avis autant d'excellentes avancées que de points discutables, notamment Metro Style tout tactile imposé comme menu central même aux PC) nous savons d'ores et déjà que seul Windows 8 offrira une vision cohérente réexploitant le savoir-faire des équipes en place pour développer du PC au Smartphone en passant par les tablettes. Si Windows 8 ne fait pas un grand succès au niveau grand public, il y a fort à parier que les DSI réfléchissent à deux fois avant d'engager et de former du personnel supplémentaire pour gérer iOS et Android alors qu'ils possèdent déjà les compétences pour faire du WinRT...

C'est à ces entreprises, à leurs DSI, à leurs développeurs que je m'adresserai bientôt au travers d'une série d'articles sur WinRT programmé en C# et Xaml. En Juin nous disposerons d'une RC de Windows 8 qui permettra bien mieux que les bêta de se faire une idée du produit final. A partir de sa sortie Dot.Blog se tournera sur les diverses possibilités qu'offrent Windows 8, tant du point de vue WinRT pour les PC, Tablettes et Smartphones que dans sa partie "classique" avec WPF et certainement encore Silverlight (car rien ne l'égale pour des applications Intranet ou extranet de qualité).

Nous verrons ainsi au fur et à mesure, comment réutiliser son savoir-faire WPF/Silverlight pour créer des applications WinRT pour tablettes ou Smartphones, et même pour PC.

Nous sommes actuellement dans le creux de la vague, il n'est plus l'heure de promouvoir WPF ou Silverlight (malgré le grand intérêt qu'ils conservent) mais il est

trop tôt pour parler sérieusement de WinRT, ne serait-ce qu'en raison de l'absence de machines (tablettes et Smartphones) pour le faire tourner. Dot.Blog se veut constructif et s'attache à transférer une connaissance utile, jouer les essayistes avec des bêta et des outils de développement pas encore finis est juste ludique mais nous sommes trop occupés pour perdre du temps à jouer...

Mais d'ici juin les choses vont changer et l'apparition d'une première tablette (la Nokia fabriquée par Asus) pourrait nous permettre de passer du jeu à la réalité en expérimentant réellement les possibilités de WinRT avec une RC très proche de la finale [la fameuse tablette n'existe toujours pas, mais Microsoft propose Surface RT]. Appliquer MVVM à WinRT sera par exemple un questionnement passionnant, la gestion du tombstoning même sur PC posera de nouvelles questions. Quelles bibliothèques utiliser ? Comment contourner le manque de certains composants ? Prendre en compte le tactile, les différents form factors ? Publier sur le market place ? Des milliers de questions de ce type attendent des réponses pratiques... J'y travaille...

D'ici là, ne lâchez pas vos compétences C# / Xaml pour céder aux sirènes de Html 5, les sirènes, dois-je le rappeler, créatures fantastiques, étaient craintes des marins car elles faisaient échouer les bateaux qui cédaient à l'appel de leur chant si mélodieux, mais fatal...

HTML5 La norme qui n'existera jamais (Officiel) !

Publication : 24 juillet 2012

L'imbroglia d'un buzz qui se finit en eau de boudin...

Je vous l'ai annoncé depuis longtemps, notamment dans "Html 5 la fin tragique d'un buzz", mais aujourd'hui nous le savons, c'est sûr : HTML 5 est une norme **qui n'existera JAMAIS**. Il est temps de ressortir Silverlight que vous aviez peut-être rangé trop vite dans un tiroir...

Des raisons techniques

J'en ai parlé suffisamment ici dans plusieurs billets (comme par exemple "[Html 5, la fin tragique d'un buzz](#)", ou bien "[Facebook lâche Html 5, trop lent](#)") je n'y reviendrai pas car aujourd'hui j'ai **beaucoup plus drôle** à vous annoncer !

L'éclatement et la mort de la norme, c'est maintenant !

Le gag suprême c'est en effet l'annonce ces jours-ci du **divorce entre W3C et WHATWG marquant la fin de l'unité claire et nette qui fait un standard**, ni plus ni moins...

WHATWG ? Kesako ?

On n'en entendait que peu parler, mais ils sont très influent. En effet, le W3C c'est l'organe de normalisation notamment de HTML. Fait partie de cette instance un groupe d'éditeurs et d'industriels, le WHATWG, pour "Web Hypertext Application Technology Working Group", entendez par là des gars qui veulent faire au plus vite des sous avec HTML alors que le W3C s'intéresse surtout à pondre des normes quitte à prendre son temps.

Pourquoi le divorce ?

Le divorce entre WHATWG et le W3C s'explique tout naturellement : depuis un bon moment on sent bien que HTML 5 ne sera jamais terminé ou dans tellement longtemps (on parlait de 2014 dernièrement) que cela est une vraie gêne pour ceux qui veulent l'imposer et faire de l'argent avec, les gars du WHATWG.

Ces derniers ont donc décidé de "doubler" le W3C et de prendre leur indépendance pour pondre leur propre norme...

BOOMMM !

Le W3C continuera à travailler de son côté, et le WHATWG va créer sa propre numérotation et ses propres "versions officielles" de la "norme" HTML.

BOOMMM : Le standard vient d'exploser, vous avez entendu la détonation ? !

Le tour de la presse

Je vous livre en vrac quelques références dans la presse de ces derniers jours. Il faut consommer ça sorti du four sinon ça retombe comme un soufflet ! Comme tous les coups d'état les gars du WHATWG ont attendu entre le 14 juillet et le 15 août pour faire leur putsch, au moment où l'occident se dore la pilule au soleil de vacances méritées... Coupant l'herbe sous le pied à ceux qui voudront en discuter à la machine à café en septembre "tu connais la nouvelle ? HTML 5 a forké !" "C'est bon, on est tous au courant, c'est du vieux..." et hop, on étouffe l'affaire.

PC In-pact nous instruit dans "[HTML 5 : Le W3C et le WHATWG se séparent à cause de divergences insolubles](#)" sous-titré "c'était presque trop simple..." ça donne le ton.

Slashdot titre "[HTML5 Splits Into Two Standards](#)". **HTML se coupe en deux standards**. Ce n'est pas moi qui le dit hein...

Et de conclure avec sagesse "*Whatever happens, the future has just become more complicated — now you have to ask yourself 'Which HTML5?'*".

C'est à dire "**Peu importe ce qui arrive, le futur est juste devenu plus compliqué – maintenant vous devrez vous demander "quel HTML 5?"**". Je crois que c'est clair non ?

SoftPedia nous explique l'histoire dans "[No more HTML5, It's all HTML now](#)" (**Plus de HTML5, c'est tout du HTML maintenant**). Clarifiant ce qui ne peut plus l'être et

s'interrogeant d'ailleurs sur le futur de ce qu'on appellera désormais "HTML" tout court vis à vis de "HTML5". Ils notent que beaucoup de navigateurs utilisent déjà (plus ou moins) la pseudo norme WHATWG sauf Microsoft (le boxon était bien avancé il faut dire...). Surtout **ils notent que désormais les concepteurs de sites en HTML vont avoir à supporter des tas de "normes" qui changent en permanence**. Un enfer quoi...

Le plus amusant de tout c'est peut-être le blog de WHATWG qui vous l'annonce sans honte dans son billet "[HTML is the new HTML5](#)" (HTML est le nouveau HTML5, *et amusez-vous bien les gars j'ajouterais !*). La blague c'est le sous-titre du blog : "Please leave your sens of logic at the door, thanks !" : "**S'il vous plait laissez votre sens logique à la porte, merci !**". Ils ont du faire l'école du rire ou un truc comme ça, ce n'est pas possible autrement...

Tout le reste est à l'avenant. Je vous laisse, je suis tellement tordu de rire que j'ai du mal à taper !

Conclusion

Personne ne voulait d'un pseudo standard pas bouclé implémenté par chaque navigateur à sa propre sauce.

Qui voudra maintenant risquer son argent dans des développements incertains pilotés par deux normes concurrentes évoluant à leur gré et implémentées différemment par les navigateurs en dehors de fous furieux ?

HTML 5 est bel et bien mort.

Ou bien est-ce "HTML" puisqu'il faut l'appeler comme ça ...

Lequel ? Celui du W3C ou du WHATWG ?

Les deux ? Mais non ! Alors HTML 5.1 ou 5.2 ?

Pff vous n'avez rien compris !

A moins que ce délire soit le sommet de tous les délires récents en informatique et qu'en effet il n'y ait qu'une seule chose à comprendre, choisir HTML 5 c'est être fou à lier et avoir beaucoup d'argent à perdre !

Les lecteurs de Dot.Blog ne pourront pas dire que je ne les avais pas prévenu depuis longtemps, c'est déjà ça.

Stay Tuned !

(Excusez les tâches sur la feuille, je pleure de rire et ça coule !)

Silverlight 5 : Hommage à un oublié...

Publication : 8 juillet 2012

Un dernier hommage à Silverlight. Flash-back nostalgique (sans jeu de mot !).

Le buzz Html 5, le "big shift" de Microsoft, l'arrivée de Windows 8, tout cela est tellement énorme que pendant ce temps la version 5 de Silverlight est sortie dans l'ombre de ce brouhaha, une version oubliée, maltraitée et mal traitée à qui je me dois de rendre hommage, non à titre posthume mais pour l'avenir.

9 Décembre 2011

Cette date aurait dû rester dans les mémoires... C'était la release de la version finale de Silverlight 5. Hélas, comme je le disais en introduction, tous les buzz, tout ce bruit autour de technologies dépassées comme Html ou de technologies non encore relâchées (Windows 8), bref tous ces "non évènements" ont fini par étouffer le seul vrai évènement marquant : la sortie de Silverlight 5.

Un évènement marquant d'abord parce que Silverlight est arrivé à une maturité exceptionnelle, en étant l'un des plus beaux outils pour développer des applications RIA (Rich Internet Application), le plus beau pour être honnête. Ensuite parce que cette version 5 tient en elle la clé de nombreux développements à venir notamment en entreprise pour les intranet et les extranets. Car ne doutons pas un seul instant que quel que soit l'adoption de Windows 8 et WinRT, la grande majorité des entreprises ne basculeront pas leur parc informatique sous Metro Style avant quelques années.

Alors une fois qu'on a étouffé Silverlight 5 et qu'on refuse de passer immédiatement à Windows 8, on fait quoi ? Du Html comme il y a vingt ans ?

Soyons sérieux deux secondes...

Non, pour les entreprises la solution la plus rationnelle, la plus mature, la plus efficace, c'est Silverlight !

Alors fêtons, même à retardement le 9 décembre 2011 !

Une histoire longue qui n'est pas terminée

Il est bon de rappeler l'histoire de Silverlight car cela nous permet de prendre conscience qu'il faut des années pour qu'un environnement de développement devienne mature et que le chant des sirènes Html 5 sera depuis longtemps noyé dans les flots incertains des modes et des buzz quand, peut-être, enfin, existera-t-il des outils de développements à la hauteur de ce qu'est capable de faire Silverlight aujourd'hui...

Version	Ajouts	Release
Silverlight 1	<ul style="list-style-type: none"> Appelé WPF/E au départ 	2007

	<ul style="list-style-type: none"> • Le plugin est invoqué depuis la page html, il charge le code Xaml qui est exécuté. • Présence de l'API DOM • Gestion des formats Xml et Json 	
Silverlight 2	<ul style="list-style-type: none"> • Intégration d'un framework .NET • Peut exécuter des programmes écrits dans tous les langages de la plateforme • Xaml et code sont compilés et assemblés dans un package zippé, le XAP • Accès aux données REST, WCF Services, ADO.NET Data Services 	2008
Silverlight 3	<ul style="list-style-type: none"> • L'intérêt des développeurs se fait plus fort, SL surpasse Flash côté développement • Support de l'element to element Binding • SaveFileDialog permet de sauvegarder des fichiers partout sur la machine • Ajout de l'Out Of Browser • Peut tester automatiquement les mises à jour au lancement et les installer dans la foulée 	2009
Silverlight 4	<ul style="list-style-type: none"> • Toujours de nouvelles fonctionnalités • Support de la Webcam et du microphone • Support des impressions • Support du drag'n drop et du presse-papiers • WCF RIA services • Support de COM 	2010
Silverlight 5	<ul style="list-style-type: none"> • Devient une plateforme mature de développement pour les applications LOB • Debug du data binding • XNA 3D et support interne de la 3D • Améliorations multiples du rendu du texte • Ajout de P/Invoke • In-Browser HTML • Support 64 bits 	2011

Conclusion

Ce billet est un bref hommage à Silverlight et surtout une façon de rappeler à tout le monde que d'une part Silverlight, à ce jour, est toujours un produit vivant [Fin du support en 2021], et que, d'autre part, il reste le seul environnement de développement sérieux et mature pour de très nombreuses applications en mode "full web".

