Développer du logiciel, une douce folie.

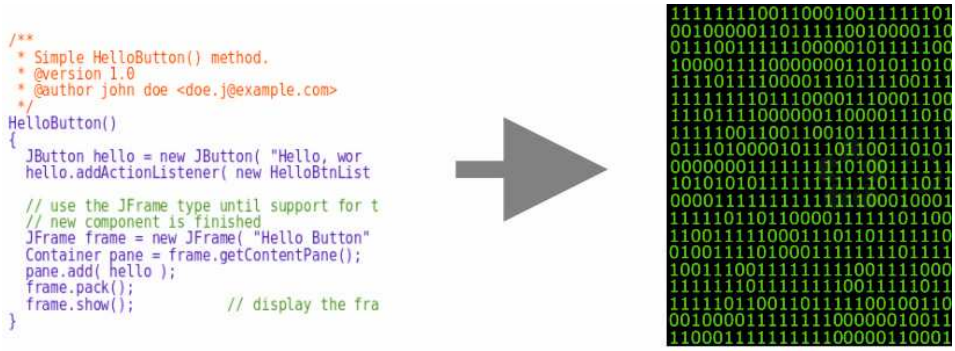
juin 2017

"L'informatique" est une activité encore jeune mais surtout méconnue. Tout s'appelle informatique et tous sont des ingénieurs sans que chacun sache exactement à quoi cela correspond.

Je m'intéresse ici plus particulièrement au développement de logiciel, à mon sens la partie la plus noble et la plus exigeante de l'informatique.

Développement de logiciel

En deux mots, développer du logiciel consiste à écrire un programme dans une langue compréhensible par l'homme et qui sera traduite dans une langue interprétée par la machine.



Des sciences de l'ingénieur, il est admis que l'écriture de code informatique est la plus complexe. Par exemple, dans les autres disciplines, la construction d'une machine ou d'un ouvrage d'art, la complexité est proportionnelle à la taille de l'ouvrage alors qu'en informatique elle est exponentielle.

Construire un programme demande une capacité d'abstraction exceptionnelle, maitriser des systèmes extrêmement complexes, appréhender des problèmes concrets et les traduire en solutions pertinentes.

La complexité même d'un programme est telle qu'il est impossible de rendre un produit parfait. Ceci explique que les "bugs" font partie intégrante d'un programme.

Enfin, il faut suivre des évolutions techniques et méthodologiques où les vérités d'hier ne sont plus celles d'aujourd'hui et certainement plus celles de demain. A ce propos on considère que chaque saut technologique laisse 30% d'informaticiens sur le bas-côté.

Méthodes

Un ensemble de méthodes accompagnent les concepteurs de programmes.
S'agissant des méthodes de conception, le nombre et la diversité des méthodes montrent à quel point l'approche n'est pas déterministe.
Au travers du temps, les assertions qui président aux méthodes ont évoluées.
Par exemple, j'ai appris au cours de mes études qu'il ne fallait en aucun cas coder un programme avant d'avoir spécifié l'entier des fonctionnalités souhaitées. Pourtant, en trente ans de carrière, je n'ai jamais vu un seul vrai cahier des charges !
Diverses raisons expliquent ce manquement à la règle, l'une est la simple incapacité à formaliser les processus.
C'est pourquoi des méthodes dites agiles permettent une plus grande souplesse dans la conception de programme, ce qui est plus en phase avec les réalités de terrain.

S'agissant des méthodes liées au développement pur, diverses technologies accompagnent les développeurs dans le but de rendre les programmes les plus fiables possibles.
Tests de non régression, réusinage de code (refactoring), gestion des versions du code et j'en passe sont autant de produits et de méthodes qui surchargent la simple écriture du programme.
Ce sont des compétences supplémentaires qui doivent être intégrées au processus d'écriture.

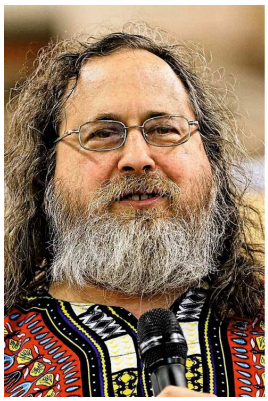
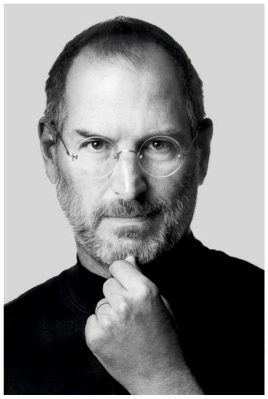
Ressources humaines

Chaque évolution représente une rupture qui, additionnées à un rythme implacable, sont autant de connaissances acquises qui deviennent obsolètes et qui déstabilisent leur détenteur.

L'ensemble des complexités énoncées doivent être réalisées par des professionnels aux compétences multidimensionnelles éprouvées. Autant dire que ces professionnels talentueux sont très rares.

Une formation de haut niveau et une longue expérience sont un gage de qualité. D'ailleurs, il ne viendrait à l'idée de personne de demander au terrassier de concevoir un gratte-ciel, quand bien même l'expérience du terrassier est à intégrer dans le processus de conception.
Pourtant, en informatique, le mélange des genres est permanent, avec des résultats en rapport avec le manque de structure observée.

