

L'APPARITION INOPINÉE D'UNE NOUVELLE STRATÉGIE DE DÉVELOPPEMENT OU LA REVANCHE DU LIÈVRE SUR LA TORTUE

Dans cette grande entreprise française du secteur de la Défense, une équipe se voit contrainte de réaliser un lourd développement technique en deux ans, au lieu des quatre années prévues. Elle court-circuite donc la procédure de développement habituelle (le cycle en V, qui fait référence dans le monde de la Défense) et en invente une autre, qui ressemble en fait à s'y méprendre aux méthodes agiles qui se diffusent en ce moment dans le milieu informatique. Mais, face à l'urgence du développement, et aveuglés par les subtilités des cycles de développement, la plupart des acteurs de l'entreprise ne perçoivent pas cette innovation, dans laquelle ils voient un échec en termes d'organisation. Cette nouvelle stratégie audacieuse a pourtant été couronnée de succès : elle mérite d'être analysée et reproduite.

Par **Éric HUBER***

Ces analyses ont été élaborées à l'occasion d'un stage de première année d'ingénieur des Mines, position intéressante qui permet de jouer sur deux tableaux : restant une année entière en entreprise, le jeune ingénieur se retrouve pleinement impliqué dans les activités et dans les crises de l'entreprise, mais, en même temps, sa position lui procure une certaine neutralité-interchangeabilité qui lui permet d'échanger avec

tous les acteurs, à tous les niveaux hiérarchiques, et de construire, ce faisant, une compréhension beaucoup plus complète du fonctionnement global de l'entreprise...

* Ingénieur des Mines, chef du service Économie-entreprises à la DIRECCTE de Bourgogne.



Une équipe expérimentée. Une mécanique bien huilée. Un beau produit. Une radio... Mais un produit qui vieillit. Les contrats se font rares, l'équipe cherche une opportunité qui lui permettrait de financer une remise à niveau. Rien...

Un jour, l'occasion tant attendue se présente : un riche client est intéressé ; il veut un appareil français. Mais les conditions sont dures, très dures : il faut livrer sous deux ans. Or, un développement de cette ampleur en nécessiterait au minimum quatre. Mais le chef d'équipe veut relever le défi : il va voir le nouveau directeur, un corpsard brillant, au début de sa carrière dans le groupe. Ce dernier, réfléchit, pèse le pour et le contre, et finit par accepter.

Comment le chef d'équipe va-t-il s'y prendre ? Le gestionnaire croit toujours que pour aller vite, il faut une organisation impeccable, une discipline de fer, des processus bien rodés. Dans le cas qui nous occupe, on découvre, au hasard des événements, qu'il n'en est rien.

En effet, en guise de préalable, le chef d'équipe demande au directeur la suppression de tous les contrôles de procédure habituels. Comptables et gestionnaires de projet éclatent de rire. Mais le directeur accepte ces conditions singulières. Sueurs froides. Pourtant, la témérité ne conduira pas au fiasco : au prix d'une souplesse organisationnelle surprenante, l'équipe tiendra ses objectifs, que d'aucuns, dans l'entreprise, jugeaient insensés. Que s'est-il passé ? S'agit-il d'un simple coup de chance, d'un succès que rien ne laissait augurer, ou d'une stratégie reproductible, qui gagnerait à être davantage connue ?

UN COMBAT QUI S'ANNONCE DIFFICILE

Avant tout, pourquoi profiter d'un « programme » et tenter de réaliser ce développement en deux ans, plutôt que le lancer au calme, sans pression, en quatre ans ? Dans le milieu de la Défense, on appelle « programme » (ce terme provient de l'expression « programme militaire ») un projet de développement correspondant à la commande d'un client identifié. Un programme implique donc l'obligation de fournir un certain nombre d'appareils répondant à un nombre importants de critères, dans un délai déterminé. Cet appareil (par exemple, une radio) s'intègre souvent dans un système plus vaste (par exemple, un avion de chasse) réunissant des appareils provenant de fournisseurs variés. Un programme, c'est une occasion unique de travailler avec un client pour développer l'essentiel des nouvelles fonctions et pouvoir montrer, par la suite, aux clients futurs le nouveau produit en état de marche, intégré sur un appareil connu dans tous les pays. C'est donc l'assurance de trouver un marché pour son nouveau produit. Un programme, c'est aussi la possibilité de contourner l'absence de

budget de recherche spécifique suffisant pour la mise au point dudit produit.

Mais un programme, ce n'est pas seulement une lutte contre la matière, c'est aussi un combat contre des personnes. Car, qui dit programme dit client (souvent étatique), qui viendra contrôler régulièrement l'avancement des travaux et qui est susceptible de retarder les paiements si les délais ou les performances attendues ne sont pas tenues : travailler dans le cadre d'un programme augmente donc la pression pesant sur tous les agents.

Dans son combat de deux ans qui ne fait que commencer, le chef a un atout, mais aussi de nombreux ennemis.

Son atout principal est son expertise technique, fruit d'une longue expérience dans le domaine de la radio. L'unité de 500 personnes d'un groupe de Défense français dans laquelle il évolue est assez spécifique : elle est spécialisée dans la vente d'appareils de communication. Remplissant en grande partie des contrats militaires, elle vend soit des développements pour des petites séries (quelques dizaines d'appareils), soit des systèmes uniques (système de communication militaire d'un pays, réseau de communication d'une série d'engins...). L'activité est donc tirée par les contrats gagnés, par les programmes militaires. On en sert un petit nombre, une cinquantaine en permanence, de tailles variables (de quelques dizaines de milliers d'euros à une centaine de millions). En résumé, cette unité gère en parallèle une somme de projets innovants, qui sont des activités non récurrentes par excellence.

Ainsi, dans ce monde systématiquement neuf (pour l'essentiel des collaborateurs), la maîtrise de l'inconnu structure les relations entre acteurs : le pouvoir des chefs repose dans une large mesure sur leur connaissance du portefeuille de projets et des acteurs-clés ; l'expérience est la qualité essentielle de chaque collaborateur, aussi bien dans son aspect technique (qui permet d'anticiper les coûts pour établir de bons devis, de savoir corriger les problèmes techniques, d'être capable d'inventer de nouvelles solutions pour ne pas reproduire les mêmes erreurs) que dans son aspect commercial (la connaissance personnelle des clients, et de leur pays, le cas échéant, est un facteur déterminant dans l'obtention de contrats).