Que le buzz Html 5 et la haine de Steve Jobs pour Flash aient pu tuer l'avenir des plugins pour le web grand public, cela est possible. Le grand public, la "masse", n'a jamais été considérée comme une "élite" au comportement pesé et intelligent, c'est un euphémisme...

Silverlight ne s'adressera plus aux masses. Certes.

Mais vous ?

Vous, le développeur, ingénieur de formation, vous, le Directeur des Systèmes Informatiques, diplômé, supposé intelligent et œuvrant pour l'intérêt de votre employeur, vous tous, auriez-vous échangé votre cerveau pour celui d'une midinette suivant les modes et les aboiements des gurus morts ?

Oseriez-vous lancer votre carrière ou votre employeur sur des chemins aussi ringards qu'hasardeux que sont ceux de Html 5 pour créer toutes les applications internes de l'entreprise et qui en feront l'efficacité et la compétitivité de demain ? Obliguez-vous votre entreprise à une opération couteuse de mise à jour de vos licences Windows 7 en Windows 8 dans l'urgence, sans attendre que cette nouvelle plateforme ait fait ses preuves en entreprise juste pour tout réécrire en WinRT ?

Bien sûr que non ! Ou bien vous ne mériteriez pas votre salaire... Et je suis certain que vous êtes des professionnels, avec tout ce que cela implique de sérieux, de réflexion et de sagesse.

Alors ne perdez plus votre temps.

Pour tous vos développements internes, deux seules technologies s'offrent aujourd'hui à vous : Silverlight pour du full-web ou WPF pour des applications plus lourdes.

Ouvrez-vous à WinRT si vous avez à supporter les tablettes et les applications mobiles sur smartphone, c'est aussi le choix de l'intelligence. Cela vous coûtera toujours moins cher que la mode (encore une) totalement stupide du BYOD qui va vous envahir de iPad que vos équipes ne savent pas gérer. Mieux vaut acheter des lots de tablettes WinRT et ré-exploiter le savoir-faire de vos équipes que de "gratter" quelques euros en utilisant le matériel des employés mais vous imposant l'embauche et la formation de personnel qualifié sous iOS, un contre-sens économique...

Dans tous les autres cas, n'hésitez pas, vous avez assez perdu de temps comme cela. Les solutions matures et éprouvées existent depuis longtemps et elles ont un nom : Silverlight et WPF.

Ne jetez pas WinRT par les fenêtres : offrez-lui en ! (la solution pour un WinRT adapté au PC)

Publication : 4 mai 2013

Article plus récent qui prend en compte ce qu'il faut bien convenir d'appeler l'échec de Windows 8 et de WinRT. Que cela soit les ventes de Windows 8, sous celles de Vista, ou de Windows Phone 8 très loin derrière Apple et Google, qu'il s'agisse de l'adoption massive des entreprises ... de Windows 7, des pertes énormes (\$900 millions) pour Surface 1, etc..., j'explique ici ce qu'il faudrait faire pour que WinRT prenne la place qu'il mérite. Parfois trop d'innovation tue l'innovation et revenir à des bases plus consensuelles n'est pas forcément une reculade mais juste du bon sens.

Le mode fullscreen sur PC de bureau de WinRT est une erreur en termes d'UX. Toutefois ma critique sera constructive : j'ai trouvé une solution qui va satisfaire tout le monde et qui pourrait inspirer Microsoft pour rendre Windows 8 plus attractif...



Spécial anniversaire : 600eme Billet ! Illustration : Valérie / Olivier

Rendez le "S" à Windows !

Windows 8 est un OS de grande qualité. Deux petites choses malgré tout l'empêchent de s'imposer à mon sens. Ce ne sont pas des options mais un forçage et elles ne sont pas correctement adaptées aux PC :

- La suppression du menu "démarrer" du Bureau Classique
- Et le pire du pire, l'atroce confusion qui tue : le mode fullscreen qui veut transformer des PC modernes ayant de grands écrans, parfois 2 (jusqu'à 4 chez moi) en de grosses tablettes.

Ces deux points sont des handicaps rédhibitoires qui obèrent sérieusement l'avenir de WinRT sur PC tout simplement *parce qu'il y a forçage au lieu de choix.*

Il y a une politique du "*flingue sur la tempe*" au lieu d'une simple *proposition d'innovation qu'on peut activer ou non.*

Ces deux points concernent WinRT et l'UX Modern UI de Windows 8 mais absolument pas l'OS lui-même en tant que tel. *Mais hélas WinRT et Modern UI entraînent dans leur rejet Windows 8 avec eux, comme deux mauvaises fréquentations font plonger un bon garçon de famille...*

Je n'aime pas qu'on m'impose des voies, des visions de ce qui doit être le futur contre mon gré, mes habitudes et mes besoins. Et je ne suis pas le seul.

Dès lors que quelqu'un veut me faire faire quelque chose que je ne veux pas "mais c'est pour bien", j'ai les dents qui poussent, le poil qui se hérissé et je me transforme en Wolverine (cf. l'illustration du jour signée de Valérie et de moi-même). C'est un *basic instinct*, c'est vital, la seule personne que j'autorise à juger ce qui "bon pour moi", c'est moi, et moi seul.

En ayant forcé ce qui n'aurait dû être que des options (quitte à fournir Windows paramétré de base en Modern UI pour *inciter* l'utilisateur à l'utiliser) Microsoft a créé une sorte de rejet animal. Personne n'aime avoir un flingue posé sur la tempe. Régler ces problèmes est pourtant simple et ne remet pas en cause les qualités de l'OS en tant que tel ni de Modern UI et WinRT. *Il suffit juste de laisser l'utilisateur choisir ce qu'il considère être le mieux adapté à ses besoins.*

Il faut sauver à la fois Windows 8 et WinRT sur PC :

- Sauvez le soldat Windows 8 qui est un excellent OS en autorisant le boot sur bureau classique et le retour du bouton Start.
- Ne pas jouer le déni sur les vrais problèmes posés par Modern UI sur PC (inadapté, inesthétique, fullscreen gênant)

- Sauvez le soldat WinRT car il en vaut la peine. Car oui, aussi paradoxal que cela puisse paraître à première lecture j'aime aussi WinRT qui est une belle plateforme de développement unifiée.

J'aime Windows 8, et j'aime WinRT.

C'est le mélange des deux qui ne "marche pas" sur PC de bureau, en tout cas pas de la façon dont Microsoft l'a fait, à la fois techniquement et commercialement. Et non pas à cause de problèmes particuliers liés à l'un ou l'autre, mais simplement à cause du forçage Modern UI et de *l'absence de choix pour l'utilisateur*.

Les perspectives de Windows 8.1

Les bruits courent... Les "leaks" plus ou moins sérieux filtrent ici ou là, mais en réalité on sent bien que plus que des secrets bien gardés façon Apple Microsoft est en train de naviguer à vue et on ne sait pas ce qu'il y aura ou non dans Windows 8.1, eux-mêmes doivent en discuter tous les jours avec des arbitrages certainement musclés dans une ambiance tendue, stigmate des blessures profondes laissées par les luttes intestines allumées par Sinofsky... Deux ou trois choses semblent être acquises mais on doit prendre tout cela avec beaucoup de circonspection. Par exemple le fameux bouton Start devrait revenir. Mais avec de grosses limitations qui l'éloignent toujours de l'attente en la matière. Idem pour WinRT où un mode de partage de l'écran 50/50 compléterait le partage 25/75 actuel.

Si Microsoft limite les changements importants à ces petites modifications, ils perdront à la fois sur le terrain de la séduction, ce n'est pas cela que les utilisateurs attendent, et sur celui de la crédibilité, car ce sera une reculade malgré tout. Quitte à reculer, quitte à prendre en compte les besoins de l'utilisateur et les raisons de son rejet de Windows 8 et WinRT, autant y aller franchement !

Ce billet présente deux changements essentiels qui devraient tout simplement être repris dans 8.1 si Microsoft veut montrer que le message a été reçu.

Le menu démarrer, un problème qui se règle...

Le menu Modern UI est très bien fichu, la recherche d'une information, d'une application est simplifiée : il suffit de taper son nom... C'est plus rapide que la recherche dans le menu Démarrer qui réclame d'avoir été ouvert avant de taper sa recherche.

J'en conviens aisément.

Toutefois lorsqu'on ne travaille que sur le Bureau Classique, ce menu à tuile fullscreen est une gêne énorme. Personnellement il m'agace, et les allers et retours entre les bureaux me donnent le tournis.

Qui plus est, le menu à tuile est une bonne idée pour un écran de smartphone bas de gamme où puissance et mémoire imposent de n'installer que deux ou trois applications, mais c'est une catastrophe sur un PC de bureau possédant jusqu'à une bonne centaine d'applications (et même plus, j'ai déjà plus de 100 applications sur mon Galaxy SIII !). On ne fait que faire défiler les tuiles sur un ruban virtuel sans fin, on se lasse très vite. Taper le nom de l'application semble rapide... C'est vrai, mais encore faut-il se rappeler du nom de ce fichu utilitaire qu'on n'utilise que deux fois par an et dont on a besoin tout de suite...

Le menu "Start" original est bien mieux adapté au bureau classique, c'est aussi simple que cela. La segmentation par répertoire est un aiguillage que les libellés de groupe de tuiles de Windows 8.1 ne remplacent pas car le premier est automatique alors que l'autre réclame de l'organisation et de la frappe... Tout simplement.

Il faudrait donc que Microsoft rétablisse ce menu en bureau classique pour ceux qui travaillent de cette façon. Et ce ne sont pas de dangereux anarchistes anti-M\$ qui ne le disent, non, **juste des gens qui comme moi n'utilisent que des logiciels professionnels majoritairement Microsoft... Dont aucun ne fonctionne en Modern UI...**

Mais ne restons pas trop longtemps sur ce sujet, je vous ai présenté dans ces colonnes plusieurs solutions comme [StartMenu8](#) ou l'excellent [Classic Shell](#) que j'utilise personnellement et sans lequel je reformaterais mes machines Windows 8 en Windows 7. Classic Shell est gratuit de plus.

Le premier point est ainsi réglé : booter en bureau classique et avec un menu "démarrer" s'effectue simplement et gratuitement à l'aide d'un logiciel tiers. Il faudrait que cela soit de nouveau intégré à Windows 8. C'est possible, la preuve StartMenu8 ou Classic Shell le font...

Régler le problème du double bureau et du fullscreen WinRT

Le second problème de Modern UI c'est le mode fullscreen sur PC. Les applications conçues pour tablettes s'y affichent en énorme sans aucun gain pour l'utilisateur. **Ces applications devraient s'ouvrir dans des fenêtres sur le bureau classique.**

En réalité, sur PC de bureau au moins, le menu à tuile devrait disparaître au profit d'un seul bureau : le classique, et les applications WinRT devraient pouvoir y côtoyer les applications classiques. On devrait pouvoir déplacer les fenêtres, les retailler, les positionner sur un écran ou un autre, voire les recouvrir partiellement ou non qu'il s'agisse d'applications Win32/64 ou d'applications WinRT. C'est tout bête, c'est simple même. Mais est-ce possible allez-vous dire ?

ModernMix, le nom est lâché. Et ce n'est certainement que le premier d'une longue série, comme cela a été cas pour les programmes s'attaquant au boot en bureau classique et au retour du menu Démarrer comme Classic Shell.

Qu'est-ce que ModernMix ?

