Chapitre 3
Les principes du DevOps

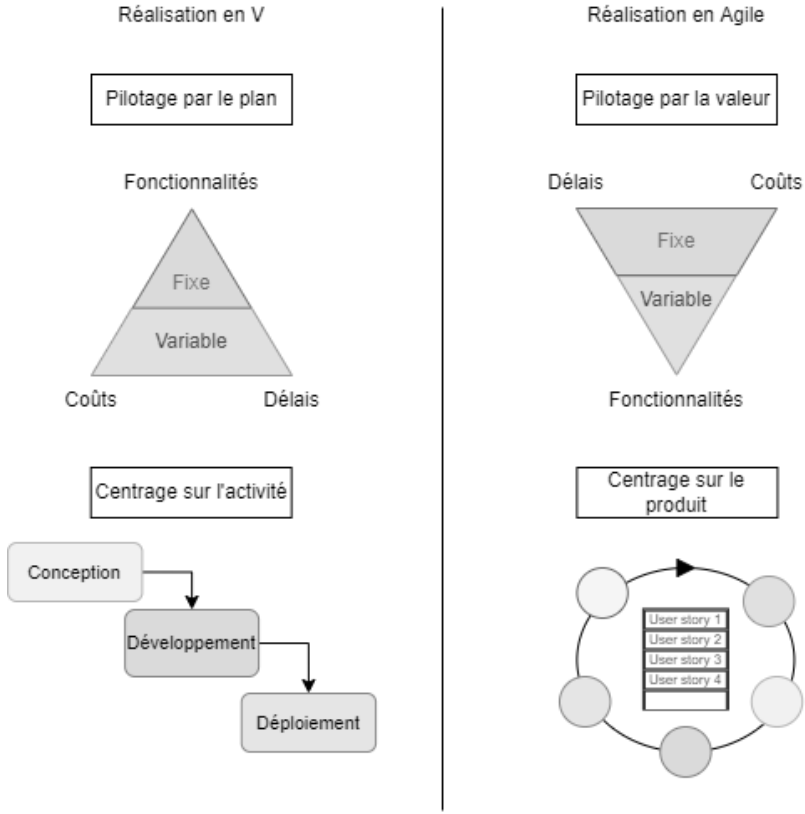
1. Être centré sur le besoin client

1.1 La démarche agile

Le chapitre Pourquoi le DevOps et contexte d'apparition a pu mettre en évidence le fait que le DevOps est apparu comme une nécessité pour compléter la démarche agile sur le volet opérationnel. Le chapitre Le mur de la confusion a également montré que l'objectif premier est de permettre aux équipes de développement et de production de travailler ensemble, dans un même mouvement, pour un même objectif, au-delà des silos.

Cet objectif est à la fois simple à comprendre et demande une vraie remise en question dans sa mise en œuvre. Il s'agit, pour aller droit au but, de se concentrer uniquement sur la valeur apportée au client, en satisfaisant à ses besoins le plus efficacement et le plus rapidement possible. C'est le crédo agile.

Pour le démontrer, remontons un peu en arrière afin de comprendre en quoi cette pratique est construite sur un changement de perception du besoin qui amène à un changement sur la façon de concevoir les projets :



Ce schéma montre qu'en premier lieu on inverse la pyramide définissant le triptyque [Périmètre/Fonctionnalités, Coût/Ressources, Délai], où le périmètre fonctionnel n'est plus l'élément fixe sur lequel s'ajustent le coût et le délai, mais devient l'élément variable, et par voie de conséquence fige le coût des ressources et le délai de déploiement.

La raison est simple : il s'agit de se donner la possibilité d'adapter le périmètre fonctionnel en fonction du vrai besoin, sans que cela nécessite davantage d'investissement.

© Editions ENI - All rights reserved

Ensuite, il montre le passage d'une méthode linéaire (même s'il est appelé cycle en V) dans laquelle se succèdent des activités phares réalisées par des équipes spécialisées (la spécification, le développement, le déploiement), vers une activité cyclique qui englobe toutes les activités à la fois, réalisées par une équipe pluridisciplinaire.