À la tête d'une équipe technique impliquant une vingtaine de personnes et reposant sur quelques anciens valeureux (affectivement qualifiés de « moustachus »), le chef domine cet environnement. Ayant vécu le développement du produit phare de la génération précédente, qui était resté une « vache à lait » pendant dix ans, son équipe est la seule à comprendre le fonctionnement du nouveau produit. Les plus jeunes se sont en effet contentés de vendre la même radio, en y apportant des adaptations mineures, selon le type d'appareil sur lequel elle était destinée à fonctionner, mais aucun ne s'était penché de manière approfondie



sur l'architecture profonde de cette radio. C'est donc le chef du service technique qui prend en main la conduite du projet.

Un premier ennemi, coriace : le poids de la structure.

Car, dans une entreprise « moderne », et notre entreprise s'enorgueillit d'en être une, on ne travaille pas comme ça : on suit des procédures, on écrit des rapports, on informe les autres équipes. Chacun a un rôle à jouer, une position à occuper, au sein d'une « matrice organisationnelle ». Car, depuis une vingtaine d'années, dans les organisations ayant à mener des développements techniques, il est désormais d'usage de parler de « matrice organisationnelle ».

En effet, auparavant, une entreprise était traditionnellement découpée selon une seule dimension, par type de tâche à effectuer : direction financière, direction des ressources humaines, direction de la production. On retrouvait, par exemple, au sein de cette dernière, un service Pneumatiques, un service Carrosserie, un service Moteurs... Mais ce découpage unidimensionnel ne facilitait pas la communication entre les différents métiers. Or, cette communication est nécessaire pour concevoir efficacement de nouveaux produits. On a donc créé de nouveaux services chargés de faire travailler ensemble les représentants de chaque métier en vue de mener à bien un développement pour un client. Il est donc de coutume de parler de services de type « métier » quand la fonction des membres de ce service est de maîtriser techniquement une technologie, et de services de type « projet » quand le rôle des membres de ce service est de bien comprendre les besoins du client, de s'assurer de sa satisfaction en temps et en heure. Et c'est le service projets qui, dans une entreprise moderne digne de ce nom, organise le développement.

Cette fois-ci, pas question de lui laisser la main : il lancerait la procédure habituelle. Mais on doit travailler deux fois plus vite... ; il faudra donc mettre le service Projets hors jeu.

Mais il y a un autre ennemi, plus insidieux encore.

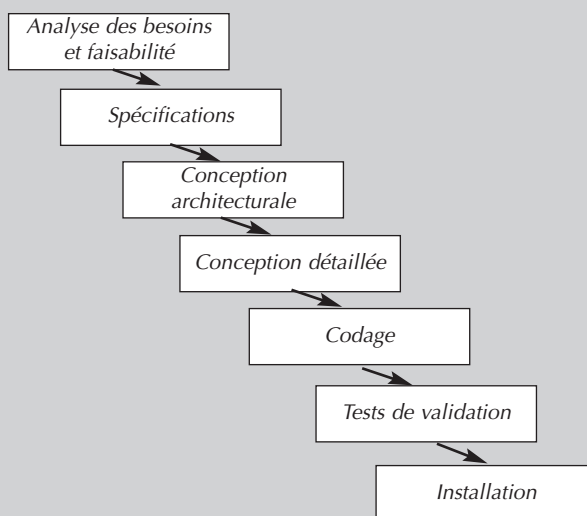
Dans les entreprises modernes, les développements doivent suivre des étapes, toujours les mêmes, que tout le monde connaît. Ces étapes ont un nom, identique dans toutes les entreprises de développement : le cycle en V. Cette notion étant assez classique, nous la présentons dans l'encadré ci-contre (que les initiés pourront sauter). Cette notion, toutefois, est plutôt complexe et les étapes à suivre sont nombreuses : la plupart des acteurs de notre entreprise sont conscients qu'ils n'en connaissent pas toutes les subtilités. Ils pourraient être aidés par la lecture des nombreux documents d'explication disponibles sur le réseau de l'entreprise, mais la longueur des explications dissuade la plupart des ingénieurs et managers...

LE CYCLE EN V : UN STANDARD RIGIDE

Aujourd'hui, de nombreuses industries ne vendent plus des produits finis, mais des projets, c'est-à-dire l'engagement de réaliser un travail avant une certaine date. Le BTP, l'industrie aéronautique, la Défense, l'industrie logicielle... en sont quelques-uns des nombreux exemples. Pour les entreprises de ces secteurs, leur aptitude à bien estimer les coûts et délais et à tenir leurs engagements est le facteur essentiel de leur réussite et de leur compétitivité. Il s'agit donc d'un sujet hautement stratégique.

Concrètement, les contrats qui lient les différentes parties prenantes d'un projet (les différents sous-traitants, le maître d'œuvre...) sont basés sur des standards de pensée appelés « cycles de développement ». Les premiers travaux effectués dans le BTP proposaient un modèle de développement linéaire dit « cycle en cascade », dans lequel les étapes se déroulent les unes après les autres : plans, fondations, maçonnerie, finitions...

Voici un exemple relatif à l'écriture d'un programme informatique simple :



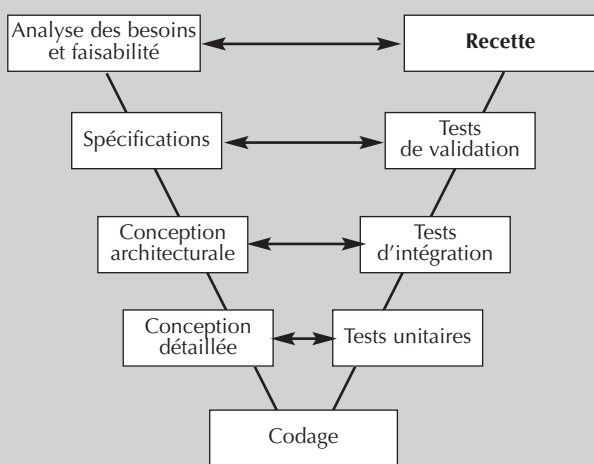
Qu'est-ce que cela signifie ? Imaginons que l'on veuille construire une montre. Analyse des besoins : on décide de faire une montre métallique, mécanique. Spécifications : elle doit donner l'heure, dévier de moins d'une seconde par jour. Conception architecturale : il faut un cadran, un bracelet, des aiguilles... Conception détaillée : l'aiguille fera 1 mm de large et 12 mm de long, le cadran aura 25 mm de diamètre... Test de validation : on mesure que l'aiguille fait la taille voulue, le bracelet l'épaisseur voulue. Installation : on assemble les pièces entre elles, on vérifie que la montre donne l'heure exacte, etc.