C'est une application utilitaire, comme Classic Shell, mais dont le but est différent. Ici il s'agit de rendre le fameux "S" à Windows, de rendre les applications WinRT autonomes afin qu'elles puissent (ce n'est pas une obligation, on peut le régler pour chacune) tourner dans des fenêtres sur le bureau classique...

A quel prix ?

Pour l'instant ModernMix n'est pas gratuit, il vous en coûtera \$4,99, c'est à dire moins de 4 euros (3,89 à l'heure où j'écris ces lignes). Donc presque gratuit.

Où le trouver ?

On le télécharge ci-dessous, en version d'essai 30 jours gratuits, et on peut aussi l'acheter (il existe une version plus chère mais qui n'ajoute rien d'autres qu'une sorte de bundle avec des applications pas tout à fait indispensables, mais à chacun de voir, ici je ne m'intéresse qu'au sauvetage de WinRT...).

<http://www.stardock.com/products/modernmix/>

Quel effet ?

Sans ModernMix, si vous appelez le programme Météo de Windows 8 (simple exemple), vous aurez sur un écran de PC un affichage de ce type :

Without ModernMix



C'est joli, mais 80% de la surface de l'écran ne sert qu'à l'affichage d'un Jpeg de nuage (ou autre selon le temps prévu). Parfaitement acceptable sur une tablette 10" ou de 7" cette mise en page est une catastrophe sur les grands écrans dont sont équipés les PC modernes.

Avec ModernMix :

With ModernMix



La météo comme d'autres applications WinRT s'affiche dans une fenêtre retailable, avec la possibilité d'épingler une app dans la barre, et bien entendu de "mixer" les apps WinRT avec des applis classiques, et de trimballer tout cela sur ces deux écrans (ou plus) pour agencer l'espace de travail tel qu'on le désire et non tel qu'un faux visionnaire l'a décidé de façon autoritaire à votre place...

Je n'ai aucun intérêt dans ModernMix, et on ne me l'a pas offert non plus comme cela arrive de temps en temps.

Non, **c'est juste que depuis que je l'ai installé il m'arrive de me resservir pour la première fois depuis un an de certaines applications WinRT.** Cela devrait faire réfléchir Microsoft car mon comportement n'a rien d'exceptionnel et représente forcément une tendance non nulle...

Il est vrai qu'une fois la période d'essai terminée j'ai désinstallé ModernMix... Alors ? Tout ça pour ça ?

Il s'avère qu'au final je n'utilise aucun logiciel WinRT, aucun ne me sert, j'ai tout ce qu'il me faut en bureau classique... Même 4 euros finalement, dans mon cas au moins, cela ne les valait pas. Voir la météo dans une fenêtre ? Je le fais déjà avec le browser en allant sur des sites spécialisés. Le reste ? Je m'aperçois que j'ai désinstallé tous les softs WinRT et qu'aucun ne me manque.

Paradoxal ?

Non.

Dans cette guerre fratricide entre deux bureaux, le bureau classique reste le plus essentiel. Je peux vivre sans WinRT pas sans bureau classique. Alors à la fin on se construit un monde pratique où tout est sur le même bureau...

Et justement, c'est bien ce que je veux démontrer ici : si Microsoft veut sauver WinRT il doit l'intégrer au bureau classique sur PC de bureau pour banaliser les applications WinRT et faire en sorte que les utilisateurs apprennent à les apprécier... et que des

développeurs aient envie de se risquer à écrire pour cette plateforme... Actuellement la politique du flingue sur la tempe fait qu'on choisit le bureau le mieux fichu et celui dont ne peut pas se passer : le classique. Ce qui signifie la mort de WinRT à terme.

Conclusion

Je veux que Windows 8 soit apprécié à sa juste valeur. Je désire que WinRT puisse connaître le succès qu'il mérite. WinRT c'est la possibilité unifiée de développer pour PC et pour tablettes, c'est fantastique. **Mais il faut y aller en douceur et dans le respect des choix de l'utilisateur.**

La crise aidant, faire les bons choix technologiques est devenu un casse-tête car tous impliquent des risques financiers (que cela soit en achat de machines, en formation du personnel, des utilisateurs ou des développeurs, etc).

Microsoft a eu cette idée extraordinaire de nous proposer *un OS unifié sur tous les form factors*, hélas, cette démarche a été gâchée par de mauvais choix, par un esprit de fermeture, une stratégie du flingue sur la tempe, des luttes intestines destructrices.

Microsoft semble rester sourd aujourd'hui à ce qui se dit, se pense, c'est vraiment dommage. On voit bien que les grandes nouveautés de Windows Blue semblent s'inscrire dans la suite logique de ce déni comme le nouveau mode de partage 50/50 de l'écran. Ne serait-il pas plus simple de remettre un vrai splitter avec toutes les positions intermédiaires ? C'est quoi ce forçage stupide ? Pourquoi cet aveuglement ? Les plus grands font des erreurs, et plus grosses sont ces dernières plus il est difficile de revenir en arrière sans perdre la face. Il serait dramatique que Microsoft continue d'insister sur une voie qui, de toute évidence, ne mène pas au succès. Il serait grand temps qu'après le départ salutaire de Sinofsky, la direction dans sa totalité soit renouvelée pour qu'enfin les entêtements suicidaires ne soient plus que du passé...

Entre temps, en espérant un renouveau du sommet de la pyramide qui permettra de tout remettre à plat, nous qui savons que technologiquement Microsoft propose ce qui se fait de mieux sur le marché *nous avons le pouvoir d'infléchir ces forçages idiots* qui obèrent sérieusement le futur de ces produits et de Microsoft même en tant qu'acteur de poids sur le marché.

Il ne s'agit pas bien entendu de "vouloir le bien de Microsoft contre son gré", ce que je n'accepte pas pour moi je ne peux l'imposer aux autres !

Non, il s'agit d'être des acteurs actifs de notre propre avenir, d'être des consommateurs adultes, et de dire "non cela ne nous convient pas" et de proposer des solutions dans un esprit ouvert et constructif.

Aujourd'hui utilisez Classic Shell et ModernMix et faites comprendre à Microsoft que c'est cela que doit contenir par défaut Windows Blue, et pas seulement de pouvoir changer la taille des tuiles du menu Modern UI...

Quand on réfléchit avec un esprit ouvert, quand on abandonne les réflexes fascisants, on trouve toujours des solutions qui préservent l'essentiel des produits et surtout qui savent préserver l'essentiel d'une communauté : l'unité face à la concurrence.

Il est temps que Microsoft écoute de nouveau ses plus fidèles alliés, ceux qui connaissent le terrain, ceux qui entendent les clients, les vrais utilisateurs de leurs produits. Le gouffre qui semble se créer entre ces derniers et Microsoft apparaît tous les jours de plus en plus grand, dans l'incompréhension générale.

Déjà, comme je l'indiquais en introduction de ce 600ème billet, l'avenir s'écrira désormais avec plusieurs OS et non plus avec un seul de chez Microsoft. Des batailles ont été perdues, il ne faudrait pas que Microsoft perde définitivement la guerre sur tous les plans.

Microsoft annonce une refonte de Windows 8 historique

Parution : 7 mai 2013

Une annonce d'un second couteau, des fuites d'actionnaires qui demandent la tête de Ballmer, des aveux en demi-teinte d'un échec cuisant. Les américains sont très forts pour les « mea culpa » humiliant en public, les repentir qui tirent des larmes... Mais tout cela n'est que du vent en général et Windows 8.1 et son faux bouton Démarrer a été pris pour une humiliation de la part des consommateurs. Toutes ces promesses de changement verront-elles le jour dans la version suivante, l'après Windows 8.1 ou bien le nouveau raté de cette version démontre-t-il l'autisme de Microsoft et sa volonté de continuer un forcing suicidaire ? Pour l'instant je n'ai aucune information sur la prochaine version de Windows, on reste donc dans l'expectative, comme nous l'étions quand j'ai écrit cet article ! L'annonce du départ de Ballmer n'a pas l'air de changer les choses. Espérons...

On l'attend, on l'espère. Il y a ceux qui ne disent rien et ceux qui osent dire que quelque chose ne va pas avec Windows 8 et WinRT. Finalement parler a du bon sur le silence, Microsoft aurait entendu les reproches et préparerait une refonte...

Une refonte, un "rework" dans le texte

Une refonte est peut-être une traduction audacieuse. Le texte original parle d'un "rework", d'un "re-travail" mais on n'en sait pas plus sur l'ampleur et les détails de ce travail.

Un virage à 180° ?

Les termes pour exprimer ce changement sont "historiques" puisqu'on parle ici d'une remise à plat "la plus grosse pour un produit depuis le revirement New Coke".

Pour la petite histoire, il y a environ 30 ans Coca-cola avait décidé de se rebaptiser "New Coke". Cela a tenu 7 mois avant que le rejet des consommateurs soit si fort que la firme soit obligée d'avouer son échec et de reprendre son nom original.

En informatique, plus récemment, certains d'entre nous se rappelleront de la même erreur effectuée par Borland voulant s'appeler "Inprise" puis reprenant le nom de "Borland" devant l'échec de ce changement d'identité.

Donc le changement qui serait opéré dans Windows 8 serait lié à un échec aussi cuisant que celui de "New Coke", un aveu intéressant, et d'aussi grande importance.

Mais voilà... Ici on parle de nom commercial, Coke, Inprise, etc, de la cosmétique pure. Est-ce que Microsoft, toujours pilotée par la même équipe dirigeante qui a commis toutes les erreurs des dernières années, aura la capacité de se remettre en question en profondeur ? C'est un mystère.

Le terme utilisé en américain est un "U-turn". On peut le traduire par "revirement" ou "virage à 180°", on comprend l'image, un "virage en U", l'américain est une langue très imagée.

Mais ce n'est pas un constat extérieur fait sur des annonces précises, c'est Microsoft qui le dit. Et qu'entendent-ils par là ?

Quel contenu, quelle ampleur ?

C'est la pochette surprise.

Car finalement, pour l'équipe en place, manger son chapeau va déjà être très désagréable. Et dans cette situation on est plutôt tenté de faire de grand discours mais pas forcément de lâcher sur l'essentiel...

La question reste donc de savoir quel contenu sera radicalement modifié et de quelle ampleur sera ce coup de balai. Par exemple WinRT sera-t-il conservé en l'état ou bien deviendra-t-il multi-fenêtré comme il serait nécessaire de la faire ? Le bureau classique sera-t-il remis en valeur ou simplement pourra-t-on démarrer sur ce dernier par une option ? Les langages et plateformes tels que WPF et .NET propres au bureau classique vont-ils être remis à l'honneur, ou bien continuera-t-on à faire croire que WinRT est la solution unique notamment dans les entreprises qui n'en veulent absolument pas ?

Une annonce par la bande

Pour l'instant tout cela reste donc mystérieux.

Mais peut-être est-on enfin sur la bonne voie après tant d'égarements.

Toutefois l'annonce est faite par un second couteau et quelqu'un de très éloigné de la technique puisqu'il s'agit de [propos tenus](#) par Tammy Reller, le chef du marketing

et des finances de MS dans une interview donnée au Financial Times. On aurait préféré une annonce en bonne et due forme par Ballmer dans une revue technique renommée par exemple.

Il s'agit forcément d'une stratégie : on fait annoncer par un type un peu éloigné de la réalité technique un truc renversant, mais sans aucune précision, et on attend de voir les réactions. Si ça plait, alors le big boss peut se mouiller, si ça ne plait pas, on ignorera cette parole qui n'engage personne.

Conclusion

On sent frémir l'aveu d'un échec cuisant. Celui de la stratégie, du design, de l'UX, de tout ce qui vient de Sinofksy.

Microsoft a perdu beaucoup de temps et de crédibilité. Ça serait vraiment une bonne nouvelle si ce revirement, ce "U-turn", était franc et massif et qu'il apportait enfin de la visibilité et surtout un véritable avenir pour Windows 8 et WinRT qui ont beaucoup à offrir pour peu qu'on les adapte à la demande.

Attendons la suite...

WPF : Et si on en reparlait sérieusement !