Coder est un métier qui demande une grande intelligence, une passion dévorante, voire des dispositions mentales spécifiques particulières. Je suis d'autant plus à l'aise de l'affirmer car je considère ne pas avoir les qualités requises.


Stallman, initiateur logiciel libre – Jobs, fondateur d'Apple – Assange, cybermilitant, wikileaks.

A l'origine du succès et des innovations de rupture, nous trouvons souvent des personnalités excessives, parfois extravagantes, toujours supérieurement intelligentes et possiblement troublées. A propos de « hackeur » (pirate informatique) j'ai pu lire que leur personnalité se définissait entre ados surdoués et informaticiens autistes… que dire de plus ?

En la matière, le talent se détecte selon des critères spécifiques à la branche.
Par exemple il est intéressant d'observer l'initiative de Xavier Niel (patron de Free) en matière de formation. Son "école 42" est un établissement d'autoformation qui s'enorgueillit de former les meilleurs informaticiens de France, cela en dehors du cadre de l'éducation nationale. Pourtant, aucun diplôme préalable n'est requis pour y être reçu. Seules des dispositions mentales naturelles à appréhender l'informatique sont nécessaires. Ces bonnes dispositions sont testées lors d'un cursus de sélection en immersion appelée "la piscine".

On l'aura compris, l'informatique réinterroge nos façons d'appréhender les savoirs.

La réalisation et la constitution d'équipe

Dans cet environnement mouvant et complexe, il s'agit de fédérer des forces capables de produire des logiciels fiables, pertinents, performants et évolutifs. Pas exactement une mince affaire.

La taille du défi explique en partie la piètre qualité des logiciels en circulation.
Manque de fiabilité, non déterminisme des résultats, bugs bloquants sont le lot des logiciels mal conçus.

Plus pervers sont ceux qui délivrent les fonctionnalités requises mais qui sont incapables de tenir une charge ou une utilisation fortement multi-utilisateurs sans parler de performances brutes misérables.
Les lacunes de conception ont toujours pour cause des manquements dans l'ingénierie logicielle.

Constituer une équipe productive est un art qui représente un équilibre fragile fait de divers arbitrages.

Il est difficile d'établir les règles du succès tant les paramètres sont nombreux et les environnements différents. S'agissant de gérer une équipe de développeurs, voici quelques points d'attention issus de ma propre expérience :

1) Privilégier la stabilité

   La réussite des projets est un mix entre des compétences élevées en ingénierie logicielle et les connaissances métiers. Tout naturellement des équipes rodées, qui s'entendent bien et dont les personnalités sont connues représentent un réel bénéfice.

   Combler des retards par l'engagement de ressources supplémentaire est une erreur fatale connue sous le nom de "loi de Brooks". D'où l'importance de soumettre son projet aux bonnes pratiques car en la matière, la navigation à vue est létale.

2) Eviter les mercenaires

   Cette proposition est un corollaire de la précédente, aborder un projet avec des ressources venues de divers horizons et sans cohésion est peu rentable.
   En d'autres termes, engager des ressources le temps du projet est un exercice qui demande une grande expérience, de solides compétences de gestion et surtout un budget important.

   Les projets devant s'effectuer dans le cadre d'enveloppes budgétaires fixes dans des environnements peu expérimentés aux développements logiciels ont toutes les chances d'échouer.

3) Privilégier les petites équipes

   Les grandes équipes sur des projets ambitieux sont à l'origine d'échecs mémorables.
   Les bonnes pratiques considèrent qu'une équipe efficace ne devrait pas dépasser sept personnes.

4) Privilégier la qualité à la quantité

   Cette proposition semble évidente, cependant elle prend une dimension particulière dans le développement logiciel où la quantité de ressources dévolue à un projet ne comblera jamais la qualité.
   En effet l'augmentation de coordination et la baisse de qualité de la production péjorera le projet à une vitesse d'autant plus grande que les ressources augmentent.

5) Maintenir le mouvement

   Cette affirmation va à l'encontre de la stabilité puisqu'elle propose des changements réguliers dans l'équipe. Une équipe figée aura tendance à se reposer sur ses acquis et construire des certitudes dangereuses dans un environnement aussi mobile que celui-ci.

   Si des têtes nouvelles doivent régulièrement rafraichir l'équipe, le responsable veillera au quotidien à créer une tension afin d'éviter les trop grandes zones de confort.
   Cette contrainte est particulièrement délicate à mettre en œuvre car devant l'évolution permanente et déstabilisante de l'environnement informatique, les réflexes d'autoprotection vont opérer pour se protéger des changements naturellement anxiogènes.

Le mix de ces conseils est très délicat à mettre en œuvre. Et lorsqu'on atteint l'objectif, le maintien de l'équilibre est un exercice particulièrement délicat.

Enfin, je ne peux que conseiller de capitaliser sur l'expérience acquise. Que les projets soient réussis ou en échec, un exercice d'introspection et d'analyse permettra de dégager les conditions du succès.
Les conclusions seront formalisées et partagées afin d'entrer dans un processus d'amélioration constante.

Pascal Rulfi, est ingénieur consultant et chef d'entreprise à Genève.