Mais ce schéma linéaire s'est avéré trop simpliste pour réussir des développements techniques complexes. Car l'installation, l'assemblage final échoue systématiquement (le bracelet fait la bonne largeur, mais les fixations n'ont pas le même écartement sur le bracelet et le cadran, par exemple). Les spécifications ont été mal écrites et ne répondent donc pas aux besoins du client.

Pour s'assurer que les spécifications écrites au début correspondent bien aux besoins des clients, on s'est donc mis à réfléchir à la façon dont on testera les différentes parties au moment où on écrit ces spécifications. Car, dans un cycle en cascade, on ne se met d'accord sur les tests que lorsqu'on aborde la phase de test, donc à l'avant-dernière étape.

L'industrie de la Défense travaille donc selon un cycle dénommé cycle en V, représenté par le dessin un peu obscur ci-dessous (« recette » est le nom donné à la présentation finale du produit fini au client) :



Chronologiquement, le développement se déroule de gauche à droite. On commence donc en haut à gauche, puis on descend la branche gauche du V jusqu'à la pointe, puis on remonte la branche droite : quand on veut mener un projet à bien, on commence par analyser les besoins, puis on écrit les spécifications de l'ensemble du produit. Chronologiquement, les étapes sont donc celles du cycle en cascade.

Par contre, à chaque étape de gauche, on se met d'accord sur les tests que l'on fera à droite (d'où la double flèche du dessin) : dans un cycle en cascade, c'est à la fin du cycle que l'on écrit le test et que l'on vérifie si notre montre de tout à l'heure le passe avec succès ; dans un cycle en V, on écrit le test en même temps que les spécifications, au tout début, et on vérifie si la montre de notre exemple le passe avec succès, à la fin.

Qu'est-ce qui change, pour notre montre ? Concrètement, on va écrire les spécifications et décrire les tests physiques que la montre devra passer en même temps, au début du projet. Par exemple, si on dit que la montre ne doit pas dévier, on écrit dès le début que le test de validation sera : « dévier de moins de 1 seconde en 24 heures de fonctionnement, cela étant

contrôlé avec tel ou tel moyen ». Si le bracelet doit être résistant, que les spécifications disent « résiste à une tension de 20N », on écrira le test exact pour savoir dans quelle direction s'exercera la contrainte... Ensuite, bien plus tard, on vérifiera que la montre passe ces tests avec succès.

Ensuite, le passage d'une étape à une autre (d'un rectangle à un autre, dans notre diagramme) est très strictement institutionnalisé : de nombreux documents contractuels sont à rendre (par exemple, le dossier complet des spécifications (à la fin de la phase Spécifications), ou alors les multiples comptes rendus de tests, en fin de phase de test (ainsi que les éventuelles traductions) et on effectue, en général, une présentation officielle devant le donneur d'ordres (qui, souvent, se déplace depuis l'étranger pour l'occasion)... Cette institutionnalisation est ancrée dans les mœurs du monde de la Défense : tous les contrats signés entre deux entreprises ou avec des fournisseurs contiennent des références explicites aux étapes-clés d'un cycle en V.

Toutefois, malgré les précautions prises, un cycle en V ne se déroule pas sans heurts, en un seul passage : les tests prouvent très souvent que l'assemblage des pièces ne fonctionne pas comme prévu. On comprend *a posteriori* que les spécifications étaient inadaptées ou incompatibles entre elles : on revient donc, horizontalement, à la partie gauche du V, et on s'efforce d'expliquer au client les raisons pour lesquelles les spécifications initialement validées ne fonctionnent pas. Plus le produit est complexe, et plus ces retours sont coûteux.

Ce format de cycle de développement est devenu le langage commun du monde de la Défense. Les grands groupes l'utilisent, de même que les ingénieurs de l'armement des différents pays et les militaires contrôlant l'avancement des programmes.

Il a cependant deux défauts majeurs :

- La forme du cycle a tendance à faire oublier aux acteurs externes (managers, qualité) que la réalisation concrète d'un appareil (la pointe du V) ne se fait pas en temps nul, et cet oubli conduit fréquemment à sous-estimer la durée d'un développement ;
- Un autre effet est la certitude que tout problème dans la phase remontante du V est la conséquence d'une erreur d'ingénierie amont, dans la phase descendante. Mais en réalité, nul n'est omniscient et une ingénierie amont parfaite imposerait de tester l'objet grandeur nature, et donc de l'avoir réalisé, ce qui ne se fait que dans la partie droite du V... Le risque d'une procrastination éternelle est grand... Même si les erreurs sont moins fréquentes qu'avec un cycle en cascade, il en restera toujours.

Finalement, ce cycle, malgré ses limites lorsqu'on le vit de l'intérieur, est extrêmement facile à appréhender pour quelqu'un d'extérieur : il s'est imposé comme le standard du marché. Ainsi, aujourd'hui, il est difficile de le contourner.



Conscient des défauts et des lourdeurs d'un planning calé sur le cycle en V, bien qu'il n'en soit pas un grand spécialiste, le chef veut faire autrement. Quelle est donc sa stratégie de remplacement ? C'est ce que nous allons voir maintenant.