Publication : 29 juin 2013

L'adoption de WinRT étant ce qu'elle est, et bien que l'avenir ne soit pas écrit, une réalité s'impose : les entreprises ont massivement migré leur parc en Windows 7 avant d'être obligées d'acheter des licences Windows 8. Rien ne sert de raconter des histoires à ces clients là... WinRT ne tourne pas sur Windows 7. Silverlight a été arrêté et peut être utilisé encore un an ou deux puisqu'il sera maintenu jusqu'en 2021 mais pour le reste ? Si pour le Web ASP.NET reste une excellente alternative, pour le bureau et les « vraies » applications, WPF qui lui n'a pas été arrêté (une nouvelle version serait même en préparation) pourrait fort bien devenir, enfin, la plateforme des logiciels de demain. Et si on en reparlait sérieusement ?

Windows 8, 8.1, des tuiles, WinRT, tout cela serait l'avenir. C'est une possibilité et la plateforme est séduisante. Mais elle n'a pas conquis les entreprises pour l'instant. Pourtant il faut bien développer de nouvelles applications. Et si nous reparlions sérieusement de WPF et de l'état du marché ?

WPF L'éternel oublié mais l'inoubliable aussi...

Qui se rappelle de ce que veut dire WPF ? Je vous laisse 2 secondes...

Windows Presentation Foundation. La fondation du système de présentation de Windows. Rien de moins !

Nom de code "Avalon", vous vous rappelez ?

WPF instaure et fonde les spécifications graphiques de .NET 3.0 (novembre 2006). Il est livré pré-installé avec Vista. Il est même possible de l'installer sur XP à partir du Service Pack 2 de cet OS et même sur Windows Server 2003.

Les débuts difficiles de Vista couperont la dynamique de lancement de WPF, beaucoup d'entreprises continuant à utiliser Windows Forms par inertie. Mais Vista quoi qu'on en pense vient de tout changer : il annonce l'ère des designers, une étape essentielle en informatique professionnelle : l'aspect devient un enjeu aussi important que le contenu. Le terme d'UX viendra plus tard, mais le mouvement est lancé... Le clou sera enfoncé par le succès fulgurant de l'iPhone et de l'iPad puis d'Android.

Puis Silverlight sortira et éclipsera par son succès WPF devenu pourtant mature et doté d'une puissance incroyable et d'outils novateurs comme Blend. Toutefois ce succès sera celui d'une équipe : C# + XAML quel que soit le nom qu'on lui donne, une équipe présente depuis le début et qui se retrouve sous Windows Phone ou WinRT aujourd'hui. En tout cas WPF en tant que tel ne profitera qu'à la marge du succès de Silverlight ce dernier étant bien plus séduisant et léger que XBap notamment. Mais l'élastique se tend... Les entreprises sont alors toujours en XP et le grand public s'excite déjà sur les premiers modèles d'iPhone. L'écart devient gigantesque entre applications métier tristes au look XP et applications super lookées pour l'homme de la rue... Seulement l'homme de la rue est aussi le salarié qui utilise des applications métiers. La tension monte... et les attentes se font de plus en plus pressantes au risque de déstabiliser la sacro-sainte productivité des entreprises !

La fin de Silverlight sur laquelle nous ne reviendrons pas ici n'aura pas été l'occasion d'un vrai regain d'intérêt pour WPF car au même moment Microsoft décrétait que .NET et tout ce qui tourne autour était devenu "hasbeen", seul WinRT pouvait être "moderne". Au lieu de proposer une nouvelle voie en poussant intelligemment son pion, Microsoft s'est tiré une balle dans le pied. Ringardisant toutes leurs propres technologies matures et misant tout comme un fou le ferait à une table de poker sur un éventuel prochain coup gagnant... Pas de "plan B", aucune prudence. Ballmer a parfois des allures d'illuminé quand il harangue une salle, mais force est de constater qu'il conduit aussi Microsoft comme un illuminé. C'est affreusement triste au regard de la beauté et de l'intelligence des technologies créées par les ingénieurs de MS.

A chaque fois en avance sur son temps, à chaque fois brillant et unique en son genre, le pauvre WPF a été étouffé par telle ou telle autre annonce ou rumeur. Jamais il n'eut véritablement son heure de gloire. Seule plateforme totalement vectorielle, animée, réactive, WPF n'a pas été adulé comme il le mérite.

Et pourtant !

Alors que les entreprises sont en train de migrer vers Windows 7, soit "Vista 2", **WPF n'a jamais été autant d'actualité ! Avalon a été créé pour Vista et Windows 7 est un Vista bien plus qu'un Windows 8.**

WPF est le seul environnement moderne pour développer pour _toutes_ les versions en activité de Windows et donc pour tous les Windows 7 en cours d'installation qui seront là pour des années !

Windows un socle fissuré ?

Les échecs de Vista et Windows 8 cumulés pèsent lourds sur le marché car ils s'étalent sur une longue période créant beaucoup d'incertitude, une perte de confiance et de l'immobilisme, ce n'est bon pour personne.

Windows 7, très bon techniquement mais un simple 'Vista 2' sans grande originalité (tout avait été dit et fait dans Vista) en vient même à jouer la planche de salut pour les entreprises qui en ce moment continuent des migrations vers cet OS.

Windows qui a été le socle de l'avènement de l'informatique dite "micro" et de la démocratisation du PC jusque dans le salon de la mamie du Cantal, a cédé du terrain. C'est historique. Pour la première fois depuis son avènement les parts de marché de Windows s'effritent.

Le socle se fissure.

Une fissure n'annonce pas forcément la mort du bâtiment, combien de maisons tiennent toujours debout et sont habitées malgré quelques fissures dans leurs vieux murs... Mais tout comme les rides sur le visage de celui qui prend de l'âge, les fissures d'une maison annoncent plus la fin que le début, même si cette fin n'arrivera que dans longtemps.

En mai 2013 les parts de marché de Windows (toutes versions) étaient de 80.68% alors qu'en juillet 2012, soit seulement dix mois avant, elles étaient de 83.55% soit une baisse de 2.87% ce qui est à la fois peu et considérable dans ce que cela signifie.

A qui profite le crime ? Assez peu à iOS qui dans la même période passe de 5.90% à 6.50% soit 0.6% de hausse, quasiment pas du tout au Mac qui reste stable (6.23% à 6.49%), très peu à Java ME (0.82% à 1.21%) et un peu plus à Android qui prend presque 1% de 1.81% à 2.78%. Les "autres" profitent aussi de la situation en passant de 1.69% à 2.33% (chiffres [Net market Share](#)).

On voit clairement que dans la fissure Windows s'engouffrent surtout les OS mobiles et principalement Android. Mais tout le monde en profite, _aucun autre_ OS ne baisse, seul Windows cède du terrain. Il s'agit donc bien d'un problème spécifique à cet OS et non d'une tendance du marché plus ou moins globale, d'un marasme ou de la "crise" qui a bon dos quand il s'agit de trouver des excuses à de piètres performances commerciales...

Fissuré mais majoritaire pour longtemps !

La grande maison Windows est fissurée, elle prend un peu l'eau par mauvais temps, mais elle tient debout et surtout elle occupe toujours son "bout" de terrain : 80.68% des parts de marché global de tous les OS c'est énorme, un sacré "bout" de terrain !

Certes, avec des pertes de près de 3% par an le lent déclin pourrait devenir rapidement une bérézina si cette tendance devait se confirmer. Mais on en est encore loin, de nombreuses choses peuvent se passer sur le marché dans les années à venir, ne jouons ni les cassandre ni les oiseaux de mauvais augure. La situation est préoccupante mais plus de 80% du marché est une position de roi voire d'empereur que tout le monde envierait !

Même si le rythme de 3% annuel de baisse de parts de marché devait se poursuivre durant les 16 prochaines années, il resterait à Windows 49% du marché face à des tas d'OS qui se partageraient le reste et donc seraient loin d'une telle domination... 16 ans c'est sacrément long ! Bien plus qu'une implémentation d'application métier en tout cas (l'application peut vivre plus longtemps encore en tant que concept, mais les implémentations ne peuvent rarement survivre plus de 5 ans sans refonte drastique).

Cela signifie qu'il faut être réaliste, ni défaitiste en voyant la mort de Windows ou de Microsoft pour demain, ni stupidement enthousiaste en jouant les groupies d'un Windows 8 ou Windows Phone qui ne savent pas prendre de place significative sur le marché.

Personnellement je n'ai pas une nature de "fan", ni en sport, ni en musique, et encore moins dans mon métier. J'agis rationnellement et de façon constante avec une approche pragmatique des faits.

Donc, dans un tel marché qui ne passionne plus, sauf exceptions, dans un environnement social dégradé par les crises successives qui comme les banlieues qui s'étendent au point de se confondre donnent l'impression d'une seule et même interminable entité glauque, il est impératif de s'en tenir à des constats factuels. Seuls points de repères fiables, comme les oasis qui mouchètent les cartes du désert et permettent de tracer des routes au milieu d'un grand rien, *les faits, rien que les faits, permettent de viser juste et de jalonner le futur d'étapes réalistes.*

Et les chiffres sont têtus, Windows est là, bien là, et pour longtemps.

Windows va durer, mais quel Windows ?

Le raz de marée Metro / Modern UI / WinRT / Windows 8 (quel nom donner ?) n'a pas eu lieu. La mise à jour 8.1 s'annonce décevante car Microsoft n'a pas vraiment entendu les utilisateurs ni les entreprises. Le forcing maladroit WinRT / Modern UI continue. Mais soyons francs, le flingue sur la tempe n'a jamais été une manière durable de séduire, c'est une technique de violeur et pas de Don Juan. On obtempère pour sauver sa peau, mais ce n'est pas de l'amour...

Remettre un bouton "démarrer" qui renvoie au menu à tuile est presque une injure pour l'intelligence des utilisateurs, c'était déjà le fonctionnement de Windows 8 (avec le bouton Windows du clavier mais bon...) et c'est toujours moins bien que ce que Classic Shell ou d'autres font gratuitement (dont le boot en bureau classique) ! Quant à la possibilité de partager l'écran en 3 au lieu de 2, franchement quand on s'appelle "Windows" cela fait de la peine de faire moins bien que Windows 1.0 (même principe de non recouvrement mais au moins le nombre et la taille des fenêtres étaient à la discrétion de l'utilisateur). Pour ce qui est des entreprises, deux ou trois ajouts sur la sécurité ou la possibilité de vider un disque dur à distance ne révolutionneront rien.

Le grand public est coincé : malgré les lois interdisant cette pratique, tout nouveau PC est fourni de force avec Windows pré-installé (et grassement payé mais sans CD ni possibilité de l'installer ailleurs dans le respect de la licence). Donc, mécaniquement tous les nouveaux PC étant en Windows 8, les parts de marché de cet OS vont grandir. Certains spécialistes des prévisions prévoient 25% de marché d'ici quelques années et certainement beaucoup moins en entreprise.

A l'heure actuelle comment se répartit le marché Windows ?

En gros 57% du marché global est constitué de Windows 7 qui a gagné plus de 10% en un an. XP disparaît progressivement passant de 32% à 16% sur la même période. Vista était déjà anecdotique il y a un an et l'est encore plus aujourd'hui (de 4.8% à 2.10% soit une baisse de 50% environ). Windows 8 se place à moins de 8%.

L'évolution à venir devrait tendre pour 3/4 de Windows 7 et 1/4 de Windows 8, avec des XP et des Vista résiduels.

Dans un tel marché, surtout en entreprise, il n'est pas possible de développer en WinRT puisqu'à la différence de WPF qui peut s'installer même sur un XP SP2, WinRT ne marche que sous Windows 8. Pas de compatibilité avec le parc existant.

L'avenir : WPF, en toute logique

On le voit clairement, qu'il s'agisse de la situation actuelle ou à venir WinRT ne peut pas être utilisé pour développer en entreprise car sa pénétration est trop faible. Windows 7 vient remplacer XP et cela pour plusieurs années.

Or, en toute logique, seul WPF permet de programmer des applications compatibles avec le parc existant et celui à venir pour des années.

Passéisme, WinRT-bashing ou réalisme ?

WinRT est une belle plateforme très intéressante.

