

# Maquette et optimisme, un risque important



décembre 2014

Depuis toujours la conduite d'un projet informatique est un exercice périlleux dont le résultat n'est pas toujours à la hauteur des attentes.

Dans cet article, je propose d'aborder brièvement les méthodes traditionnelles de conception et de m'interroger sur la pratique consistant à formaliser les besoins sur la base de maquettes.

## Hier

Une longue liste de couacs informatiques illustre l'issue de bon nombre de projets de développement. Par le passé il était admis que seuls 3% des projets étaient livrés dans les délais, dans le budget fixé et pour les fonctionnalités convenues.

Afin d'améliorer ces résultats, dès les années 1970, les théoriciens ont formulé un certain nombre de méthodes (notamment Merise en France et Hermès en Suisse) pour maîtriser les réalisations et pour convenir d'un cadre descriptif pour le maître d'ouvrage.

A son corps défendant, l'informatique est une science horriblement complexe dont la difficulté n'est pas proportionnelle mais exponentielle à la taille de l'ouvrage. De plus, il fallait composer avec les donneurs d'ordre sans expérience sur les mutations numériques.

Enfin, le développement d'un programme nécessitait de produire la totalité du code, généralement de manière monolithique donc peu flexible aux changements.

C'est ainsi que les méthodes de l'époque préconisaient des cahiers des charges détaillés et fermés. Chaque changement donnait lieu à un avenant au contrat qui pouvait devenir extrêmement onéreux selon l'état d'avancement du projet.

En faisant un parallèle avec la construction d'un pont à haubans, cela reviendrait à demander en fin de réalisation une modification afin qu'il devienne suspendu. Impensable.



Les cahiers des charges exhaustifs étaient la réponse au besoin de formalisation d'un projet.

La complexité de décrire le besoin allié à la difficulté d'exprimer une vision claire et pérenne de l'organisation m'ont toujours fait paraître le cahier des charge comme un vœu pieux, voire un fantasme. A telle enseigne que je n'en ai jamais vu un rédigé dans les règles de l'art.

## Aujourd'hui

Les années 1990 voient émerger de nouvelles méthodes qui permettent de répondre de manière différente au besoin de formalisation et d'évolution d'un projet informatique dans le cadre d'un développement traditionnel.

L'évolution des outils de l'ingénierie logicielle et la possibilité de définir des architectures plus flexibles a permis l'émergence de méthodes qui répondent mieux aux besoins du marché et qui sont plus performantes en terme de résultat net. Le vocabulaire des méthodes dites agiles s'enrichit de nouveaux venus : RAD, XP, SCRUM.

La mise en application des méthodes agiles donne d'excellents résultats mais impose ses propres exigences parmi lesquelles je note :

- Une équipe de développement soudée, compétente et communicante. Petit, agile et rapide plutôt que énorme et procédurier.
- Des objectifs partagés, réalistes et auxquels tous adhèrent.
- Un client impliqué et dont les objectifs stratégiques sont clairs. De plus, il est partie prenante et suit son projet.

L'ensemble de ces contraintes implique une bonne maturité des acteurs, ce qui n'est de loin pas une généralité.

### **Alternative**

Une alternative consiste à mettre en place des progiciels de gestion intégré (ERP) adaptés à ses propres besoins.

Sachant qu'il vaut mieux acheter le meilleur que de tout reconstruire, envisager l'introduction d'un "intégré" est une stratégie légitime qui minimise le risque d'un projet aventureux.

Pour certains, c'est même l'occasion de restructurer l'entreprise en imposant une organisation qui devra s'adapter au produit plutôt que le contraire.

J'observe que cette vue théorique fait trop souvent l'économie d'une analyse approfondie considérant que ce qui marche ailleurs doit fonctionner ici. Dans la foulée, puisque qu'il s'agit de déployer un progiciel structurant, on néglige le suivi de projet avec, à l'arrivée, des morceaux de fonctionnalités sans cohérence. Par conséquent, qui n'apporteront pas le bénéfice attendu.

Impliqué dans des arbitrages pour des projets en souffrance, une pratique récente m'interpelle : *la maquette powerpoint*.

La maquette (mock-up) est généralement basée sur des "workshops" qui sont des réunions entre brainstorming et foire d'empoigne où le prestataire tente de comprendre les grandes lignes du métier du client.

Censé être un spécialiste métier, le prestataire tente de guider les débats et synthétise les échanges en vue de paramétrer une solution qui doit correspondre au mieux aux besoins du client.

Afin de transcrire la synthèse des débats dans un "ersatz" de solution logicielle, les prestataires ont pris l'habitude de dessiner des maquettes (des écrans) constituant l'adaptation de leur progiciel au besoin du client. L'intention est de donner une "idée" de à quoi va ressembler la solution.

Cette maquette est discutée avec le client et deviendra le document de spécification du projet.

Dans le cas où nous avons à faire à un progiciel vertical dont les fonctionnalités correspondent très bien au métier du client, cette démarche a le mérite de faire l'économie d'une fastidieuse analyse et une structure de projet importante.

Au demeurant, cette méthode, ou plutôt cette absence de méthode, donne une vision statique de la solution proposée et ne dit absolument rien sur les flux de données et les processus métier.

En clair, la correspondance entre les processus métier du client et les fonctionnalités du programme relèvent de la chance.

Cette tare devient criante lorsqu'on assemble plusieurs solutions de divers éditeurs. Le plus souvent, la segmentation des solutions est faite par métier où l'on choisit le meilleur programme dans chaque domaine, en minimisant la cohérence du tout sur la promesse d'une interconnexion aisée. Enfin, si le client est organisé en silo, nous avons l'assurance d'avoir maximisé les risques d'échec.

S'il y a difficulté ou échec, le constat se fera très tardivement dans le processus de réalisation du projet. Les licences auront été acquises et les prestations seront consommées à hauteur du budget sans possibilité de corriger le tir.

De plus, en cas de procédure d'arbitrage, en l'absence de documents de référence sérieux, rien ne permettra de déterminer les responsabilités des acteurs. Il ne reste alors au client qu'à remettre la main au portefeuille ou dans le pire des cas enregistrer une perte sèche.



J'illustre souvent le propos par l'ascension du Mont-Blanc par beau temps. Deux méthodes s'affrontent. La première, légère, consiste à miser sur la vitesse, ce qui est possible avec un équipement minimum et en comptant sur la clémence des éléments. La seconde, lourde qui implique plus d'équipements et d'infrastructure, donc plus de lenteur et d'effort.

La méthode légère peut fonctionner en cas de situation idéale; le beau temps se maintient, la force physique est au top durant toute l'ascension.

En revanche, si le mauvais temps arrive ou qu'un imprévu surgit durant l'ascension, le risque de l'expédition devient maximal voir léthal.

L'équipement lourd permet d'envisager des plans alternatifs et diminue le risque en cas d'événements imprévus.

Il en est de même avec les projets abordés avec optimisme et avec une structure légère. Pour le décideur il s'agira d'évaluer avec lucidité et attention les points de vigilance afin d'estimer avec objectivité le risque et les conséquences de la méthode choisie.

En tout état de cause, j'invite les responsables de projet à la plus forte méfiance lorsque le projet est abordé sur la base de workshops et de maquettes.

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

© 2014