UNE STRATÉGIE AUDACIEUSE ET INNOVANTE

LES ÉTAPES CLÉS D'UN DÉVELOPPEMENT

- Année n-1 : négociation des termes du contrat avec le client. L'ordre de l'ajout des fonctions a été choisi par les équipes techniques en fonction du schéma de développement qu'ils comptaient suivre.
- Année n : signature du contrat.
- t+ 6 mois : présentation d'un prototype intégrant de nouvelles fonctions (environ 20 % des nouveautés).
- t+ 1 an : présentation d'un prototype intégrant la plupart des nouvelles fonctions (environ 70 % des nouveautés).
- t+ 18 mois : présentation d'un prototype intégrant l'ensemble des fonctions.
- t+ 2 ans : production en petite série du produit définitif.
- Année n+3 : fin des livraisons.

La structure d'une radio étant toujours identique, le chef propose de repartir de l'architecture de la radio actuelle pour concevoir la nouvelle. Les améliorations à apporter ici sont toutefois loin d'être mineures et plusieurs performances-clés ont dû être augmentées de manière significative (parfois d'un facteur supérieur à deux) : saut de fréquence, portée de communication... L'ancien modèle comportait deux versions légèrement différentes, une version sol, qui restait à terre, et une version aéroportée, montée sur les avions ou sur les hélicoptères. Dans la nouvelle radio, les deux modèles devaient être fusionnés afin d'en simplifier l'utilisation et la maintenance. De plus, une nouvelle fonctionnalité permettant la communication avec d'autres types d'appareils (développée par une unité voisine de la même entreprise) devait y être intégrée. Le saut technologique était donc important.

Cependant, pour profiter d'un programme existant, un peu de subtilité s'impose : pour s'assurer de l'obtention du contrat, la radio a été présentée au client comme une radio « sur étagère », c'est-à-dire déjà opérationnelle, avec seulement quelques développements spécifiques à réaliser (en l'occurrence, tout ce qui concernait son insertion dans un nouvel hélicoptère). Certes, le concept de la nouvelle radio est opérationnel, puisqu'il s'agit d'une amélioration de l'ancienne. Mais de nombreuses modifications restent à apporter. L'écart est de taille...

Le langage à tenir lors des présentations au client et des validations d'étapes est donc subtil et les personnes extérieures à l'équipe finissent par ne plus distinguer ce qui est complètement vrai de ce que l'on présente comme tel au client.

Pour s'affranchir de la lenteur d'un cycle en V habituel, avec deux ans de « descente du V » puis deux ans de « remontée du V », le chef invente donc une nouvelle stratégie de développement : le développement sera donc uniquement rythmé par les présentations de prototypes (nommés « recettes », dans le monde de la Défense) très fréquentes que le contrat impose en présence du client (ces présentations sont calquées sur le mode de pensée du cycle en V, mais d'un cycle en V court de deux ans, puisque la radio en elle-même est censée exister, et que seuls quelques petits développements spécifiques sont censés être nécessaires).

Mais à chaque nouveau prototype présenté au client, on aura en réalité développé un morceau de la nouvelle radio (par exemple, on aura mis totalement au point le boîtier de la nouvelle radio, pour le jour où on devra présenter la manière de fixer la radio dans l'hélicoptère...).

En conséquence, le temps de préparation de chaque visite du client est important : il faut veiller à ne pas s'écarter, lors des présentations, du cœur des travaux effectués officiellement pour ce contrat...

Le responsable de programme ne s'y retrouvera jamais : il ne parvient pas à concevoir que l'on puisse ne pas utiliser un cycle en V, une notion qu'il maîtrise, au demeurant, mieux que les autres. Les sommaires tentatives d'explication du chef d'équipe ne le convainquent guère (elles ne convainquent d'ailleurs pas davantage les autres acteurs). Elles sont d'ailleurs plutôt prises comme des excuses de circonstance de la part d'un concepteur brillant mais parfois désorganisé, tout au moins travaillant à l'ancienne, plutôt que comme une innovation de fond portant sur la notion de cycle de développement, une notion plutôt ardue et relativement peu étudiée par la majorité des membres de l'entreprise.

Dans la pratique, le responsable de programme s'occupe donc uniquement des fréquentes visites des clients, réputés très difficiles et particulièrement capricieux, ce qui, finalement, arrange bien notre chef d'équipe. Ce client est incompetent techniquement et scientifiquement, mais il est très au fait des pratiques standard de gestion de programmes calquées sur le cycle en V : l'entente n'en est que plus difficile.

Le client devient donc un nouveau problème pour notre chef... En effet, pour garder une forme de maîtrise sur le déroulement du développement, le client en critique l'organisation et insiste pour multiplier les jalons formels et les contrôles ritualisés. Voilà qui contrarie bien notre équipe, qui, de toute façon, n'observe pas ces étapes traditionnelles. Malgré tout, pour rassurer ce client, il faut organiser très régulièrement des réunions d'information, au cours desquelles l'équipe technique explique le fonctionnement et

l'avancement des développements, en essayant de faire coller ce qu'elle fait avec la structure habituelle d'un développement en cycle en V : subtile acrobatie...

L'un dans l'autre, notre équipe technique s'autogère. Le chef d'équipe définit les priorités du moment. Malgré le temps perdu pour imaginer des présentations client crédibles, les temps des différents jalons sont plus ou moins tenus.

La livraison de prototypes fréquents est censée stabiliser le développement, en évitant les « régressions » (au cours d'un développement, on appelle régression le moment où les modifications apportées lors de l'ajout d'une nouvelle fonction perturbent le fonctionnement d'une fonction développée précédemment). Un projet qui s'autorise à laisser apparaître des régressions risque de ne jamais aboutir, car, dans un tel cas, plus personne ne sait si le développement avance, ou s'il recule... Les gestionnaires de projet sont donc toujours terrifiés par l'apparition de régressions. Mais dans l'urgence précédant les présentations au client, et afin de le satisfaire quant aux nouveaux points qu'il regarderait avec précision, on laissait apparaître des régressions sur des points qu'on espérait mineurs. À la longue, cette habitude avait fini par ralentir l'ensemble du développement, puisque l'on passait son temps à travailler sur des problèmes qui devraient être résolus, mais dont on avait perdu la trace.

Six mois avant la fin du développement, une grosse divergence avec le client, sur un point de détail, fait craindre l'échec du programme : il devient nécessaire d'être irréprochable lors de la prochaine présentation de prototype.