On peut même en entreprise envisager des développements ponctuels pour Surface car utiliser des applications existantes sur un écran tactile d'une dizaine de pouces est

une stupidité. Les applications Windows classiques sur de si petits écrans ne sont raisonnablement pas utilisables.

Mais on peut en revanche concevoir des applications WPF au look Metro adaptées aux petits écrans. C'est une alternative tout à fait convenable et parfaitement adaptée aux Surfaces "Pro" ayant un bureau classique puisque Microsoft s'est encore une fois tiré une balle dans le pied en sortant trop vite la "Pro" qui a tué tout espoir de succès pour la RT.

Il n'y a donc pas de "WinRT bashing", professionnellement il ne s'agit pas d'aimer ou non, il faut uniquement viser juste et cela se fait en prenant le marché tel qu'il est et tel qu'il va évoluer et non comme Microsoft le fantasme. A chacun ses préoccupations, on comprend fort bien que MS pousse ses produits, mais nous, développeurs ou DSI, devons prendre des décisions fondées sur la réalité.

Promouvoir WPF aujourd'hui est-il une forme de passéisme ?

Non plus. WPF c'est XAML et le plus souvent C# et qui oserait penser que ces langages sont "dépassés". Windows 8 intègre de base le moteur XAML, ce n'est même plus une option installable à part ou découplée de l'OS, c'est l'OS, on est donc en plein dans le futur !

Quant à C#, toujours dans l'innovation et à la pointe des besoins des développeurs, qui pourrait voir le moindre signe de déclin ou de faiblesse ? Face à l'anachronisme d'un C++ mort depuis des années, à côté de VB qui n'a jamais été le choix majoritaire des développeurs malgré un succès non négligeable, face à un F# sympathique mais follement essayiste, C# s'impose comme le choix de la raison.

Il n'y a donc rien de passéiste à promouvoir une plateforme qui est toujours en avance sur tout ce qui se fait ailleurs et même chez Microsoft !

WPF est ainsi un choix de raison mais aussi un choix d'avenir et surtout le choix de la compatibilité sans renoncer à satisfaire les attentes des utilisateurs en termes d'UX.

Conclusion

Le temps des atermoiements doit prendre fin un jour où l'autre... Les DSI ont fait le dos rond en laissant passer l'orage du doute et des angoisses.

Seulement l'orage ne se termine pas. Tout le monde aurait aimé que Windows 8 soit un succès éclatant, que WinRT séduisent tous les développeurs, que les gens s'arrachent Surface RT et que les Windows Phone envahissent le monde entier.

La position "dominante" de Microsoft ? Honnêtement les professionnels de l'informatique s'en contre-fiche tout comme les DSI en entreprise. Au contraire ! Cette domination a permis l'explosion de la micro-informatique et a autorisé l'avènement d'une informatique d'entreprise ultra-performante. La stabilité d'une domination, c'est comme la paix, c'est bon pour le business !

Le chaos du marché actuel, les erreurs de Microsoft, tout cela est nuisible pour tous. On ne peut franchement pas s'en réjouir.

Mais il nous faut être réalistes : *les entreprises doivent aujourd'hui encore plus qu'hier miser sur leur informatique pour être plus performantes, plus réactives, plus agressives, plus rentables.*

Il est temps que les DSI comprennent que la stratégie du "dos rond" est terminée. Que *dans le contexte économique actuel l'attentisme est un suicide.*

Et pour développer pour le parc de machines actuel et à venir tout en assurant une modernité indispensable à l'adoption des logiciels par les utilisateurs, si vous connaissez un meilleur choix que WPF, n'hésitez pas à m'en faire part !

WPF : Retour sur une véritable avancée

Publication : 22 juillet 2013

Après avoir rappelé et démontré que WPF est une plateforme d'avenir, la seule en entreprise au moins, il m'a semblé intéressant de reprendre de vieilles publicités WPF pour vous montrer que déjà il y a quelques années, cette plateforme avait séduit de grands noms, ce que WinRT n'a pas réussi à faire à ce jour, et que déjà les applications WPF de cette époque avait dix ans d'avance en terme d'UI et d'UX. Imaginez ce que vous pouvez en faire aujourd'hui avec la version 4.5, et toutes les avancées en termes de Design qui sont intervenues... Tour d'horizon salutaire pour enfoncer le clou d'un choix de la raison qui est en accord avec celui du cœur !

J'ai beaucoup parlé dans le passé, j'en parle encore. WPF est la technologie de création d'interfaces visuelles la plus aboutie et la plus mûre du marché. Revenons sur quelques réalisations qui en démontrent la modernité absolue.

UX et UI

Une bonne UI est le point d'orgue d'une bonne UX. Si l'UI ne fait pas tout, une mauvaise UI mènera directement à une mauvaise UX.

Et pour réussir une bonne UI il faut les bons outils, ceux qui libèrent la créativité sans la brider dans des normes limitatives et parfois très pauvres, des outils qui savent aussi se plier aux exigences du fonctionnel.

Cet outil existe depuis un bon moment, et il est capable de produire des exécutables qui tournent sur tout le parc Windows installé !

Oui, je l'affirme, WPF est ce que Microsoft a fait de mieux dans les 25 dernières années. Cela englobe XAML et C# qui compte beaucoup dans ce mariage d'exception, sans oublier .NET qui est la plateforme la plus pratique à utiliser tout en étant complète, compréhensible et totalement objet.

Des exemples

Microsoft avait publié un très joli catalogue virtuel de réalisations WPF pour en démontrer le potentiel et **surtout sa parfaite adéquation avec les besoins des entreprises et des utilisateurs.**

Mike Taulty l'avait mis en ligne durant l'été 2009, heureusement car ce document est devenu introuvable chez Microsoft, la purge stalinienne imposée par Sinofsky étant passée par là...

Rendez visite à cette page pour voir pourquoi WPF est toujours d'actualité : <http://www.mtaulty.com/downloads/WpfShowcase/test.html>

Et voici une petite compilation des UI les meilleures qu'on peut y trouver (en sachant que depuis de nombreuses autres ont été créées partout dans le monde !). Ne vous sentez donc pas "seul" si vous prônez WPF dans votre société....

LINE OF BUSINESS INFORMATION WORKER

Lawson
Lawson Smart Office

The Lawson Smart Office leverages WPF to deliver a user experience that is unequalled in the ERP space in terms of its beauty and usability. The Lawson Smart Office delivers on the needs and expectations of a future generation of ERP users.

"WPF allows us to build visually rich applications and still maintain a clear separation between the user interface and the business logic."

— Matthew A. Bee, Director - User Productivity Platform, Lawson Software

Accele
Accele Mobile Office

Complete mobile office application for state and local government agencies. It helps employees manage field tasks like inspections and code enforcements.

"The product provides a robust and intuitive solution that greatly enhances the productivity of our users."

— Maury Backman, President & CEO, Accele

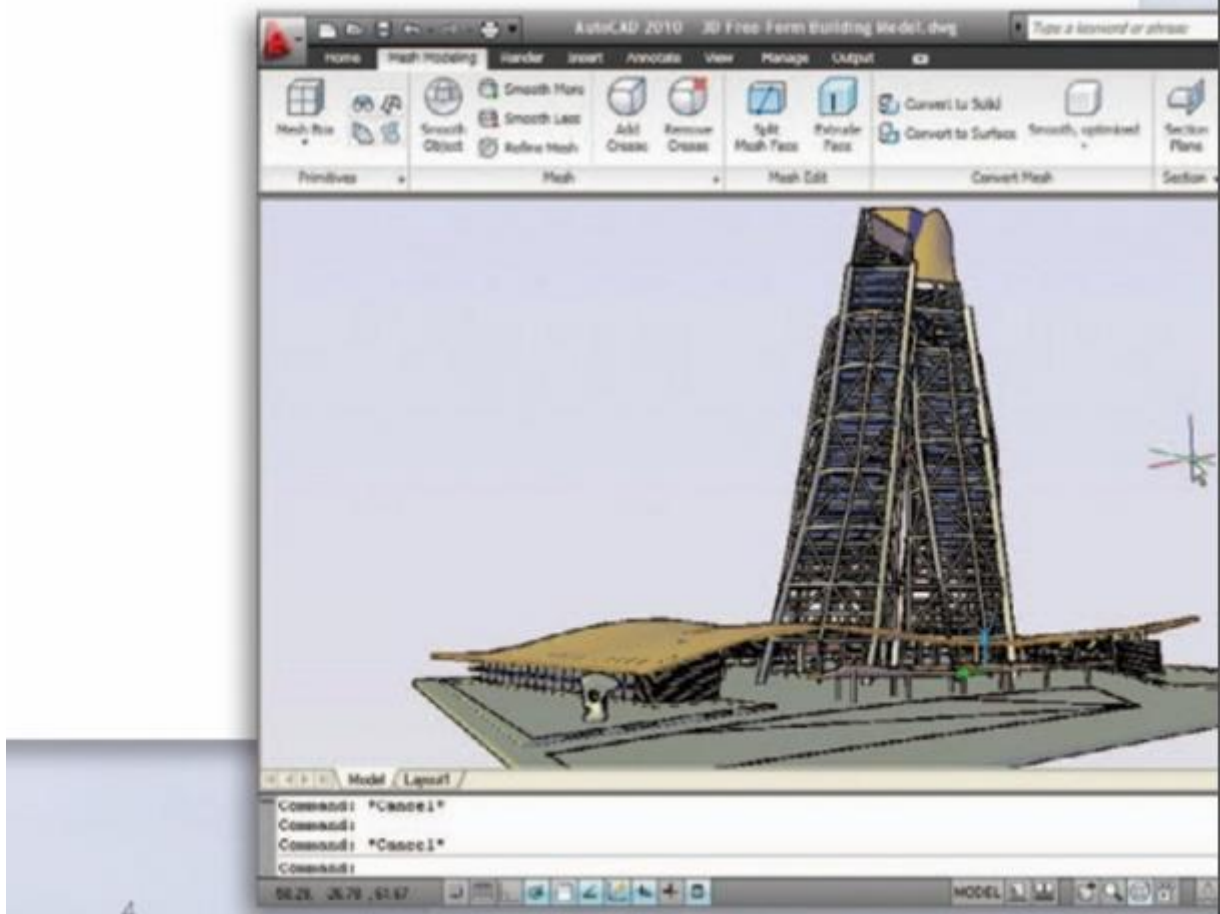
ERP, Géolocalisation, cartographie, des besoins toujours aussi essentiels !

Autodesk AutoCAD 2010

AutoCAD is one of the world's leading applications for 2D and 3D CAD. AutoCAD helps users speed documentation, share ideas seamlessly, explore ideas more intuitively in 3D, and customize programs for their specific needs so they can take design further.

"The new ribbon-based User Interface in AutoCAD makes features more discoverable. AutoCAD customers are increasing their productivity by having the right tool available at the right time and in many cases, customizing the UI so they can tailor AutoCAD to fit their particular project needs."

—Eric Stover, AutoCAD Product Line Manager, Autodesk, Inc.



Dessin technique, illustrations complexes, 2D et 3D interactives...

DIGITAL MEDIA CREATION

Roxio

Roxio Creator 2009

Roxio Creator 2009 is the only complete digital media software that lets family members of every age and experience level to easily create, preserve, and share their stories and passions.

"Using Windows Presentation Foundation in redesigning our flagship product allowed us to create an exciting, visually rich application that bolsters our leadership position in the digital media market."

— Vito Salvaggio, VP of Consumer Products



Création de média numériques, gestion de média, peut-on trouver préoccupation plus "moderne" ?

Lenovo ThinkVantage

Portfolio of IT management tools to manage every stage of the PC life cycle.

"The goal of Power Manager is to provide the advanced power management settings in the simplest manner. WPF helps to develop attractive yet simple user interface very fast, WPF also supports all our multi-language requirements, making localization very easy."



Gestion des devices d'un ordinateur, intégration avec le hardware car l'accès à ce hardware est possible et non pas limité par une sandbox ou des certificats au prix exorbitant...

HP
HP Total Care Advisor

HP Advisor helps make your PC easier to use –view your status at a glance, access support, discover more ways to use your PC (edit photos, create video, manage music, etc.), and navigate easily to your favorite software and services.