Ce changement est consécutif du précédent. En effet, si vous souhaitez adapter le périmètre fonctionnel de votre application, il n'est pas possible de suivre un grand plan défini à l'avance et fixant définitivement la liste des fonctionnalités ainsi que leur contenu. Il faut au contraire se donner la possibilité de revoir ces fonctionnalités régulièrement en avançant à petits pas et en réévaluant la pertinence de ce qui a été fait et de ce qui reste à faire. Cette méthode est dite itérative (par petits pas) incrémentale (qui redéfinit le contenu à chaque pas).

1.2 DevOps est une pratique agile centrée sur le besoin client

Le passage vers un modèle agile a cependant rapidement rencontré une difficulté structurelle qui réduit largement son intérêt et sa portée. En effet, la démarche itérative incrémentale, qui repose sur l'idée de requestionner la pertinence du périmètre et des fonctionnalités à chaque itération, suppose de baser ses décisions sur des données concrètes d'utilisation.

Ces données doivent provenir du retour fait par des utilisateurs réels, les testeurs ou les responsables produits n'étant pas des utilisateurs réels. Elles doivent également fournir un état de l'adéquation entre ce qui a été réalisé à l'issue d'une itération i et le besoin réel. Ce feedback fournit des données tangibles qui servent à réévaluer la pertinence des fonctionnalités développées et à faire les ajustements qui s'avéreraient nécessaires par rapport au périmètre et au contenu fonctionnel. On dit que l'on pivote, c'est-à-dire que l'on change la direction de l'objectif à atteindre.

L'importance de ces boucles de feedback et de la remise en cause du besoin de façon systématique a été popularisée dans le best-seller d'Eric Ryes : *Lean Startup*. Mais dans la réalité des DSI, cette démarche se confronte à une étape qui ne rentre pas immédiatement dans le modèle organisationnel agile : le passage en production.

Les équipes opérations se concentrent sur la stabilité des systèmes. Elles repoussent donc l'idée de déployer régulièrement les changements d'un produit afin d'en "tester" l'adéquation avec le besoin utilisateur, ce qui introduirait un risque d'instabilité systématique.

C'est pour traiter cette difficulté que l'idée du DevOps est née, en intégrant l'équipe "ops" à l'équipe produit, lui permettant ainsi de participer et d'anticiper chaque changement en cours de réalisation, dans un délai très court. Cela permet d'aller rencontrer les utilisateurs le plus souvent possible afin d'obtenir des feedbacks factuels basés sur des clients réels et critiques. En d'autres termes, le DevOps va conduire à l'aboutissement du modèle agile en autorisant de rechallenger le besoin à chaque itération et en lui permettant de se centrer entièrement sur les besoins des clients.