Le client constate que la « traçabilité » des exigences (savoir dire quel test permet de vérifier la réalisation de telle ou telle exigence contractuelle : il s'agit donc de la fonction qui, à chaque objectif assigné par le contrat, associe un test du produit dont la validité assure que le produit remplit bien son objectif) est réalisée avec un tableur informatique standard (ce qui n'est pas grotesque, lorsque ce que l'on fabrique est une radio) et non pas avec le logiciel idoine utilisé par les constructeurs aéronautiques (qui gèrent des projets bien plus complexes). Le client menace de ne pas verser les paiements suivants du développement : on met donc en place, dans l'urgence, le logiciel en question, en y ajoutant de nombreux tests, souvent redondants, à seule fin de rassurer le client.

Cette péripétie ouvre les yeux de tous : l'absence de mesure reconnue de l'état d'avancement commence à devenir insupportable. Du point de vue officiel, on travaille toujours en cycle en V traditionnel. Mais, ayant fait le choix de ne pas suivre ce modèle dans la pratique, on a négligé les indicateurs qui permettent habituellement de suivre l'avancement du projet (base de données des *bugs* constatés et/ou corrigés, résultats de certains tests réalisés en avance), ceux censés pouvoir fournir les preuves et les mesures de l'avancement des travaux. De plus, les qualitéiciens avaient renoncé

à suivre ce projet, qui ne suivait aucune de leurs recommandations.

À ce stade, une intervention fut donc décidée pour clarifier la situation et estimer une date réaliste pour la fin des travaux.

L'action mise en place pour rectifier le tir fut somme toute pragmatique : il a fallu commencer par recenser l'ensemble des activités à effectuer (documentations à rédiger, problèmes techniques persistants à résoudre) en s'assurant de limiter le nombre d'oublis, puis demander à différents protagonistes expérimentés d'estimer la durée de chacune des tâches. Cela permit d'obtenir une évaluation réaliste du travail restant à faire, prenant en compte la difficulté des tâches.

L'optimisation de l'ordonnancement de ces actions a ensuite été plus aisée.

Enfin, la mise en place d'un indicateur d'avancement basé sur cette cartographie des tâches restantes évaluées de façon crédible permet de partager les prévisions avec le reste des acteurs et de faciliter les arbitrages quotidiens entre priorités, en donnant à tous une vision générale.

Après ces corrections, le développement suivit son cours. Malgré de multiples frayeurs, et grâce aux nombreuses nuits blanches consenties par les équipes techniques, le produit fut réalisé dans les délais...

UNE VICTOIRE REMARQUABLE, OBTENUE GRÂCE À LA RÉINVENTION DES MÉTHODES AGILES DE DÉVELOPPEMENT

Si les acteurs extérieurs à l'équipe technique ont pris ces péripéties pour des conséquences évidentes de la mauvaise planification de ce développement (« les contrôles sont mal faits », « il n'y a aucune vérification », « le chef technique a pris le pouvoir : c'est le début de la fin ») et de l'absence de respect scrupuleux des étapes d'un cycle en V, telle n'est pas notre analyse. Il nous semble, en effet, que, sans que le reste de l'entreprise s'en rende réellement compte, l'équipe technique a furieusement innové, en termes d'organisation. Pour être plus précis, elle a, sans le savoir, plus ou moins réinventé des formes d'organisation des développements que certains programmeurs informatiques essaient de répandre depuis quelques années. Mais ces nouvelles méthodes de développement, appelées « méthodes agiles », n'avaient pas été pensées pour des développements matériels, ni même essayées jusqu'ici. L'encadré ci-après présente dans le détail ces méthodes relativement nouvelles, mais plutôt subtiles. Les présentations qui en sont parfois faites aux clients et aux managers les laissent dubitatifs ; ils ont donc tendance à se raccrocher au concept du cycle en V, un peu obscur lui aussi, mais dont on a pris l'habitude de parler, ce qui le rend plus familier. Que le lecteur néophyte se rassure, après la lecture de l'encadré : s'il est complexe, il est loin d'être le seul !

LES MÉTHODES « AGILES » OU DÉVELOPPEMENTS PAR CYCLES ITÉRATIFS : UNE INNOVATION ORGANISATIONNELLE ISSUE DU MONDE DU LOGICIEL

Depuis quelques années, sous l'impulsion de l'industrie du développement logiciel (lassée par la rigidité du cycle en V), de nouvelles stratégies de développement ont vu le jour. Souvent regroupées sous le vocable de « méthodes agiles », elles ont pour objectif de proposer aux développeurs de logiciels une stratégie de développement plus efficace.

Plutôt que de penser la réalisation du projet comme un unique cycle de plusieurs mois, dont l'échec est très coûteux, l'idée fondamentale est de se le représenter comme une suite de petits cycles de quelques jours, chaque petit cycle correspondant à un petit progrès sur le développement (typiquement, dans le monde du logiciel : l'ajout d'une nouvelle fonction au code initial). Cette façon de travailler rend l'équipe beaucoup plus réactive aux changements de priorités, et facilite l'apprentissage en cours de développement, en autorisant une rapide prise de recul entre chaque cycle.

Détaillons maintenant le contenu d'un cycle court, « brique de base » de ce type de développement.

Comportant quatre étapes, un cycle court a pour but d'ajouter une nouvelle fonction.

1 – au lieu de commencer par écrire des spécifications, on commence par écrire le nouveau test qui permettra de s'assurer que la nouvelle fonction répond aux besoins ;

2 – on réalise la fonction de la façon la plus directe possible ;

3 – ensuite, on vérifie que notre programme, avec sa nouvelle fonction, passe bien le nouveau test, ainsi que tous les précédents tests issus des cycles antérieurs (afin d'éviter les régressions) ;

4 – enfin, dernière étape, qui porte l'étrange nom de « refactorisation », on se demande si on aurait pu trouver une façon plus économique (ou plus intelligente) de réaliser la nouvelle fonction.