"With WPF, we develop new Advisor functionalities very rapidly. WPF allows our developers to more easily and quickly match our designer's creativity. The ease of using WPF enables to develop new and innovative features, while managing ongoing release and maintenance to our broad install base."

– Diane Weiss, Product Manager Experience Software, HP

Apporter un réel soutien aux utilisateurs, c'est la base même de l'informatique d'aujourd'hui et de demain...

Intel

Intel Rapid Storage Technology



Intel® Rapid Storage Technology user interface allows users to easily manage intel® desktop and mobile platforms equipped with SATA disks.

"WPF impressed our developers with its power and flexibility. We had a tight schedule and using WPF made it easier to achieve our goals and a high deliver quality, visually appealing interface with enhanced usability."

– Susan Foster, Intel Corporation.

AMD

AMD Fusion Media Explorer


AMD Fusion Media Explorer is a 3D immersive social media and digital media browser

"There is no better (and faster) way to navigate your media library, than our fluid, 3D, ribbon."

– Casey Gotcher, Director of Product Marketing, AMD.



Intel, AMD comme HP, Lenovo, Roxio, BMW, Dassault (système CATIA), ... tous ont fait confiance à WPF et aucun ne le regrette car toutes ces applications fonctionnent toujours sur le parc de machines en 2013 et fonctionneront encore longtemps, loin des modes et des caprices des OS à double tête...



The screenshot displays the AirStrip patient monitoring interface. At the top, it shows the patient's name 'RAY, DEBBIE' and 'SU802' with an 'Incision Time 6:15 AM'. The interface is divided into several sections:

- Patient Views:** A sidebar on the left lists various data sources like Vitals, ECG, PACE Images, Blood Pressure, Chart Event Table, Video, Electronic Records, and Lab Data.
- Patient Status:** A section on the left provides patient details: Name (Debbie Ray), Gender (Female), Age (37), Date of Birth (6/22/76), SSN (xxxx-7124), Doctor (Dr. Wallace), Emergency Contact (Kurt Ray), and Emergency Phone Number (254544-9025).
- Vital Signs:** The main area shows a heart rate of 72 and respiratory rate of 18. Below these are waveforms for ECG and respiratory patterns.
- Medical Data:** There are sections for 'PACE, Last Image (all) - MRI, 7 Images, 3/15/09' and 'Video Feed, VIDEO 8' showing a surgical procedure.
- Temperature and Blood Pressure:** Two small charts at the bottom show temperature trends and blood pressure readings over time.

AirStrip
AirStrip

Console for monitoring real-time patient data which is intuitive, highly discoverable and unifies a powerful set of decision making tools.

"Windows Presentations Foundation makes this type of dynamic, modular user interface possible, while also delivering far more intuitive data visualizations."

—Cameron Powell, MD, Airstrip Technologies

Affichage temps réel de données vitales, fiabilité, élégance, puissance, WPF peut être utilisé dans tous les cas de figure.

Hong Kong Disney

Hong Kong Disney Park Explorer

Windows gadget that allows you to virtually navigate the Disney park and plan a fun vacation.

"Advertisers see the branded gadgets as another opportunity to engage with customers. Your brand is front and center with them."

—Robbie Bach, Vice President, Microsoft E&D Division



T-Online

Meine Software

Suite of consumer applications that brings the online information you care about to your desktop. Includes weather, time, news, search and other service. Integrates with Windows Sidebar for one-click access from the desktop.

Jeux, graphismes modernes et séducteurs, réactivité, liberté de s'adapter au contexte et non pas obligation d'écrire des logiciels contraints par le contexte technique...

Nokia Music
Nokia Music

Lets you search, organize and listen to your music collection in one easy place. Allows you to discover and purchase new music from the Nokia music store.

Equalizer settings
Flat

12 44 175 290 500 3K 6K 8K 16K

Une époque où même Nokia savait encore faire les bons choix technologiques !

TOUCH & SURFACE

HP

HP TouchSmart

Companion Software for HP TouchSmart PCs. The software brings the digital world to your fingerprints: simplifying access to photos, videos, music, and other applications.

"Windows Presentation Foundation provided HP with the basis for building experience oriented applications in a timely manner for HP's flagship TouchSmart PC. The unique ability to blend together applications, UI and media content provided a tremendous advantage to HP in terms of creativity and time to market"

– Paul Rygaard, Director Consumer PC SW & Services



Le Tactile en voilà une préoccupation moderne s'il en est ! Alors n'oubliez jamais que Surface, la vraie, la grosse, la première, était programmable en WPF ! Qui peut le plus, peut le moins...

Conclusion

C'est vrai, certains look "dark" ne sont plus dans la mode dépouillée de l'instant, mais changer une ressource couleur en Xaml quand il est bien écrit cela prend quelques secondes sous Blend ! J'ai d'ailleurs produit (et mis en téléchargement gratuit) il y a quelques temps une simulation du look Metro pour faire des menus à tuiles ou des transitions de page à la Metro... WPF se plie à toutes les exigences.

Certes ce petit voyage qui nous a ramené en 2009 peut sembler un peu trop nostalgique. Mais en matière de technologie il n'y a pas de nostalgie à avoir : ça fait le job ou ça ne le fait pas. **Et WPF faisait le job et le fait toujours !** Demandez à tous ceux dans le monde qui utilise Java s'ils ne se sentent pas hasbeen d'utiliser un langage presque aussi vieux que Delphi ! Non. Car ça faisait le job à l'époque et ça le fait toujours. WPF aussi.

Etre moderne ce n'est pas faire table rase de tout même de ce qui est bon pour le seul plaisir d'avoir quelque chose à dire, pour faire de la com' ... **Être moderne c'est innover et pour cela on utilise souvent des outils matures.** Même Picasso utilisait le même genre de chevalet que Da Vinci pour poser ses toiles et les peindre ! On peut être en avance sur son temps avec des idées radicalement neuves, cela ne veut pas dire, et bien au contraire, renier les outils qui ont fait leurs preuves ! WPF est le chevalet de vos futurs logiciels, pensez-y !

WPF reste la plateforme absolue, la plus aboutie de l'industrie informatique. Et pour se confronter au parc informatique de demain constitué à plus 70% de machines Windows 7, je vois mal quelle autre plateforme pourrait apporter plus de souplesse et de portabilité. Plus récent il n'y a pas, plus vieux c'est trop antique (comme Windows Forms par exemple)...

Pensez-y...

BYOD, CYOD, MAM, MDM, MIM vive les unités mobiles !

Publication : 17 septembre 2013

Mettre un peu d'ordre et de raison parmi tous les modes de développement et de travail qui impliquent de plus en plus d'intégrer les unités mobiles dans les entreprises, voilà une noble cause défendue par cet article. Acronymes étranges, bonnes idées et moins bonnes se côtoient pour que vous sachiez où mettre vos pieds !

Dans la continuité des billets consacrés au développement cross-plateforme il est important de se questionner sur la gestion des smartphones et tablettes en entreprise. Le BYOD et le CYOD sont-ils des réponses aux besoins conjoints des entreprises et de leurs salariés ? Quels sont les possibilités, les risques et les solutions ? Qu'est-ce que le MDM, le MAM et le MIM ?

Introduction

Loin des billets consacrés au développement lui-même de solutions cross-plateformes je vous propose aujourd'hui de réfléchir à une autre facette du cross-plateforme : l'intégration des machines mobiles dans l'entreprise.

Au même titre qu'une bonne application ne peut se contenter d'un bon code, elle a aussi besoin d'une bonne UI et d'une bonne UX, l'intégration des machines mobiles, smartphones et tablettes, dans l'entreprise ne saurait se résumer à quelques lignes de code. Cela réclame un plan réfléchi tout autant au niveau technique qu'au niveau légal.

Les points clés d'une intégration en entreprise

La gestion des périphériques

La gestion des périphériques est un facteur clé qui va affecter combien votre entreprise devra investir pour le développement des applications et leur suivi. Notamment comment allez-vous suivre les périphériques au sein de la société ? Et lorsqu'ils sont utilisés à l'extérieur de celle-ci ? Combien de fois aurez-vous à remplacer les téléphones ou tablettes et en conséquence mettre à niveau vos applications pour soutenir les nouveaux modèles et leurs OS ? Comment gérer les changements de versions parfois incompatibles avec le matériel (Windows Phone 7 vers WP 8 en a été un exemple) ? Comment gérer les environnements de développement (l'émulateur iPhone ne tourne que sur Mac. Il faut en prévoir l'achat et la maintenance) ? Les appareils eux-mêmes peuvent représenter un investissement important toutefois celui-ci sera vite négligeable comparé aux frais de développement, de maintenance et d'évolution des logiciels si la gestion elle-même de ces appareils n'a pas été réfléchi et planifiée. Si ces questions ne sont pas abordées et réglées en amont vous découvrirez à l'usage toute une série de couts cachés qui grèveront dangereusement votre ligne budgétaire !

BYOD, CYOD ou achat ?

Nous avons vu monter la tendance du Bring Your Own Device (BYOD) ces derniers temps au sein des entreprises. Combien de fois avez-vous pu constater lors de réunions en interne ou chez des clients que chacun tapotait ou regardait sur son iPad personnel, son iPhone, son téléphone Android ou Windows Phone ? La question est de savoir si vous pouvez vous permettre de soutenir une logique de libre choix des machines au sein de votre entreprise ! Cela est très lourd à prendre en charge même si vous ne soutenez que la version actuelle de chaque plate-forme. Que devez-vous privilégier : aller vers les appareils les plus populaires ou devez-vous choisir une plate-forme qui correspond le mieux à votre écosystème actuel et distribuer des appareils appartenant à l'entreprise ? Il n'y a pas de réponse facile ici, mais vous devrez avoir trouvé votre réponse avant de vous lancer. Pour cela vous devez pouvoir chiffrer à la fois les couts de développement liés à la couverture des plateformes, leurs évolutions et les couts liés au matériel (assurance, réparation, vol...).

Trois options sont ouvertes, l'achat et la distribution de matériel, le BYOD (chacun amène son appareil personnel), et le CYOD (Chose your own device) où l'entreprise sélectionne une gamme limitée de machines parmi laquelle les salariés font leur choix.

L'achat

L'achat reste la solution la plus simple et de loin la plus économique : le matériel ne vaut rien comparé aux coûts de développement et de maintenance. La journée d'un ingénieur peut revenir au prix d'un ou plusieurs téléphones haut de gamme... Imposer un modèle, un OS, ne soutenir qu'une solution semble le plus pratique. Sauf que cela ne peut s'accorder avec les usages des utilisateurs qui aujourd'hui dans un marché mature possèdent déjà leur propre équipement.

S'il semble naturel que l'entreprise impose au bureau le choix d'un PC et de son OS, cela devient tout autre chose lorsqu'il s'agit d'un smartphone par exemple. C'est une machine "personnelle" bien plus qu'un PC, plus que personnelle, elle est "intime".

Si l'entreprise impose une machine fermée, l'employé utilisera à côté son propre matériel. C'est un choix acceptable mais peu pratique. Par exemple, même si cela frôle l'illégalité dans certains cas, l'entreprise aime bien pouvoir joindre certains salariés, surtout les cadres, le soir ou le week-end... Si le smartphone personnel est différent de celui de l'entreprise, ce dernier sera rangé hors des heures de travail. Joindre la personne "professionnellement mais hors des heures de bureau" deviendra impossible. Il sera plus facile pour le salarié de se soustraire à de telles communications. Si le salarié ne dispose que d'un seul appareil, on le joindra partout même en vacances... Il ne s'agit pas ici de favoriser de telles utilisations extrêmes proches de la tyrannie ou du harcèlement. Mais dans le cadre d'une utilisation "normale" motivée par des raisons valables (décalage d'un rendez-vous en dernière minute, information essentielle à communiquer) il est préférable que le smartphone personnel et professionnel ne fasse qu'un... Les tablettes posent d'autres problèmes similaires mais pas identiques.