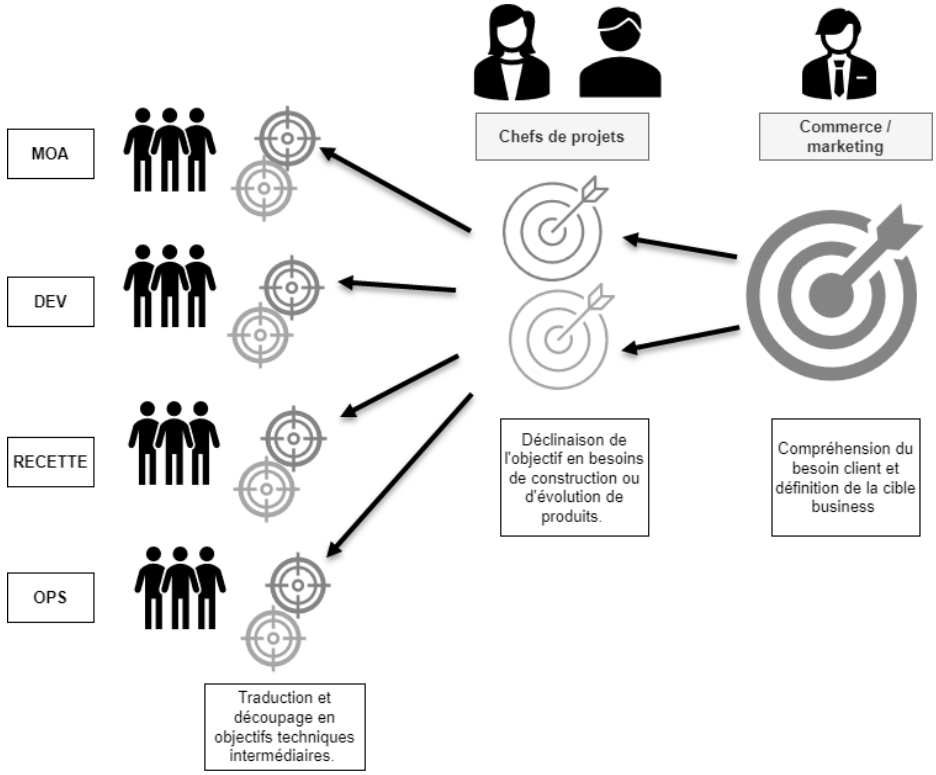
Il est par ailleurs important de ne pas différencier l'agilité et le DevOps en deux méthodes distinctes. En réalité le DevOps, du fait de sa participation au centrage client, fait partie de l'ensemble des pratiques qui définissent un modèle d'organisation agile, le DevOps étant la pratique en constituant l'aboutissement.

2. Construire en étant conscient de l'objectif

2.1 Le problème des objectifs intermédiaires

Dans une entreprise organisée en domaines d'expertise, la vision du produit détenue par chaque équipe se limite à l'engagement de ce qu'elle réalise pour les autres équipes. La vision d'ensemble de l'objectif est comprise au mieux par le chef de projet, qui détient une vue opérationnelle apportée par le cahier des charges ou par le donneur d'ordre du métier. En réalité, le vrai objectif business n'est souvent connu que des services marketing ou commercial qui déterminent un objectif en fonction des données à leur disposition. Ces données ne sont malheureusement pas toujours récentes, peuvent être partielles, ou déjà préinterprétées selon le canal qui les procure. Ce qui a été compris de l'objectif va ensuite être décliné dans les équipes opérationnelles.

© Editions ENI - All rights reserved



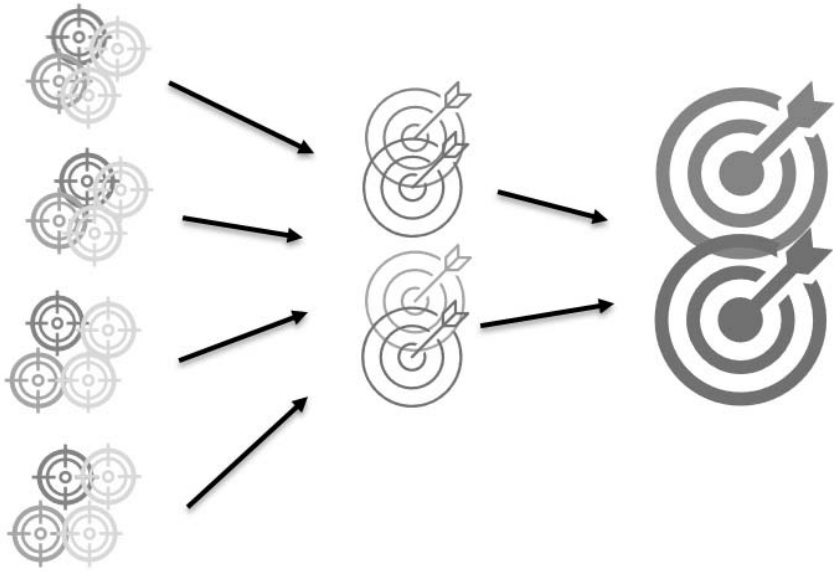
Dans le schéma précédent, il apparaît clairement que l'objectif est décliné du haut (le management ou les sachants), vers le bas (les équipes opérationnelles). À chaque déclinaison, l'objectif perd en cohérence et en vision d'ensemble, pour finalement être découpé en tranches techniques. C'est ici la pure application du taylorisme, avec une spécialisation des équipes techniques qui construisent des bouts de produits, sans savoir exactement ce qui sortira de leur assemblage, cette vision étant contrôlée par le management de haut niveau.

