

Comment gérer les projets de développement ?

Introduction :

L'évolution d'un Système d'Information nécessite l'adaptation d'applications existantes ou le développement de nouvelles. Le développement d'une application relève d'un processus complexe, souvent soumis à de nombreuses contraintes évolutives. Dans ce contexte, quelles méthodes utiliser pour répondre au mieux aux besoins du client, tout en respectant les délais et les coûts ?

I Les méthodes classiques de développement... :

1. Les méthodes classiques de développement :

Les méthodes classiques de développement reposent essentiellement sur l'utilisation d'outils mathématiques, tels que les diagrammes de Gantt ou les graphes d'ordonnancement (MPM, PERT).

Parmi les méthodes traditionnelles, on retrouve le modèle en cascade, ou linéaire, dans lequel les étapes s'enchaînent chronologiquement et où le passage à l'étape suivante est entraîné par la validation de l'étape précédente. Il existe aussi le modèle en V, dans lequel on conçoit des tests ciblés pour chaque phase de spécification ou de conception, ce qui permet d'anticiper les résultats attendus et de cibler, en cas de problèmes, la phase à revoir.

Il s'agit donc d'outils rigides qui ne laissent absolument pas la place à l'adaptation aux évolutions des spécifications fonctionnelles, de l'environnement et du contexte de développement...

2. Les risques et limites de ces méthodes :

Ces méthodes entraînent des risques pour le produit final. En effet, selon une étude commandée par le Congrès américain, on compte seulement 2% de logiciels qui fonctionnent à la livraison, 45% qui ne seront jamais utilisés et 30% payés, mais jamais livrés...

Les problèmes rencontrés sont généralement de quatre ordres :

- **les problèmes de respect des délais :**
Lorsque le planning de réalisation de la solution applicative est établi, en début de projet, les délais fixés sont généralement trop courts. Plus le développement de la solution avance, plus les retards pris s'accumulent et moins il est possible de les corriger.
- **les problèmes de respect des coûts :**
De manière générale, le coût du projet est volontairement sous-estimé dans le but de proposer un prix attractif au client et de le pousser à contracter avec la société de services.

- **les problèmes de respect des spécifications fonctionnelles :**

Dans le cas de nombreux projets, les spécifications fonctionnelles sont décrites dans un cahier des charges remis au prestataire et font l'objet de longues négociations contractuelles; chacune des parties au contrat tentant de se protéger de la défaillance de l'autre. Cependant, bien souvent, le client ne sait pas réellement ce dont il a besoin, au début du projet, et ce n'est qu'en cours de réalisation qu'il commence à percevoir ses besoins réels. Mais, à ce moment, il est trop tard, le contrat ayant été approuvé par les parties en début de projet, il ne peut faire l'objet d'aucune modification. Ainsi, le client reçoit, à la fin du projet, une application qui ne répondra pas, ou que partiellement, à ses besoins.

- **les problèmes techniques :**

Les méthodes mises en œuvre ne laissent la place aux tests qu'à la fin du développement. A ce moment, il est généralement trop tard pour corriger l'ensemble des bugs et des dysfonctionnements rencontrés. Le projet livré ne sera donc pas exempt de problèmes techniques.

Il ... s'opposent aux méthodes dites "agiles" :

1. Présentation :

Afin de résoudre les nombreux problèmes rencontrés avec les méthodes de gestion classiques, des méthodes dites "agiles" ont été mises en place. Ces méthodes doivent permettre d'assurer la flexibilité de l'offre (le logiciel) à la demande (du client) et de garantir une adaptation aux changements de contextes et aux modifications de spécifications intervenant pendant le processus de développement logiciel.

Les bases sur lesquelles repose le concept de l'agilité ont été formalisées dans le Manifeste Agile :

- interaction entre les individus plutôt que respect de processus formalisés
- développement logiciel plutôt que documentation exhaustive
- coopération avec le client plutôt que négociation de clauses contractuelles
- adaptation permanente plutôt que suivi d'un plan rigide défini en début de développement

2. Principes :

- **satisfaire le client :**

Le but premier d'un projet de développement est de satisfaire les besoins du client en lui proposant une application qui soit adaptée à ses besoins. Ainsi, on ne développe pas des fonctionnalités inutiles pour le client.

- **livrer vite et de manière régulière :**

Pour que le client ait toujours une vision globale de la solution développée et pour que les éventuelles erreurs de conception puissent être corrigées suffisamment tôt, il est nécessaire de procéder à des livraisons de l'application régulièrement et dans des temps assez courts.

- **coopérer avec le client :**

Le client doit s'investir dans le projet de développement afin que le prestataire soit sûr de développer une application qui réponde toujours à ses besoins.

- **travailler en équipe :**

Le travail en équipe, auto-organisée et qui communique, doit permettre une stimulation des développeurs et le développement d'une solution sans problème d'interopérabilité entre les modules, qui auront été développés en utilisant des normes de développement communes.

3. Exemples de méthodes agiles :

- la méthode Scrum
- la méthode XP (eXtreme Programming)
- la méthode RAD (Rapid Application Development)

Conclusion :

Aujourd'hui, en raison des limites des méthodes classiques de gestion de projets, les entreprises, y compris les plus grandes, se tournent vers les méthodes agiles. Ces méthodes, qui placent le client au cœur du processus de développement logiciel, semblent être une réponse aux questions de respect des délais et des coûts.

Sources :

- www.gestion-projet-informatique.vivre-aujourd'hui.fr
- referentiel.institut-agile
- www.aubryconseil.com
- www.commentcamarche.net
- www.blog-gestion-de-projet.com