BYOD

Le BYOD est entré doucement en entreprise. Doucement car il n'est pas une panacée non plus. Principalement parce que l'entreprise accepte dès le départ le foisonnement des OS, des form factors, etc, ce qui va coûter une véritable fortune en développement et en gestion de parc. Cela répond aux problèmes soulevés par l'achat pur et simple évoqué plus haut, on est sûr que le salarié aura l'appareil avec lui partout et tout le temps ce qui permettra de le joindre et de lui communiquer des informations à tout moment. Mais à quel prix ! Offrir un smartphone coûte bien moins cher.

Lorsque le marché du smartphone était dominé par l'iPhone, il était facile de faire du BYOD car en réalité les gens (surtout les cadres concernés par ces opérations) avaient

tous le même téléphone, la même tablette... Mais la chute d'Apple (13,5% du marché à ce jour), l'émergence de Google (plus de 80% du marché), le maintien même à un niveau modeste de Windows Phone (de 3,5% à 8% selon les études), tout cela rend le BYOD un véritable enfer à gérer. A moins d'être une entreprise d'envergure et de disposer d'une grosse équipe d'informaticiens en interne, le BYOD ne semble plus être une solution raisonnable aujourd'hui.

CYOD

Le CYOD tente de couper la poire en deux. Comme toute solution intermédiaire elle a des avantages mais aussi l'inconvénient de nager dans la zone de "ni-ni"... Ici l'entreprise fait un choix de matériel et d'OS et elle propose au salarié de choisir celui qu'il préfère. En général le CYOD est une offre donnant-donnant. Le salarié peut choisir un modèle plus luxueux, mais sa participation à l'achat sera plus importante. En laissant cette liberté on s'assure que le matériel sera bien celui utilisé tout le temps par le salarié (il sera donc joignable facilement), en même temps le salarié y trouve son compte puisque l'achat étant subventionné par l'entreprise cela lui permettra d'acquérir un matériel proche de ses désirs pour un investissement moins grand. L'entreprise ne pouvant forcer un salarié à payer son matériel, le CYOD intègre forcément un "plan de base" avec un modèle totalement gratuit. Forcément moins luxueux ce modèle de base ne devra toutefois pas faire peser une contrainte encore plus couteuse sur le service de développement !

Pour l'entreprise le choix du CYOD simplifie la gestion du parc de machines, abaisse les coûts de développement et de maintenance des logiciels par rapport au BYOD. Subventionner un smartphone ou une tablette n'est pas un coût insurmontable face aux économies de fonctionnement. Le choix se fera d'ailleurs plus sur une série d'OS et de form factors que sur une marque. Si Apple ou Windows Phone ne proposent que peu de choix, l'offre Android est pléthorique (d'où son succès d'ailleurs). Mais le CYOD ne règle pas tous les problèmes. Il y aura toujours des salariés ayant leur propre matériel car quand on embauche quelqu'un aujourd'hui il y a fort à parier qu'il possède déjà une machine... Comme je le disais le CYOD se situe dans une zone de "ni-ni".

Le suivi des machines

Quelle que soit la solution adoptée il faudra suivre les machines... Ce suivi peut s'entendre sur le plan matériel (dépannage, remplacement, machines en prêt pendant une réparation...) autant que sur le plan logiciel.

Les smartphones et les tablettes sont des machines faciles à voler, bien plus qu'un portable avec sa sacoche, son alimentation, sa souris. Ces machines, surtout les smartphones, sont aussi plus faciles à perdre.

Dans le cas du BYOD l'entreprise se lave les mains du matériel. C'est en cela que certains ont cru y voir une bonne solution. C'est une vision à court terme et qui ne garantit plus rien à l'entreprise. En cas de vol ou de casse l'employé restera donc sans

instrument de travail... A moins qu'on ne se serve pas réellement de cet outil pour le travail, mais dans ce cas quel intérêt de parler de tout cela ? !

Dans le cas de l'achat ou du CYOD, l'entreprise devra donc faire sienne les problèmes de vols, d'accidents et de pannes. En dehors de la gestion normale de tout actif et du fait même de la taille et de l'itinérance des smartphones et tablettes il faudra assurer correctement le matériel contre tous ces aléas.

Dans tous les cas il faudra faire signer à l'employé un document rédigé par un avocat qui précisera les responsabilités de chacun. Ce que l'entreprise prendra en charge, ce qui sera à la charge du salarié, les cadres d'utilisation "normaux" et les autres (par exemple l'utilisation du smartphone sur une plage sera-t-il couvert contre les dégâts du sable, de l'eau, contre le vol ? qui payera en cas de problème, l'entreprise, son assurance, l'employé ?).

Les données

Autre point essentiel.

Comment protéger les données de l'entreprise ? Contre la perte, le vol, le partage inopiné ou volontaire, la divulgation à des tiers ?

Comment faire lorsqu'un salarié est remercié ? Doit-il restituer l'appareil ? Même en cas de fourniture par l'entreprise, s'il y a placé des données personnelles que deviennent-elles ? La porte à des procès sans fin s'ouvre ici... D'où la nécessité d'un accord signé par le salarié et rédigé par un avocat pour le compte de l'entreprise !

Les données privées et celles de l'entreprise ne doivent pas se mélanger, comment le garantir techniquement ?

Comme on le voit une bonne intégration des unités mobiles réclame de protéger l'entreprise contre bien des aléas auxquels on ne pense pas toujours !

Je n'ai pas abordé ici le problème des forfaits qui posent les mêmes questions que la machine (payés totalement par l'entreprise, participation du salarié ? – Le salarié choisit l'opérateur et le forfait ou ici aussi on met en place une solution de type BYOD / CYOD / Achat ? – Quelle responsabilité financière de l'entreprise en cas de dépassement du forfait ? En cas d'achats en ligne non professionnels ? – Etc !).

MDM contre MAM et MIM

Concernant la gestion des machines il existe des solutions plus ou moins contraignantes, plus ou moins adaptées au cas de chaque entreprise. C'est un aspect très important de la mise en œuvre d'applications pour unités mobiles. Différentes approches se retrouvent sur le marché mais aucune n'est une panacée, encore un choix à bien réfléchir !

MDM

MDM signifie "Mobile Device Management" (Gestion des Unités Mobiles). Il se présente généralement sous la forme d'un logiciel que les entreprises peuvent utiliser pour verrouiller, contrôler, crypter, et appliquer des politiques de sécurité sur les appareils mobiles comme les tablettes et les smartphones.

Dans le passé (avant 2007), la plupart des gens associaient le MDM avec Blackberry. Mais maintenant que la plupart des utilisateurs préfèrent les iPhones et les appareils Android de nombreux éditeurs de logiciels de MDM sont apparues qui permettent aux services informatiques de gérer ces nouveaux appareils, tout comme ils ont géré les Blackberry dans les dix dernières années. Ce sont des solutions éprouvées, mais le MDM est drastique, trop certainement aujourd'hui.

L'avantage du MDM est qu'il permet au service informatique d'avoir le plein contrôle sur l'appareil. L'entreprise « possède » la machine (pas seulement en tant qu'actif mais comme un démon possède une âme) et elle peut faire ce qu'elle veut dessus à distance. Du temps des premiers Blackberry payés par les entreprises et confiés aux salariés ce n'était pas un problème. L'entreprise pouvait dire : «Voici votre appareil. Il appartient à la société, nous en avons le contrôle total, nous savons tout ce que vous faites dessus et nous pouvons le récupérer quand nous le voulons". C'était la joyeuse époque où on voyait souvent des personnes trimballer dans leurs poches ou sacs à main deux ou trois téléphones, s'affolant à la moindre sonnerie et testant chaque machine pour voir laquelle était en train de sonner. On voyait aussi des gens avec un téléphone à une oreille décrocher le second et le mettre à l'autre oreille (le patron qui appelle pendant qu'on parle à sa femme, ou le contraire, deux personnes avec qui on ne rigole pas et à qui il faut répondre illico ! 😊).

Mais une fois que l'iPhone et les téléphones Android sont sortis les utilisateurs ont commencé à apporter leurs propres machines personnelles au bureau. Je connais des clients chez qui on ne peut pas accéder à Gmail et d'autres services sur le réseau de l'entreprise par exemple. Sécurité qui n'a plus aucune efficacité ! Les gens amènent donc leur appareil personnel et grâce à la 3G peuvent continuer à envoyer et recevoir des mails personnels dans leur bureau... Tout comme ils peuvent télécharger un logiciel ou des données et les installer sur leur PC même si le site de téléchargement n'est pas accessible par l'Internet de l'entreprise... Cela remet en cause toutes les restrictions de certains services informatiques...

Les services informatiques ont initialement rechigné à accepter sur leur réseau les machines personnelles mais les fournisseurs de MDM sont alors venus et ont dit, "Ne vous inquiétez pas, vous pouvez utiliser notre logiciel de MDM pour contrôler ces iPhones et Android pour les rendre parfaitement sécurisés !"

Certaines entreprises ont alors trouvé cela tout à fait intéressant, Mais pour les utilisateurs ... ce n'était pas du tout intéressant on se l'imagine !

Et c'est normal. Si vous êtes un utilisateur final et que vous achetez votre propre iPhone, voudriez-vous donner le plein contrôle de votre machine aux informaticiens

de votre entreprise ? Voulez-vous leur laisser le droit de voir tout ce que vous faites, les applications que vous installez, les contacts que vous appelez et quand vous les appelez ? Et cela tout le temps même le soir, même le week-end ? Leur laisseriez-vous la possibilité d'effacer à distance votre appareil tout entier, même par erreur, y compris tout ce qui est personnel ? Voulez-vous leur laisser imposer un mot de passe complexe avec une minuterie de 10 secondes de verrouillage de l'écran au point que prendre une photo de vos enfants deviendra une opération infernale ?

MDM c'est au final comme un root kit que votre département IT met sur votre téléphone. La plupart des utilisateurs vous dirons "Non, merci !". Mettant à bas ainsi toute votre logique d'intégration des unités mobiles dans votre entreprise ! Ainsi, alors que MDM était une idée acceptable et justifiée à l'époque où les entreprises étaient propriétaires des téléphones, maintenant que les utilisateurs sont propriétaires de leurs propres téléphones et y placent une partie de leur vie privée dessus, opter pour MDM est vécu comme un fascisme intolérable et inacceptable. Ce mode de gestion devient d'ailleurs au fil du temps et des changements du marché un anachronisme sans intérêt. Alors, quelle est l'alternative ?

MAM

Non, ni miam-miam, ni les initiales d'une politicienne. Mobile Application Management : gestion d'applications mobiles.

MAM c'est comme MDM sauf qu'il est seulement appliqué à des applications spécifiques sur un périphérique au lieu de l'ensemble de la machine. L'idée générale est que l'utilisation de MAM permet à un département informatique de verrouiller, contrôler et sécuriser simplement les applications spécifiques de l'entreprise, tandis que tout le reste sur le téléphone pourra être laissé à l'utilisateur. Les stratégies MAM ou MDM s'emploient plus souvent sur smartphone que sur tablettes tout simplement parce que les premiers disposent d'un moyen de communication permanent alors que rares sont les tablettes disposant d'une puce 3G autonome (et elles ne sont généralement pas allumées en continue comme les téléphones) et aussi parce que les tablettes sont plus chères et moins nombreuses que les smartphones. Mais tout cela évolue très vite et les évidences d'hier ne seront pas celles de demain.

Par exemple, une entreprise pourra utiliser MAM pour transmettre des mails sécurisés, des données de calendrier, des contacts et des applications de reporting à un smartphone. Donc, si l'utilisateur veut juste ouvrir son téléphone pour prendre une photo ou jouer à Angry Birds, il n'aura pas besoin d'utiliser un mot de passe pour déverrouiller le téléphone et personne ne pourra suivre cette utilisation-là de sa machine. Mais si l'utilisateur clique sur l'application e-mail sécurisée, il sera invité à saisir un mot de passe pour avoir accès à cette application.