On voit que dès la première déclinaison, la vision d'ensemble est découpée en besoins sur plusieurs produits, ce qui contribue à éparpiller la vision de l'objectif. Ensuite, le chef de projet reproduit ce découpage en assignant des tâches spécifiques à diverses équipes techniques en fonction de leur domaine de compétence et d'expertise.

2.2 La reconstruction difficile de l'objectif global

Le problème, au moins pour le domaine de l'informatique, c'est que cette fonction de répartition du travail n'est pas réciproque. C'est-à-dire que le réassemblage des objectifs intermédiaires ne reconstruit pas nécessairement la vision d'ensemble qui a servi à leur élaboration.

Si nous souhaitons compléter le schéma en tentant de réassembler les objectifs intermédiaires vers l'objectif de départ, nous nous rendrons compte que l'objectif n'aura été atteint que partiellement, voire pas du tout. Le schéma suivant met en évidence que les objectifs reconstruits ne coïncident pas exactement avec les objectifs de départ, un peu comme une photo floutée, où l'on pourrait distinguer une forme générale, mais où chaque élément aurait subi un décalage.



© Editions ENI - All rights reserved

Mais pour quelle raison ne retrouve-t-on pas l'image de départ ?

Cela tient au fait que chaque équipe réalise son travail en faisant des ajustements qui collent avec l'atteinte de son objectif intermédiaire, mais sans connaître l'impact sur l'objectif global. Ces ajustements sont la conséquence d'un manque de compréhension, de précision ou de clarté dans les spécifications ou dans les comportements attendus. Mais ils sont aussi liés au fait que l'assignation des objectifs intermédiaires ne prend pas assez en compte toutes les contraintes existantes qu'il n'est pas facile de contourner, autant pour des questions de budget, de délai, que de faisabilité. Des ajustements, que l'on pourrait même nommer des arrangements, sont donc nécessaires afin de remplir les objectifs intermédiaires. Mais ils créent des décalages qui, mis bout à bout, rendent floue la photo de l'objectif global.

Ce constat est appuyé par les études portant sur la réussite des projets, notamment celle du Standish Group Studies Chaos (https://www.standishgroup.com/news/37) qui en 2018 constatait que 8 projets sur 10 étaient en échec. L'étude a été réalisée en prenant en compte 50 000 projets à travers le monde entre 2013 et 2017, pour des projets allant de 400 000 à 2 millions d'euros. Ce qui définit un succès dans l'étude peut par ailleurs être soumis à débat car elle ne se base que sur l'indicateur PCR (Périmètre, Coût, Délai) et ne prend pas assez en compte l'apport de valeur. En revanche, il y a consensus sur ce qui pose problème : des spécifications manquant de clarté, pas d'accès au client ou pas de volonté de laisser l'accès au client. Ce sont exactement ces causes qui produisent la nécessité de réaliser des ajustements décidés sans la participation du client et sans compréhension des impacts sur l'objectif global.

À l'inverse, le DevOps demande que tous les membres de l'équipe, que ce soit le responsable métier, le chef de projet, l'équipe de développement ou l'équipe production, partagent la même vision et l'objectif global du projet. Chaque décision prise, technique ou fonctionnelle, doit l'être, non pas en prenant en compte un objectif intermédiaire local, mais en ayant conscience de son impact sur l'objectif global et en le soumettant au plus vite au jugement du client final.