



Le projet **DELIRE**
Développement par Equipe
de Livrables Informatiques
et Réalisation Encadrée

PR7 – Quels objectifs pour la phase de convergence



Où en êtes-vous ?

Vous avez terminé votre code. Bravo !

Quand je dis terminé, ce n'est pas juste avoir pissé le code. Cela signifie également avoir écrit et exécuté les Unit Tests pour valider la qualité de ce que vous avez pondue. Laquelle qualité ne se limite pas aux fonctionnalités, mais également aux performances et à la convivialité.

Mais chacun d'entre vous a (normalement) testé son composant en simulant ceux des autres. Il va falloir commencer à intégrer les différents composants entre eux pour pouvoir à terme faire des tests fonctionnels, pour vérifier que les fonctionnalités promises sont au rendez-vous, puis des stress tests pour valider que le produit ne part pas totalement en sucette quand on sort des sentiers battus.

Vous allez livrer un produit qui permettra à 10.000 ingénieurs de concevoir simultanément et en architecture répartie un document technique complexe. Super. Mais ceci présuppose une infrastructure complexe, des règles de réplication de données entre les sites, de mécanismes de back-up et restore, la gestion des droits d'accès, la création d'un nouvel utilisateur. Bref un peu d'administration.

Si vous ne documentez pas les procédures d'administration, et si vous ne documentez pas comment se comporter en cas de problème, vous risquez d'avoir la réputation d'un produit difficile à manager. Votre produit doit être accompagné d'un manuel d'installation et d'un manuel d'administration.

Les jeux vidéo ont ceci de merveilleux que vous les mettez entre les mains d'ados, et quelques minutes après ils sont déjà en train de jouer avec. Et vous ne verrez jamais vos ados ouvrir une quelconque documentation. Et peut-être qu'ils n'arriveront jamais à faire le tour du jeu, car il y a des fonctionnalités qu'ils ne découvriront jamais.

Oui bien sûr, et si tous les logiciels étaient tellement naturels, cela se saurait.

Le produit que vous allez livrer n'est pas un jeu vidéo mais un produit industriel

La population que vous visez n'est pas celle des gamins mais d'adultes travaillant dans des environnements complexes.

Il y a probablement des principes de base du logiciel à comprendre, des standards de dialogue à acquérir, des commandes ou des procédures assez complexes.

Et votre objectif est que l'utilisateur utilise au maximum les potentialités de votre produit. Il a été acheté non pour que les ingénieurs s'amuse mais pour qu'ils gagnent en productivité.

Votre produit doit être accompagné d'un manuel d'utilisation.

Vous livrez un produit industriel. Même si vous avez travaillé avec soin, il comportera des bugs. Un bug, c'est souvent un problème gênant, mais ce peut être un problème qui bloque la production. 10.000 ingénieur qui sont bloqués, c'est globalement 5 millions d'€ qui partent en fumée chaque jour. Gloups !

Si un problème bloquant se révèle, le temps de correction est primordial. Pour ce faire, il est important que le client définisse bien le scénario et la configuration de données qui l'amène à ce problème, de façon à ce que vous puissiez reproduire le problème, trouver un by-pass puis mettre au point une correction qui sera envoyée au client bloqué. Il est probable que si vous ne documenter pas très précisément la façon de vous reporter un problème, le client dans sa panique vous donne un tas de scories inutile et ne fournisse pas les informations essentielles. Votre produit doit être accompagné d'un manuel de maintenance.



Vous êtes des êtres purs, dont la motivation profonde est de faire du bon travail. Mais le fait que la qualité de ce travail soit reconnue ne vous déplairait pas. Pour ce faire, il va falloir vendre votre travail. Le moment privilégié pour ce faire sera la soutenance finale du projet DELIRE. Il sera donc nécessaire de construire puis répéter une présentation, de construire puis répéter une démonstration

Vous aviez défini des livrables en fin de phase de spécification et de structuration. Un plan superbe, qui ne pouvait que fonctionner. Mais la phase de réalisation, puis celle de convergence, vous faire apparaître des pièges et des difficultés, et le plan ne se déroulera pas comme prévu. Ce qui est important est de voir ce qui a bien marché et ce qui n'a pas bien marché et pourquoi. Dans ce cadre, il vous faudra reprendre vos livrables et raconter non plus l'histoire du petit chaperon rouge, mais la vraie vie avec le loup et la grand-mère.

Enfin, vous avez sans doute entendu ce message : « L'important n'est ni le but ni le chemin mais le cheminement ». Ce qui m'intéresse en fin de projet DELIRE n'est pas tant le produit que vous avez réalisé, mais surtout ce que vous avez tiré de vos différentes étapes au cours du projet. Chacun aura vécu son projet qui sera différent de celui de votre voisin : d'une part parce que vous êtes différents, d'autre part parce que vous n'aviez pas le même rôle au sein du projet. Une expérience ne vous enrichit que si vous la formulez : c'est le rôle du Post Mortem.



Arrêter un tanker

Arrêter un tanker, chargé de quelques centaines de milliers de tonnes de fuel, ne se fait pas instantanément. Pour pouvoir stopper à Saint Nazaire, il faut couper le moteur 200 kms avant l'arrivée.



Il en est de même d'une équipe de développement. Si on veut arrêter dans de bonnes conditions le projet, il faut commencer à freiner l'équipe un certain temps à l'avance.

Un développeur a ceci de fantastique qu'il pense que ce qu'il vient de corriger est sa dernière erreur, et que ce qu'il va livrer n'aura aucun impact. Quelques dizaines d'années d'expérience me conduisent à vous dire que la réalité est un tantinet différente.

