

Chapitre 18 – L'évolution technique et ses principes

Au cours des chapitres précédents, plusieurs principes ont été mis en action. Certaines compagnies ont suivi ces principes (consciemment ou non), ce qui leur a permis de grandir. Mais la très grande majorité des entreprises et des technologies sont allées à l'encontre de ces principes et ont par conséquent trébuché -on entend parler des compagnies qui ont réussi, plus rarement des compagnies qui ne sont allées nulle part. De la même manière, l'histoire de l'informatique a été le témoin de très nombreuses prédictions -dont la grande majorité s'est avérée erronée. Certaines technologies ont, comme prévu, changé la donne mais pas forcément comme on le pensait. Si tout le monde s'est vite aperçu que le Web allait changer beaucoup de choses dans notre mode de vie, la manière a été une autre histoire. Ce n'est pas la soi-disant "nouvelle économie" qui a tout changé mais plutôt des sites comme Facebook qui ont introduit des nouvelles pratiques et des nouvelles habitudes.

De la même manière, si tout le monde a vu l'interface graphique comme l'avenir (pas besoin de s'appeler Steve Jobs pour ça), l'adoption a été beaucoup plus lente que prévue. Et si Java en tant que plateforme a été un bide, Java en tant que langage s'est très bien implanté. Mais beaucoup de ces prédictions étaient tout simplement absurdes. Il suffit d'énumérer quelques exemples : L'intelligence artificielle (qui devait tout changer, y compris la programmation), le bureau sans papier (que même Xerox redoutait !), Java qui remplace Windows (Bill Gates a sans doute eu peur de cette éventualité mais Gates prend toutes les menaces au sérieux !), Linux qui remplace Windows (certains y croient encore...), le Network Computer qui remplace Windows (vous ne voyez pas une trame ici ?), la fin de la programmation (certains y croient encore, bis !) et les SGBDO (qui devaient nous faire oublier les SGBDR...). Certaines de ces technologies ou concepts existent toujours mais, dans l'ensemble, ils n'ont pas d'impact majeur sur l'informatique ou, en tout cas, pas celui annoncé !

Les principes énoncés ci-dessous ne sont ni dévoilés en exclusivité par ce présent ouvrage, ni nouveaux. Beaucoup ont été traités dans de nombreux livres. Mais la difficulté pour une entreprise ou une technologie est de suivre tous ces principes. Comme l'écrit Léon Tolstoï dans son livre "Anna Karénine", toutes les familles heureuses se ressemblent. Chaque famille malheureuse l'est à sa propre manière. En d'autres termes, pour réussir, une compagnie doit éviter tous les récifs. Aller à l'encontre d'un seul des principes peut mener à la catastrophe. Une compagnie qui sort le bon produit au bon prix mais au mauvais moment échouera (hélas !).

Parfois même il est facile de voir le bon principe... Après coup. Nous encourageons donc le lecteur à être clément envers les décisions malheureuses prises par de nombreux acteurs à travers l'histoire que nous venons de détailler. Avec du recul la bonne décision est toujours plus facile à voir. La victoire a de nombreux pères, la défaite est orpheline !

1- Principe du délai incompressible

Cow-boys contre chemin de fer ou que savez-vous vraiment de l'histoire de l'informatique ?

Si l'évolution technique continuait à progresser à la vitesse que l'on a connue au XIXe siècle, il est clair que nous ne connaîtrions plus aujourd'hui de délai entre une découverte et sa mise en application. Quelques exemples puisés en dehors de l'informatique : il a fallu attendre 70 ans pour que le téléphone équipe 50 % de la population américaine. Toujours aux États-Unis, il a fallu 39 ans pour la télévision par câble pour atteindre ce seuil alors que seulement 11 ans ont été nécessaires pour la radio... L'apogée est atteint avec le microprocesseur : entre la première réalisation de Marcian Hoff et Federico Faggin en 1969 et sa diffusion massive via le micro-ordinateur en 1977, il s'écoule moins de dix ans. Or, un délai apparemment incompressible est finalement apparu depuis la fin des années soixante-dix.

Une décennie, le voilà notre délai incompressible, le time to market cher aux industriels. Vérifiez, depuis le début des années soixante-dix, il faut au moins une dizaine d'années pour qu'une véritable innovation technique commence à trouver et à toucher son marché et son public. Et il faut au moins encore autant pour qu'on atteigne un premier seuil de généralisation. C'est qu'il faut une génération pour changer les habitudes des utilisateurs ou faire entrer des innovations dans le quotidien (c'est comme pour l'aviation commerciale, elle existe depuis quelques dizaines d'années mais n'est réellement à la disposition des masses que depuis peu).

Il est donc juste de dire que les évolutions significatives sont lentes et relativement peu fréquentes. En dépit de notre obsession pour la nouveauté technologique, les vraies révolutions sont en réalité beaucoup plus rares que l'on s'imagine. Le progrès de l'informatique ne passe pas par des ruptures. C'est plutôt un travail cyclique, où l'industrie et les utilisateurs remâchent le même concept deux, trois, quatre fois de suite, avant de sortir enfin une version suffisamment robuste pour s'imposer comme standard de fait. Et l'histoire montre que ce processus prend souvent au moins vingt ans...