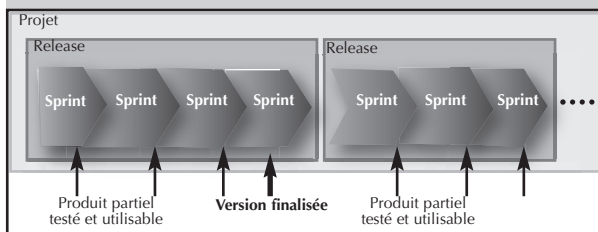
Par exemple, lors du développement de notre radio, il avait été ajouté un détecteur de panne au démarrage. Concrètement, lorsque l'on allume le poste radio, le logiciel de la radio vérifie la présence de chaque carte électronique dans le poste, ainsi que le fait qu'elles sont convenablement alimentées en électricité. L'ajout de cette fonction dans le code pourrait être fait lors d'un cycle court, en une semaine. Dans ce cas, le test doit être : quelle que soit la carte électronique mal alimentée, un message d'erreur apparaît. Ensuite, on réalise le travail proprement dit, ce qui demande de rajouter une fonction au logiciel de la radio. Puis on teste : on enlève une carte électronique, et on allume le poste, pour

chaque carte. Si on reçoit bien un message d'erreur à chaque fois, on est heureux. Enfin, et seulement à ce stade, on se demande si une optimisation de l'écriture du code informatique n'aurait pas été possible, par exemple, en passant par le même endroit du code pour chaque carte à tester...

Conceptuellement, c'est un dépassement du cycle en V, qui demandait d'écrire en même temps les spécifications et le test : l'écriture du test remplace la phase de spécifications et les spécifications disparaissent. La littérature résume cette inversion en disant que les méthodes agiles sont « *test-driven* ». Seconde subtilité : c'est dans la dernière phase (dite de « refactorisation ») que l'on se pose la question de la beauté et de l'efficacité de la solution, et non pas avant de commencer, ce qui permet de commencer tout de suite. On limite au maximum le temps perdu en sur-qualité, en spécifications inutiles, en spécifications à réécrire...

Ensuite, à la fin d'un cycle court, une fonction a donc été ajoutée à notre objet ou à notre programme informatique. Dans les méthodes agiles, l'intégration de la nouvelle fonction au produit final est donc réalisée immédiatement, dès la mise au point de la fonction, pour éviter les temps d'attente à la fin du cycle en V. Cette souplesse d'organisation est généralement fort appréciée par les développeurs.

Au bout de quelques cycles courts, on sera à même de présenter au client une version de programme plus aboutie, un prototype intégrant une nouvelle partie des fonctions que le programme final devra remplir (dans le contexte informatique, on parle plutôt de « *release* », de « sortie »).



On comprend donc maintenant le fonctionnement de la « brique de base » d'un développement agile, le cycle court. Mais comment organiser l'ensemble d'un développement ? Curieusement, la littérature est assez muette à ce sujet. Le principal apport des méthodes agiles au cycle en V semble être l'attention accordée à la réalisation concrète des objets, ce qui se trouvait être la pointe du cycle en V, et était donc facilement oublié.

Pourtant, il est possible de développer l'ensemble d'un produit complexe avec une méthode agile. Reprenons l'exemple de notre montre. Il va falloir ordonner les étapes de développement. Dans un premier cycle, je vais concevoir le bracelet, qui dimensionne la montre. Je commencerai par me fixer un objectif de taille, puis je le réaliserai. Dans un second cycle, je fabriquerai le boîtier, qui devra

pouvoir se fixer au bracelet. Dans un troisième cycle, je réaliserai le cadran, qui devra utiliser les mêmes matériaux et bien s'insérer dans le boîtier. Cycle suivant : les aiguilles, qui devront avoir la bonne taille. Je peux alors faire un « *release* », et montrer l'aspect extérieur de la montre. Ensuite, je m'attaquerai à l'intérieur, et ainsi de suite...

Dernier aspect à évoquer, l'avancement : où en est-on ? Quel pourcentage du travail reste-t-il à effectuer ? Qu'est-ce qui fonctionne aujourd'hui ? Dans un cycle en V, on mesure l'avancement étape par étape : on commence par la phase de spécification générale, en passant progressivement de 0 à 100%. Puis on commence la phase de spécification des sous-ensembles, partant de 0%, et ainsi de suite. On a attribué, au préalable, une durée à chaque étape du cycle en V, ce qui donne une mesure de l'avancement du développement. Dans un développement agile par cycle court, les étapes générales du cycle en V n'existent pas. L'avancement est mesuré par le ratio du nombre de fonctions effectivement mises en place sur le nombre de fonctions total (par exemple, si notre logiciel doit savoir envoyer des messages d'erreur à cinq autres programmes et qu'aujourd'hui il ne le fait que pour deux, on dira que l'on est à 40% d'avancement), méthode qui peut être affinée en ajoutant des pondérations. On dit donc que l'avancement est mesuré de façon fonctionnelle.

Ainsi, l'organisation par cycle court permet une grande réactivité aux changements de priorités. Les auteurs insistent toutefois sur l'importance qu'il y a à ne pas intervenir en cours de cycle court, afin de ne pas gêner le travail des développeurs et d'entretenir leur motivation.

Au sein de l'entreprise, les méthodes « agiles » se mettent en place discrètement, mais uniquement dans le monde du développement du logiciel

Ces nouveaux cycles de développement, souvent découverts sur Internet par les développeurs eux-mêmes, ont suscité l'intérêt de l'industrie du logiciel. Plusieurs entreprises ont commencé à essayer d'organiser les équipes de développement logiciel selon les recommandations de ces guides, avec des résultats apparemment très satisfaisants, malgré les difficultés de coordination avec les autres acteurs travaillant en fonction du cycle en V.

Toutefois, alors que les concepts proposés par les cycles de développement itératifs semblent assez généraux, il n'y a eu que peu de tentatives officielles de les adapter à des développements matériels.

On peut légitimement douter de l'efficacité d'un développement neuf (par exemple, la conception d'un nouveau type d'avion), qui commencerait directement par réaliser un prototype physique, sans avoir réfléchi à la structure de ses différents éléments. Mais ce cas de figure ne se rencontre pas dans

la pratique : les équipes qui innovent ont toujours en tête la structure du produit précédent, même si personne ne l'écrit explicitement.