Il faut donc expliquer au développeur que la phase probablement agréable de codage est finie, et qu'il est temps de s'attaquer à des tâches beaucoup moins glamour, mais qui permettront au projet de s'achever dans de bonnes conditions.



Il faut savoir arrêter un projet

« *Il faut savoir arrêter une grève* » Maurice Thorez



Dans une négociation patron/employés, les espoirs des employés sont immenses quand ils se mettent en grève : revalorisation du salaire, réduction du temps de travail, amélioration des conditions de travail, paiement des jours de grève. Les négociations vont être difficiles, longues. Et tout ne sera pas obtenu, tant s'en faut. Mais à un moment un compromis sera mis sur la table, et il faut savoir l'accepter. C'est un moment difficile pour le représentant syndical, qui va devoir venir annoncer à la base qu'il a dû manger son chapeau.

Dans un projet c'est la même chose. On est parti avec des grandes idées sur ce que contiendrait le produit, en phase de spécification et de structuration. Et puis, la phase de réalisation et de convergence s'enchainant, les fonctionnalités qui vont être effectivement disponible dans la première version du produit vont être beaucoup plus modestes. Ça n'est pas toujours facile à accepter, mais persévérer dans la réalisation en étant persuadé que les ennuis sont derrière soi est ; utopie. Il faut se contenter de ce qu'on a, et mener ce projet désormais modeste jusqu'à son terme.

Et là, c'est de la responsabilité du chef de projet de le faire admettre à l'équipe. Et de maintenir sa position même si il y a des contestations sur cette décision dans l'équipe.



Objectif : bétonner la qualité

En 1989, Peugeot met sur le marché la 605. Peugeot mise comme souvent sur le classicisme, le confort et les qualités routières, L'objectif est de concurrencer les voitures allemandes (Mercedes et BMW)



La voiture avait été très bien conçue. Le châssis de la 605 a souvent été cité comme faisant partie des références du moment. La 605 avait le potentiel pour atteindre son objectif.

Oui, mais...

Elle connaît des problèmes de fiabilité, en particulier sur ses fonctions électriques, lors de ses premiers mois de commercialisation. En effet, par un équipement complet (direction à assistance variable évoluée, sièges électriques chauffants, suspensions pilotées, OBD etc.) la 605 battait des records de longueur de faisceaux électrique et de câblage. Des problèmes de fiabilité également présents sur les Citroën XM.



Le bon démarrage commercial de la grande Peugeot est mis en péril et ses défauts de fabrication ne font pas bonne impression auprès de la clientèle de ce marché conservateur et impitoyable dominé par la rigueur des allemandes et le flegme des anglaises. Des campagnes de mise à niveau sont effectuées et un re-stylage remet la berline au goût du jour. Trop tard, l'étiquette de voiture à problèmes est solidement collée à la 605 qui ne se vendra finalement pas beaucoup.

Il en est de même en informatique. Si votre produit se révèle de mauvaise qualité lors de sa mise sur le marché, il va avoir une mauvaise réputation dont il n'arrivera plus à se débarrasser.

S'il manque de fonctionnalités au départ, mais que les sorties des versions successives montrent une bonne dynamique, tout en conservant un bon niveau de qualité, il finira par percer.

Si vous sortez un produit de mauvaise qualité, vous serez enseveli sous les problèmes de maintenance et vous n'aurez plus d'énergie à consacrer au développement de nouvelles fonctionnalités.

Le niveau de qualité d'un produit lors de sa première sortie sur le marché va souvent déterminer sa carrière.



Objectif : un produit industriel

Attention au produit qui apparaît comme un infâme bricolage



Avant d'être complet, votre produit doit être industriel.

Au moment où vous allez commencer à démarcher les clients, votre produit va être évalué par quelques personnes clé :

1. Le responsable de l'installation
2. Le futur administrateur de la base de données
3. Quelques utilisateurs bien calus.

S'ils ont le sentiment que votre produit n'est ni fait ni à faire, l'évaluation va être un désastre et vous serez mis à la porte avec perte et fracas.

De façon à éviter cette épreuve pénible, appliquez-vous à rendre votre produit industriel. Ceci passe, en plus des qualités intrinsèques de l'application, par la qualité de la documentation, qui doit être rédigée dans un langage destiné, non à vous faire plaisir, mais à être lue et comprise par ses futurs destinataires.

Un conseil : documentez un de vos scénarios fonctionnels comme exemple pour les utilisateurs en leur mettant des captures d'écrans.



Pas de regret : il y aura d'autres versions

Mais Antoine, beaucoup vont penser dans l'équipe que le produit est beaucoup trop léger, et que le mettre sur le marché va être la honte.

Rappelons quelques remarques de bons sens.

Il est léger, mais la différence entre un produit mécanique et un produit numérique est que dans le second cas on peut faire de l'incrémental. Dit autrement, on peut sortir les fonctionnalités au fil du temps au travers des différentes versions du produit. Donc quand vous allez présenter le produit au futur client, vantez lui les avantages de la version actuelle, mais vendez lui également les avantages des versions à venir. Et ceci est vrai lorsque vous allez faire la présentation de votre produit à vos professeurs en soutenance finale.

DELIRE : « *Faire travailler des milliers d'ingénieurs sur le même document en Concurrent Engineering et en architecture répartie* ». Ça c'était le besoin formalisé. Mais dans la réalité, DELIRE est fait dans un cours appelé « Gestion de Projet ». Et l'objectif est de vous de faire comprendre que dans un projet informatique il n'y a pas que du code à écrire : ceci ne représente que 20% du travail. En d'autres termes, ne vous polarisez pas sur le code, même si c'est un domaine où vous êtes confortable parce qu'expert, mais comprenez bien toutes les autres dimensions du projet : spécification, structuration, architecture, design, livrable, test, convergence, post mortem...