La même chose vaut pour la politique de sécurité, le cryptage, l'effacement à distance, etc. MAM pourra permettre de crypter uniquement les données qui sont stockées par les applications de l'entreprise et les autres applications de l'utilisateur

seront installées et utilisées comme d'habitude. Pour nettoyer à distance les applications gérées par MAM la société pourra désinstaller les applications concernées et effacer les données liées tout en laissant les applications personnelles intactes. Et, peut-être plus important encore MAM ne permet pas à la société d'avoir accès aux applications et données en dehors de celles gérées par elle-même. Le smartphone de l'utilisateur ne fait qu'héberger une partie gérée de l'extérieur et seule cette partie est accessible à MAM.

Si MAM règle aussi bien tous les problèmes pourquoi tout le monde ne l'utilise pas ? La principale raison est qu'aujourd'hui les logiciels MAM ne sont pas terribles... ça va le devenir certainement mais pour l'instant ce n'est pas tout à fait au point, à part peut-être des solutions comme KNOX de Samsung mais qui n'existe que sur Samsung et sous Android ce qui en limite la portée. En revanche dans une logique d'achat ou de CYOD on peut supposer offrir une gamme Samsung sur laquelle on utilisera KNOX ce qui peut grandement simplifier les choses. KNOX fonctionne en offrant une zone sécurisée en tout point identique à la zone "libre" sauf que cette partie sécurisée est étanche. Associé à une gestion à distance, l'entreprise peut faire ce qu'elle veut sur la partie protégée mais ne peut accéder à la partie libre. Samsung a généralisé Knox à partir du Note 3 en l'offrant aussi aux particuliers. Dans ce cas la partie gérée par KNOX devient une zone hyper sécurisée du téléphone dans laquelle l'utilisateur peut stocker ses données sensibles (sans la partie de gestion à distance de la version entreprise semble-t-il – mais cela serait assez logique).

Le principal problème pour les logiciels de type MAM est que les utilisateurs veulent être en mesure de télécharger des applications à partir des magasins d'applications officiels de leur système d'exploitation (Apple App Store, Android Play Store, etc). La façon dont les app stores sont structurés fait qu'il est impossible pour les fournisseurs MAM de se glisser "entre" l'application et le périphérique de l'utilisateur. Alors les concepteurs de logiciels MAM doivent faire preuve de créativité et "the" idée n'a pas encore vu le jour. Le cas de KNOX est particulier puisqu'ici le fabricant de la devise est aussi le concepteur du logiciel, c'est d'ailleurs ce qui garantit sa solidité, le logiciel peut se baser sur des capacités particulières de la machine inaccessibles aux autres logiciels.

Par régler le problème certains fournisseurs de logiciels MAM écrivent leurs propres versions des logiciels tiers habituels. Donc si vous utilisez leur MAM vous devez utiliser leur logiciel de courriel, leur application de calendrier, etc C'est facile à installer et cela simplifie la vie de tout le monde, même de l'utilisateur parce qu'il est très simple de télécharger ces applications à partir des app stores. Mais vous comprenez tout de suite où cela coince : c'est justement qu'il faut utiliser des apps qui ne sont pas parmi les meilleures... Ce sont des apps de "second choix" en termes de look-and-feel et de fonctionnalités (par exemple qui pensez-vous peut écrire une meilleure application de messagerie? Google ou Apple ou un fournisseur de MAM pris au hasard ?) De fait, si vous obtenez vos applications auprès de votre fournisseur MAM, qu'est ce qui arrive si vous souhaitez utiliser une application que le vendeur

MAM ne peut pas offrir ? (certains fournisseurs MAM ont un SDK et des API pour permettre aux concepteurs d'applications tierces de créer des plug-in pour leurs systèmes de sécurité mais cela exige toujours que les concepteurs prennent le temps de brancher leurs apps à ces plates-formes de MAM spécifiques .. toujours plus de contraintes et de "si" ...)

D'autres fournisseurs de MAM ont une approche qui consiste à "wrapper" les applications tierces dans une coquille à eux qui permet de contrôler ainsi l'exécution de l'app. Une sorte de virus donc. Le concept est efficace à défaut d'être sympathique mais difficile à appliquer réellement. Le problème est que cela ne marche pas aussi directement car comme je le disais cela serait un virus et la plupart des OS mobiles contrôlent les packages pour éviter leur détérioration. De fait le fournisseur MAM utilisant cette stratégie n'a pas d'autres choix que de demander les packages originaux aux concepteurs pour en refaire un nouveau logiciel (identique mais avec le wrap). Et là, on le devine aisément, ça devient plus compliqué. Il faut l'accord de chaque concepteur et puis il faudrait faire cela pour le million d'applications disponibles, un travail impossible. Un bon exemple : si vous souhaitez déployer en toute sécurité Microsoft OneNote pour iOS à vos utilisateurs et que vous utilisez un fournisseur MAM qui utilise la technique du wrap, pensez-vous vraiment que vous pourrez convaincre Microsoft de vous envoyer le [.ipa](#) de sorte que vous puissiez le reconditionner avec l'enveloppe MAM ? J'ai comme un gros doute !

Je vais même dire "non !" sans prendre de gros risques !

On le voit, MAM n'est pas plus une panacée que MDM. Alors ? Il existe une troisième voie MIM ...

MIM

Mobile Information Manager. Gestion de l'information Mobile.

On a commencé par vouloir contraindre les machines avec MDM (niveau device).
Devant les problèmes posés on a voulu contraindre au niveau logiciel avec MAM.
Devant les problèmes posés on espère contrôler au niveau de l'information avec MIM. C'est aussi simple que cela.

En réalité, pour contrôler l'information, si on n'agit pas au niveau de la device ou du logiciel, il n'y a plus qu'une solution, contrôler les données donc les stocker dans un espace qu'on contrôle totalement. MIM peut être ainsi décrit grossièrement comme de même nature que tous les services basés sur le cloud, des "Dropbox-like" faisant la synchronisation de fichiers et de documents entre les différents appareils. MIM devrait d'ailleurs s'appeler Mobile Data Management. Mais on revenait au sigle MDM, un peu fâcheux... L'industrie s'est donc mise d'accord pour utiliser MIM comme acronyme...

Ce qui est bien à propos de MIM, c'est qu'il utilise des procédés largement éprouvés qui fonctionnent bien. Il y a beaucoup de services grand public (Dropbox, Box,

Microsoft SkyDrive, Google Drive, etc) que les utilisateurs peuvent installer sur leurs appareils sans grandes contraintes. Et il y a beaucoup de produits basés sur une approche MIM (Nomodesk, WatchDox, RES HyperDrive, Citrix ShareFile, VMware Octopus, etc).

En fin de compte ces produits MIM pourront être déployés dans des offres de services plus larges que l'entreprise fournira aux utilisateurs (un peu comme le courrier électronique et le calendrier), et les applications clientes seront livrées aux appareils mobiles via MDM ou MAM.

Et oui ! Même si vous contrôlez les données sur un support dans le Cloud, croyez-vous que vous pouvez laisser n'importe quelle application y avoir accès ?

MIM complexifie juste le problème car au final il faudra MAM ou MDM pour déployer les applications autorisées à travailler sur les données MIM !

Donc en adoptant MIM, vous aurez du MDM+MIM ou du MAM+MIM ...

Décevant non ?

Il y aura certainement d'autres genres de M-quelque chose-M qui sortiront. VMware par exemple parle d'une sorte d'hyperviseur notamment. Mais aujourd'hui les seules solutions qui existent sont les trois présentées, MDM, MAM et MIM. Voilà trois acronymes à placer demain à la célèbre machine à café, vous allez en boucher un coin à vos collègues avec des trucs aussi exotiques !

Conclusion

Gérer les unités mobiles dans une entreprise est un casse-tête. Aucune solution n'est totalement satisfaisante en raison soit des contraintes techniques qu'elle pose, ou de celles qu'elle fait peser sur les utilisateurs ou l'entreprise. De l'achat (achat pur, BYOD, CYOD) à la gestion des données (MAM, MDM, MIM) en passant par l'écriture d'un contrat via un avocat spécialisé, intégrer les unités mobiles réclame une bonne réflexion en amont et de bons conseils !

Etant donné le succès d'Android et des appareils comme le S4 ou le Note 3, on pourrait être tenté d'adopter une stratégie CYOD + KNOX. Cela simplifie tout. Ils sont malins chez Samsung KNOX a été conçu justement pour attirer les entreprises qui ne veulent pas se prendre la tête, et éliminer du jeu les machines concurrentes ! Après tout le marché est ouvert et la meilleure des solutions est celle qui est la plus simple à mettre en œuvre. Mais limiter le choix à quelques machines pose le problème inévitable de la frustration de l'utilisateur, de la multiplication des machines en sa possession, etc. Tout cela est à penser murement avant de se lancer !

Où se trouve le cross-plateforme dans tout ceci ? D'abord on suppose qu'en général on supportera au moins deux plateformes mobiles, l'affaire est donc entendue, mais surtout rappelez-vous que le cross-plateforme "vrai" n'est pas qu'un clonage d'application, c'est aussi le partage de code et de données entre les applications

mobiles et les applications de bureau. Gérer des appareils Android et des applications mobiles WinRT sur tablette Surface le tout en ayant une version du logiciel tournant sur toutes les versions de Windows, donc en WPF, cela fait déjà un joli montage cross-plateforme qui réclamera de bien penser à tout ce que j'ai présenté dans ce billet !

Il est vrai qu'avec les séries sur le cross-plateforme que j'ai publiées ces derniers mois, côté technique vous devez commencer à vous faire une bonne idée des contraintes et des solutions. Mais il reste à trancher sur la gestion des machines...

Avertissements

L'ensemble de textes proposés ici est issu du blog « Dot.Blog » écrit par Olivier Dahan et produit par la société E-Naxos.

Les billets ont été collectés en septembre 2013 pour les regrouper par thème et les transformer en document PDF cela pour en rendre ainsi l'accès plus facile. Les mois d'octobre et novembre ont été consacrés à la remise en forme, à la relecture, parfois à des corrections ou ajouts importants comme le livre PDF sur le cross-plateforme par exemple (TOME 5).

Les textes originaux ont été écrits entre 2007 et 2013, six longues années de présence de Dot.Blog sur le Web, lui-même suivant ses illustres prédécesseurs comme le Delphi Stargate qui était dédié au langage Delphi dans les années 90/2000.

Ce recueil peut parfois poser le problème de parler au futur de choses qui appartiennent au passé... Mais l'exactitude technique et l'à propos des informations véhiculées par tous ces billets n'a pas de temps, tant que les technologies évoquées existeront ...

Le lecteur excusera ces anachronismes de surface et prendra plaisir j'en suis certain à se concentrer sur le fond.

E-Naxos

E-Naxos est au départ une société éditrice de logiciels fondée par Olivier Dahan en 2001. Héritière d'Object Based System et de E.D.I.G. créées plus tôt (1984 pour cette dernière) elle s'est d'abord consacrée à l'édition de logiciels tels que la suite Hippocrate (gestion de cabinet médical et de cabinet de radiologie) puis d'autres produits comme par exemple MK Query Builder (requêteur visuel SQL).

Peu de temps après sa création E-Naxos s'est orientée vers le Conseil et l'Audit puis s'est ouverte à la Formation et au Développement au forfait. Faisant bénéficier ses clients de sa longue expérience dans la conception de logiciels robustes, de la relation client, de la connaissance des utilisateurs et de l'art, car finalement c'en est un, de concevoir des logiciels à la pointe mais maintenables dans le temps.

C#, Xaml ont été les piliers de cette nouvelle direction et Olivier a été récompensé par Microsoft pour son travail au sein de la communauté des développeurs WPF et Silverlight. Toutefois sa première distinction a été d'être nommé MVP C#. On ne construit pas de beaux logiciels sans bien connaître le langage...

Aujourd'hui E-Naxos continue à proposer ses services de Conseil, Audit, Formation et Développement, toutes ces activités étant centrées autour des outils et langages Microsoft, de WPF à WinRT (Windows Store) en passant par Silverlight et Windows Phone.

A l'écoute du marché et offrant toujours un conseil éclairé à ses clients, E-Naxos s'est aussi spécialisée dans le développement Cross-Plateforme, notamment dans le mariage des OS Microsoft avec Android, les deux incontournables du marché d'aujourd'hui et de demain.

N'hésitez pas à faire appel à E-Naxos, la compétence et l'expérience sont des denrées rares !