Si les développements en cycles itératifs sont rares dans le contexte d'un développement matériel, c'est sans doute qu'inconsciemment, on craint (en général) que la multiplicité des essais et des prototypes ne conduise à gaspiller des ressources matérielles onéreuses. Cette crainte est infondée, à une époque où une journée de temps de travail d'ingénieur est plus onéreuse que l'essentiel du coût des cartes électroniques et où un poste radio complet équivaut seulement à quelques journées de travail, au coût horaire standard...

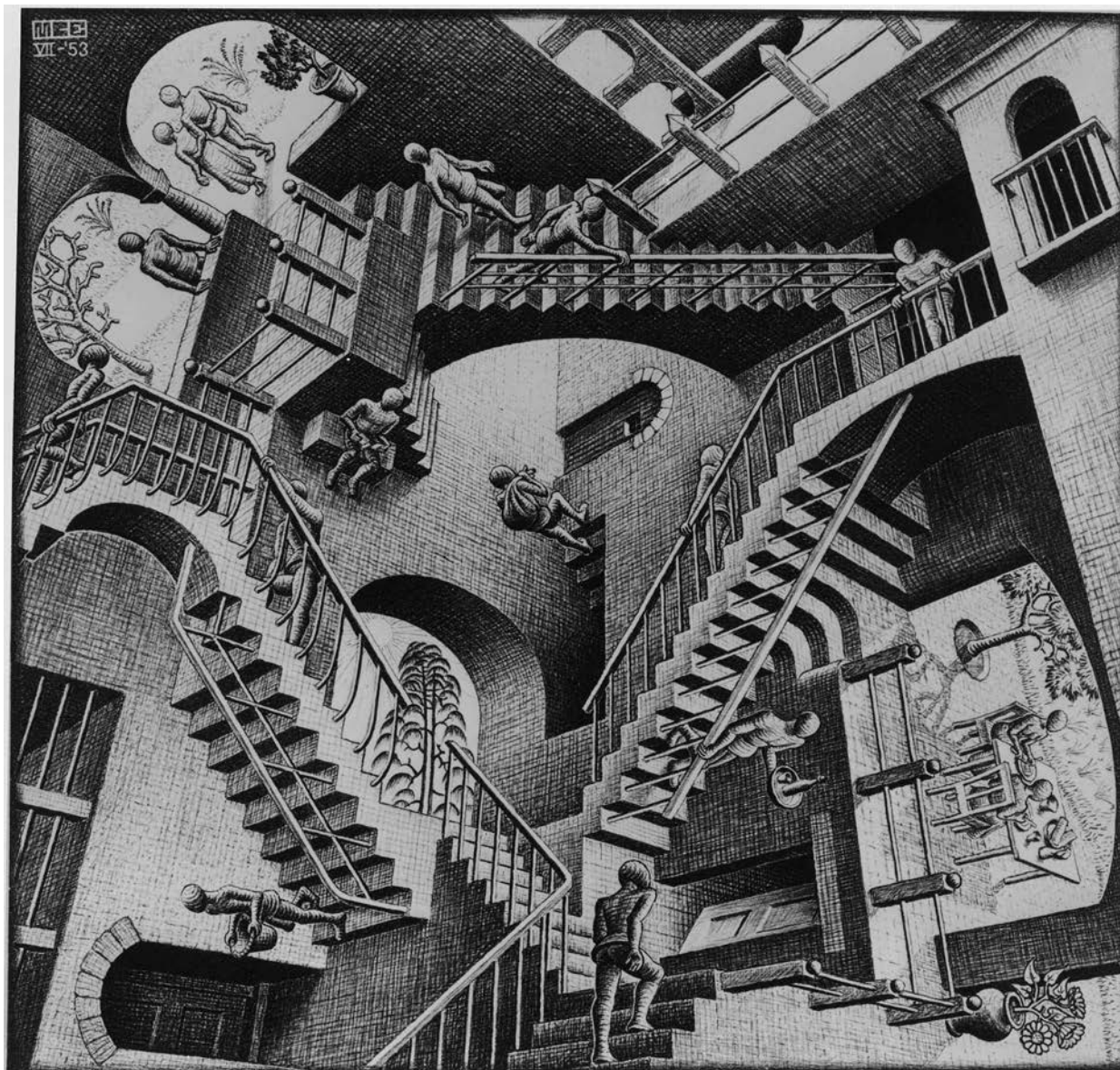
Finalement, notre équipe a adopté une philosophie de développement similaire : elle ajoute de nouvelles fonctions à chaque prototype à présenter et elle s'affranchit des premières étapes du cycle en V. Pour ce premier essai, cette stratégie de développement n'était pas pleinement rodée : la mesure de l'avancement laissa, par exemple, à désirer. Quoiqu'il en soit, le concept général est validé : le temps de développement a été divisé par deux.

UNE VICTOIRE OUBLIÉE ?

Le pari de cette stratégie de développement audacieuse a donc été gagné. Logiquement, cette méthode mériterait donc d'être analysée et reproduite. Toutefois, il est particulièrement édifiant de constater que la plupart des acteurs de notre *business unit* n'ont pas eu conscience d'avoir expérimenté une nouvelle stratégie de développement. Ils étaient persuadés de vivre un cycle en V mal fait, et non pas un développement agile. Finalement, seuls les ingénieurs de développement ont vraiment vécu un développement agile. Mais, là encore, ces derniers avaient déjà l'habitude intellectuelle de concevoir le cœur d'un développement comme une suite de cycles, une suite de problèmes à résoudre. Ils considéraient plutôt le cycle en V comme une contrainte légale un peu absurde imposée par la structure, contraignante, mais sans doute utile pour régler d'éventuels litiges entre différents contractants.

L'organisation prise dans son ensemble n'a pas pleinement compris tous les enseignements de cette aventure, puisqu'elle ne s'est pas vraiment rendu compte du caractère original de ce développement par rapport à un cycle en V traditionnel. Il nous semble donc essentiel d'en tirer les conséquences de manière explicite.

Le développement en cycles itératifs semble fonctionner pour des développements matériels. Plus naturel pour les équipes de développement, il apparaît plus efficace, surtout en cas de réutilisation de certaines parties.

M. C. Escher's *Relativity* © 2010 The M. C. Escher company-Holland. All rights reserved

« L'absence de métrique d'avancement reconnue dans le cadre de développements en cycles itératifs nuit à leur compréhension et complique leur conclusion » « *Relativity* », lithographie de M.C Escher (1953).

