

Résumé individuel
remis dans le cadre du cours

420-B34-RO – Développement des applications informatiques

Depuis ses débuts, la gestion d'un projet de développement logiciel est un défi de taille pour toute les parties prenantes. Constatant un nombre très élevé de projets de développement qui n'aboutissaient à rien au final (environ 70%), plusieurs développeurs chevronnés se sont rencontrés en 1968 lors du *NATO Software Engineering Conference* afin d'établir des diagnostics sur les problèmes les plus fréquents et tenter d'y apporter des correctifs afin d'améliorer les approches utilisées. Au fil du temps, plusieurs approches ont été développées tel que l'approche *waterfall*, *eXtremeProgramming* et la méthode agile. Dans ce résumé nous aborderons quelques différences fondamentales entre la gestion d'une équipe de développement logiciel en mode *waterfall* et en mode agile et nous ferons une présentation des piliers de cette dernière approche.

La première différence se manifeste dans la phase de démarrage, dans le modèle *waterfall* aussi appelé modèle prédictif on définit la portée du projet et on tente de concevoir entièrement la solution sans nécessairement fixer le temps nécessaire et les coûts associés. La phase de démarrage est le seul moment avant la livraison où l'utilisateur final de la solution fait partie intégrante du processus. Cette approche est plutôt inflexible et mène fréquemment à des dépassements de coûts et des délais de livraison importants. Dans le modèle agile, aussi appelée modèle adaptatif, on s'entend tout d'abord sur la date de fin du projet ainsi que sur le budget qui lui est alloué. Contrairement à l'approche *waterfall* on ne conçoit pas l'entièreté de la solution mais on tente d'identifier les fonctionnalités à implémenter de façon prioritaire. Ces fonctionnalités seront implémentées graduellement selon leur priorité dans des itérations d'une durée fixe (1 à 4 semaines) déterminée dans la phase de démarrage.

Avant de présenter les détails de l'approche je crois qu'il soit important d'identifier les acteurs principaux prenant part au processus Agile. Tout d'abord nous avons le *Product owner* ou le responsable du produit. C'est généralement celui qui fait le lien avec le client final et détermine ses besoins. Il est le responsable de la portée du projet et des fonctionnalités qui y seront incluses. Ce dernier est en charge de prioriser les fonctionnalités en les hiérarchisant selon la valeur pour le client. Il s'agit généralement d'une personne davantage orientée *business*. Il est courant que le product owner gère plus qu'une équipe scrum en même temps, il fait implicitement partie de ces équipes et assiste parfois à leur réunion. Il est responsable de la création du backlog que nous décrirons bientôt. Ensuite, nous avons le *Scrum master* qui est l'équivalent d'un chef en gestion de projet. Ce dernier est là pour supporter l'équipe et s'assurer que les règles et principes agiles soient respectées et bien appliquées tout au long de la réalisation du projet. Par la suite nous avons l'équipe scrum à proprement parler (celle-ci inclut le *Product owner* et le *Scrum master* et est généralement composée de 2 à 9 participants. Cette équipe est généralement pluridisciplinaire et regroupe tout les experts spécialisés nécessaires pour mener à terme

le projet. Cette équipe peut également contenir un ou plusieurs utilisateurs qui sont chargés de supporter l'équipe de développement notamment au moyen de l'assurance de qualité (QA). Ces derniers peuvent être utiles pour donner des rétroactions tout au long du projet notamment sur l'adéquation du logiciel par rapport aux besoins réels des utilisateurs ou identifier des problèmes qui n'auraient pas été envisagés lors de l'étape de conception.

Comme on le constate, l'approche agile est beaucoup plus flexible que le modèle *waterfall* car les utilisateurs peuvent donner leurs commentaires tout au long du développement ce qui évite des mauvaises surprises tels que des annulation de contrat si le logiciel ne convenait pas au client final. Chaque projet agile débute par un premier sprint appelé Sprint 0 où l'équipe fait tout d'abord la vision du produit ce qui consiste à faire une maquette ou wireframe de ce qu'on doit livrer et d'énoncer les objectifs d'affaires. Par la suite, le *Product owner* doit réaliser une tâche très importante : le backlog du produit. Cette étape consiste à découper le projet en fonctionnalités décrites à l'aide d'un scénario utilisateur. Ces scénarios utilisateurs doivent contenir l'utilisateur concerné par la fonctionnalité, le besoin exprimé ainsi que la justification du besoin. Cette description servira ensuite à hiérarchiser les fonctionnalités selon leur importance pour les utilisateurs finaux ainsi que le temps pour procéder à leur réalisation. C'est généralement dans un Planning Poker que l'équipe évaluera les efforts requis pour livrer cette fonctionnalité. Ainsi, on pourra découper le projet en plusieurs sprints de durée fixe et formaliser le plan de livraison.

Chaque sprint commence par la mise en place d'un backlog du sprint qui définira en tâches les efforts requis pour livrer chaque fonctionnalité. Chacun des membres de l'équipe se verra assigner des tâches selon ses compétences techniques qui devront être terminées à la fin du sprint. Chaque jour l'équipe se rencontre pour un scrum quotidien de 10 à 15 minutes afin que chacun des membres ait une vision d'ensemble de l'avancement du projet ainsi que des difficultés rencontrées. Ces rencontres aident à la réalisation du *Burndown Chart* du sprint qui consiste en un rapport d'avancement sous forme graphique qui représente évolutivement les efforts restants en heures personnes à chaque jour pour la réalisation des objectifs du sprint. On peut également se servir des tableaux Kanban offerts dans plusieurs logiciels de gestion de projet tels que JIRA. On maintient également une liste des problèmes qui sera gérée par le *Scrum master* qui tentera de les résoudre. En cas d'imprévu on doit reporter les tâches non complétées à la prochaine étape. Chaque sprint se termine par une revue de sprint qui comprend une démonstration à un public élargi des fonctionnalités du logiciel ainsi qu'une rétrospective visant à améliorer le prochain sprint dans une optique d'amélioration continue. On peut également faire état des problèmes encourus au cours de cette rencontre qui dure généralement de 1 à 4 heures.