Vingt ans de délais pour les grands succès

Il a fallu vingt ans pour que les interfaces graphiques commencent à se populariser. Les premières recherches de Xerox datent des années 1970 et le succès de Windows 3.0 n'arrive que dans les années 1990. Il a fallu également vingt ans pour que le SGBDR se généralise entre les premiers travaux de recherches d'IBM et le succès d'un Oracle. Dix ans pour que les écrans à matrices actives concrétisent le rêve de l'écran plat à haute définition (même en pleine lumière). Une dizaine d'années aussi pour les réseaux locaux alors qu'on leur prédisait un avenir fulgurant. Et dix ans encore pour que France Télécom se décide à diffuser Numéris à des tarifs acceptables !

Le délai incompressible vient de nombreux facteurs :

- Il faut du temps pour finir la mise au point (difficulté toujours minorée mais la réalité est tenace et se rappelle au souvenir des optimistes à ce moment-là...).
- Il faut du temps pour passer du discours technologique (c'est supérieur parce que...) au discours applicatif (cela va vous servir à...).
- Il faut du temps pour que la cible visée se reconnaisse.
- Il faut du temps pour les clients testent avant de commencer à généraliser (la plupart ont déjà essayé des plâtres trop frais et n'ont plus envie de jouer les pilotes d'essais !).

C'est ici qu'intervient le "principe de la masse critique" car les percées réclament une lente maturation. Bien entendu, il serait tout aussi erroné de prétendre que l'industrie informatique est restée figée lors de ces cinquante dernières années. Pour prendre une analogie, on peut comparer ce domaine à un grand plan d'eau qui s'étendrait toujours plus en superficie mais resterait peu profond, même en son centre. La surface toujours en expansion représenterait la progression des capacités de traitement qui ne cesse de progresser (et c'est cette progression qui donne l'impression d'une évolution sans frein) et la profondeur serait à l'image des fonctionnalités qui elles restent très limitées.

Une situation contrastée qui n'est pas une anomalie

Cette dichotomie qui pourrait passer pour une anomalie mais c'est là qu'il faut comprendre que l'accroissement des capacités vient surtout des progrès réalisés dans le domaine de l'électronique (comme le montrent les lois de Gordon F. Moore, voir à http://fr.wikipedia.org/wiki/Loi_de_Moore). L'informatique profite donc de ces progrès permanents mais, du coup, le logiciel est toujours en retard par rapport aux avancées du matériel (par exemple, quand les serveurs passent aux processeurs 64 bits, il faut un certain temps -qui se chiffre en années- avant que les logiciels systèmes soient capables de les exploiter et c'est ensuite au tour des logiciels applicatifs de se mettre à niveau... Cette évolution séquentielle est toujours lente alors que, de leur côté, les composants continuent à progresser en permanence). Or, c'est bien au niveau du logiciel que se font les vraies avancées en matière de fonctionnalités...

Le logiciel est forcément en retard sur le matériel

Il ne s'agit pas de dire que ceux qui s'occupent du matériel sont des génies et que ceux qui programment les logiciels sont des ânes !

Pour expliquer la lenteur des progrès côté logiciel, il faut mettre en avant le rôle des tests. En effet, écrire un logiciel est -relativement- rapide mais le mettre au point est terriblement plus lent... Les tests logiciels sont lents et complexes. Lents car il y a de nombreuses possibilités à vérifier et il faut tout revérifier à chaque modification (tests de non-régression). Complexes car le contexte est important et pour un logiciel système (par exemple), il faut pouvoir tester chaque plate-forme dans toutes les configurations possibles... On imagine vite ce que cela représente !

Logiciels et turbines à gaz, même combat !

Prenons un exemple, celui d'un « accident industriel » célèbre : les turbines à gaz GT24/GT26 d'Alstom qui se sont avérées défectueuses au début des années 2000. Pourquoi des engins aussi coûteux et complexes ont-ils été mis sur le marché sans être testés de fond en comble (ce qui aurait évité de commercialiser des turbines de grande puissance comportant un défaut de conception) ?

Tout simplement parce qu'il aurait fallu laisser tourner ces turbines pendant des années (4 à 5 ans minimum !) avant que le défaut ne se manifeste... La rentabilité du programme ne pouvait s'accommoder de tests aussi longs, la direction de l'époque a donc pris le risque de se contenter des tests habituels qui eux étaient tout à fait satisfaisants...

La vraie différence entre capacités et fonctionnalités

Ceci permet de comprendre pourquoi les tests jouent un rôle clé dans la lenteur de l'évolution du logiciel et font que l'informatique avance sur deux faces : très vite sur le plan du matériel, très lentement sur le plan du logiciel. Or, le matériel n'amène seulement que des capacités (autrement dit, des fonctionnements potentiels) alors que c'est le logiciel qui apporte les vraies fonctionnalités (qui elles débouchent sur des traitements bien réels)... Cependant, on peut constater tous les jours que l'accroissement constant des capacités et des performances du côté matériel (ainsi que la baisse des prix qui en résulte) suffit à obtenir des avancées spectaculaires dans bien des domaines. Si on avait dit à un ingénieur électronique des années 70 que nos adolescents écoutent désormais de la musique sur des lecteurs MP3 équipés de disques durs de plusieurs Giga Octets, il aurait eu du mal à le croire !

Ces éclairages permettent de répondre à la question "pourquoi l'évolution technique de l'informatique suivent-elles deux pentes aussi différentes ?".

2- Principe de la masse critique

Les évolutions les plus marquantes et les plus structurantes restent discrètes jusqu'à leur percée. Un progrès n'apparaît pas spontanément, il est le fruit d'une lente maturation qui s'étale souvent sur une décennie. Reprenons l'exemple de l'Internet dont on a déjà beaucoup parlé.