Ici, les résultats de cette expérience parlent d'eux-mêmes : le temps de développement a presque été divisé par deux. Bien sûr, tout ne peut pas être attribué au cycle de développement ; de toute évidence, la qualité de l'équipe et les efforts déployés ont eu un impact essentiel.

En ce qui concerne la méthode elle-même, la fréquence à laquelle les tests sont réalisés est nécessairement plus faible que pour un développement logiciel, car chaque test prend du temps. Mais cette difficulté supplémentaire ne semble pas avoir été rédhibitoire. Enfin, un atout non négligeable de ce type de programme de développement est son côté naturel : en rythmant les jalons institutionnels de la même façon que les développeurs les vivent, à savoir, par fonction ajoutée, on parvient à les associer et les motiver bien plus facilement. L'implication et la motivation des

équipes sont alors supérieures : c'est là un avantage managérial que l'on ne doit pas sous-estimer.

L'absence de métrique d'avancement reconnue dans le cadre de développements en cycles itératifs nuit à leur compréhension et complique leur conclusion

C'est la difficulté typique rencontrée au cours de ce développement : la mesure de l'avancement n'est pas naturelle, dans un développement par cycles itératifs. En outre, tous les acteurs du marché de la Défense fonctionnant en cycles en V, les acteurs extérieurs à l'équipe de développement ont tendance à se raccrocher aux métriques standards et à vouloir imposer leurs contraintes aux équipes de développement. Le travail des développeurs fut donc fortement perturbé.

Dans notre cas, finalement, ce n'est qu'à la fin du développement qu'une mesure adaptée de l'avancement a été mise en place ; lors des prochaines tentatives, il conviendrait de le faire dès le début. Le fait d'écrire explicitement quelles fonctions doivent être implémentées pour chaque prototype ne prendrait que peu de temps, le travail intellectuel d'ordonnement ayant été réalisé lors des négociations, avant la signature du contrat. Écrire, en face de chaque fonction, le test associé est immédiat : rédiger le document au début du développement, plutôt qu'à la fin, n'aurait pas posé problème. Il reste juste à penser de temps en temps à tenir à jour un planning des cycles courts, afin d'avoir une idée de l'ordonnement des tâches et de la fin possible du développement, fusse à un ou deux cycles près.

Les fréquents changements de priorités au cours d'un cycle court créent d'importantes pertes d'efficacité

Une conséquence intéressante de la mise en place d'un indicateur d'avancement précis prenant en compte la durée des tâches a été de pouvoir comparer l'avancement réel à la durée de chaque tâche prise individuellement. Un résultat inattendu, mais extrêmement régulier, a été la réalisation, chaque semaine, de 70% du travail prévu. Autrement dit, les changements de priorités, les interruptions et les réunions de crise ont fait perdre 30% d'efficacité au programme sur les six derniers mois, ce qui correspond à un allongement de presque trois mois.

Ainsi, on constate que les guides des méthodes agiles de développement, qui insistent sur l'importance de la protection des équipes de développement contre les contraintes extérieures pour des raisons plutôt philosophiques, ont soulevé un point essentiel. Cadencer précisément les développements et savoir à quel moment on peut changer de priorité (entre les cycles courts) et à quel moment on ne le peut pas (au milieu des cycles courts), cela permet, au final, de gagner beaucoup de temps.

Les développements en cycle en V ou par cycles itératifs décrivent des aspects différents d'un développement : ils sont complémentaires, dans les développements impliquant plusieurs équipes

En réalité, les deux types de cycle ont leurs avantages et leurs inconvénients. Chacun décrit avec précision un développement, et en laisse un autre dans l'ombre. Le cycle en V est particulièrement recommandé lorsque l'on souhaite faire travailler des équipes dif-

férentes (ou des entreprises différentes) sur un même projet. La définition précise des spécifications et des tests, ainsi que la définition précise d'une architecture, facilitent la contractualisation. Cela explique, par exemple, sa prédominance dans l'aéronautique, où des entreprises différentes fabriquent les ailes, les commandes, les réacteurs d'un même aéronef... Par contre, le cycle en V est muet sur le travail d'ingénierie lui-même : ce travail étant représenté uniquement par la pointe du V, on a tendance à l'oublier. Le cycle en V décrit donc précisément la répartition du travail entre plusieurs entités, mais assez mal le développement technique en lui-même.

Les développements en cycles itératifs font l'inverse : les auteurs passent rapidement sur le découpage du travail entre plusieurs entités, et décrivent, avec précision et pertinence, le travail d'ingénierie proprement dit.

Ainsi, dans un développement lourd, un cycle en V s'imposera, au début du travail, pour séparer les tâches entre les différentes équipes, mais il sera ensuite possible de faire travailler les équipes en cycles itératifs pour développer les morceaux de l'appareil dont elles ont la charge. Mais si un développement est réalisé par une seule équipe, le travail en cycles itératifs semble plus approprié. Pourtant, la pratique actuelle semble être celle du cycle en V, pour tout le monde et dans tous les cas...

RÉFLÉCHIR, OU COURIR ?

Dans le cadre de pensée du cycle en V, toute erreur, tout imprévu est naturellement imputé à un déficit de préparation, d'ingénierie amont, à de la négligence en phase descendante du V. À raisonner de la sorte, on passe sa vie à se préparer, à anticiper, à imaginer tout ce qui risquerait d'échouer, mais on n'avance pas. Or, il existe un optimum, spécifique à chaque situation. L'expérience qui s'est déroulée sous nos yeux montre deux choses.

D'abord, la pression pousse les hommes à découvrir des solutions inattendues : ce n'est que sous la contrainte que l'on innove. Ensuite, réduire massivement la phase de préparation ne conduit pas toujours au fiasco. On réalise parfois des exploits : on divise le temps de travail par deux.

Dans le monde d'aujourd'hui, La Fontaine a gagné. On serine à tous les enfants la fable du Lièvre et de la Tortue, et sa morale bien connue : « Rien ne sert de courir, il faut partir à point ».

Mais il ne faut pas trop traîner non plus ! ■