Méthode de gestion de logiciels : les méthodes agiles

Les méthodes agiles sont nées dans les années 1990 quand plusieurs ingénieurs ont fait le constat commun que les méthodes classiques de gestion de projets, héritées de l'industrie, n'étaient pas appropriées à la création de logiciels informatiques. Les méthodes classiques les plus souvent employées, dites « en cascade » étaient dérivées de la méthode de fabrication utilisée par Henry Ford sur les chaînes d'assemblage.

Parmi les défauts les plus souvent cités des méthodes classiques, on peut dire qu'elles ne permettaient pas de mettre l'accent sur la valeur ajoutée intrinsèque apportée par le logiciel.

Partant de ce constat, les ingénieurs de l'époque ont établi une série de valeurs et de principes qui doivent rester centraux dans l'organisation et la gestion du projet afin de produire un logiciel de qualité avec une valeur ajoutée maximisée.

Ces valeurs[1] sont les suivantes et énoncent qu'il faut privilégier :

- les individus et leurs interactions plutôt que les processus et les outils
- un logiciel fonctionnel plutôt qu'une documentation exhaustive
- la collaboration avec les clients plutôt que la négociation contractuelle
- l'adaptation au changement plutôt que l'exécution d'un plan.

Ces valeurs sont déclinées en principes :

- Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.
- Accueillez positivement les changements de besoins, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client.
- Livrez fréquemment un logiciel fonctionnel, dans des cycles de quelques semaines à quelques mois, avec une préférence pour les plus courts.
- Les utilisateurs (ou leurs représentants) et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.
- Réalisez les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont elles ont besoin et faites-leur confiance pour atteindre les objectifs fixés.
- La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le dialogue en face à face.
- Un logiciel fonctionnel est la principale mesure de progression d'un projet.
- Les processus agiles encouragent un rythme de développement soutenable. Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant.
- Une attention continue à l'excellence technique et à un bon design.
- La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.
- Les meilleures architectures, spécifications et conceptions émergent d'équipes auto-organisées.

1 Extrait de la traduction du manifeste Agile provenant de Wikipedia, site original en anglais : https://agilemanifesto.org/

Cette ressource est mise à disposition selon les termes de la licence https://creativecommons.org/licenses/by/4.0/

- À intervalles réguliers, l'équipe réfléchit aux moyens possibles de devenir plus efficace. Puis elle s'adapte et modifie son fonctionnement en conséquence.

Par la suite, ces valeurs et principes ont été repris par plusieurs méthodologies permettant de les mettre en œuvre et de proposer une gestion de projet alternative et centrée sur l'utilisateur.

Méthodologie Scrum

L'objectif ici n'est pas de détailler toutes les méthodologies existantes mais d'évoquer la méthodologie Scrum, une des plus connues et de présenter succinctement les points forts de celles-ci et ses liens avec les valeurs mises en avant dans le manifeste agile.

La méthodologie Scrum se propose de définir trois rôles à affecter aux participants d'un projet :

- Product Owner : Représentant des utilisateurs du logiciel et garant de l'expression du besoin, il détermine les priorités du projet ;
- Scrum master : Facilitateur pour l'équipe de développement, il tient un rôle d'animateur dans l'équipe de développeurs ;
- Développeur

Une fois ces rôles définis par l'équipe, il convient alors de revenir sur quelques spécificités de la méthode Scrum et sur la prise en compte des valeurs et principes agiles :

- Cycle de développement courts appelés « itérations »

L'équipe de développement doit être en mesure de faire des démonstrations du logiciel réalisé à la fin de chaque itération de développement, d'une durée fixée à l'avance (2-3 semaines en général).

- Formaliser le besoin sous la forme de « User Stories »

Dans la méthode Scrum les acteurs ayant comme rôle celui de Product Owner sont amenés à décrire leur besoin en rédigeant des scénarii d'utilisation simple du logiciel, ces scénarios étant formalisés sous la forme d'un patron de phrase suivant :

En tant que <rôle de l'utilisateur> je veux pouvoir <fonctionnalité du logiciel> afin de pouvoir <objectif>

L'objectif de ce formalisme est d'exprimer le besoin de la façon la plus simple qui soit et également la finalité de la fonctionnalité développée, cela permettra aux développeurs de faire des suggestions d'amélioration.

- Priorisation et planification

Chaque « User Story » décrite sera ensuite ajoutée et priorisée dans une liste appelée « backlog », l'objectif étant d'adresser les fonctionnalités les plus importantes en premier. Le Scrum Master évalue la complexité des « User stories » avec son équipe, celle-ci servira par la suite pour le chiffrage.

- Mettre l'accent sur la qualité au travers de tests unitaires

Les cycles de développement étant assez courts (2-3 semaines), il est important de mettre en place des tests unitaires permettant de valider le bon fonctionnement du logiciel entre deux évolutions. (Cf. module Oscar sur la production de code de qualité)

- Mettre à disposition le logiciel réalisé

L'objectif est de mettre dans les mains de l'utilisateur final un logiciel fonctionnel afin de recueillir son « feed back » qui sera plus facile à exprimer via une utilisation directe qu'au travers des moyens plus traditionnels (spécifications fonctionnelles, captures d'écrans, maquettes…). Un des défauts fréquemment constaté dans le développement de logiciel est dû à l'« effet tunnel », les développeurs ayant réalisé l'implémentation des fonctionnalités demandées sans les présenter et sans intégrer de retours de la part des utilisateurs mais uniquement sur les besoins initiaux émis.

Pour pouvoir mettre à disposition le logiciel rapidement, l'intégration continue et le déploiement continue peuvent être utilisés (cf. rubrique sur la forge logicielle)

Il est important de préciser que ce qui est présenté ici n'est qu'une illustration de ce que peut apporter une méthode agile comme Scrum a un projet. D'autres méthodologies agiles existent et peuvent être employées.

À vous de choisir celle qui pourra le mieux convenir à votre organisation. Il sera aussi probablement nécessaire de suivre une formation pour aller plus loin dans vos pratiques.

Pour citer cette fiche :
INRAE DipSO (2024). Méthode de gestion de logiciels : les méthodes agiles. Formation OSCAR Ouvrir la science, connaissance à acquérir : module Gestion et partage des algorithmes, codes et logiciels. 3 p.
oscar.inrae.fr