

Développement et mise en œuvre d'un processus de type agile-ISO 29110 au sein d'une grande institution financière Canadienne

FRANCIS PLANTE ET CLAUDE Y. LAPORTE

Résumé

La direction solutions trésorerie (DST), composée de 80 employés, est responsable du développement et de la maintenance des logiciels utilisés à la trésorerie d'une grande institution financière Canadienne. Chaque année, la DST observe une augmentation importante de son carnet de commandes pour ajouter, corriger ou modifier des fonctionnalités reliées aux applications offertes. Précédemment, le client se plaignait de la lourdeur du processus de développement et d'une augmentation importante du nombre d'incidents majeurs (panne de service) provoqués par des changements mis en production.

En réponse à cette problématique, nous avons évalué le processus de développement en comparant les activités du processus de développement (en mode maintenance) à celles du profil basique de la norme ISO/CEI 29110. Suite à l'évaluation de la promptitude au changement de la DST, nous avons établi que la DST était en mesure d'adopter l'agilité dans le cadre de ses projets de maintenance. Trois projets pilotes nous ont permis d'expérimenter le nouveau processus agile utilisant le profil Basique de l'ISO/CEI 29110. Les résultats obtenus ont démontré que les améliorations proposées et l'intégration de la méthode SCRUM ont contribué à réduire significativement le nombre et l'impact des incidents provoqués par des changements en production. Le client est maintenant ravi du nouveau système de planification, qui lui permet désormais de mieux gérer ses priorités et de connaître davantage l'état d'avancement de ses demandes. Le nouveau processus de type agile-ISO 29110 a été documenté et il est disponible à tous les gestionnaires de projets de la DST.

Mots clés

ingénierie du logiciel, normes, ISO/CEI 29110, très petits organismes, agilité ■

1. INTRODUCTION

Les méthodes agiles sont maintenant assez répandues et poussent les directions TI à revoir leur façon de développer et d'effectuer la maintenance des logiciels. Cette popularité de l'agilité a une incidence à plusieurs niveaux. La Direction solutions trésorerie (DST), d'une grande institution financière Canadienne, n'est pas étrangère à l'impact que cette méthode de développement peut avoir sur son processus d'embauche. Comme plusieurs employeurs du domaine bancaire, les responsables de la DST doivent composer avec les exigences strictes de l'autorité des marchés financiers, avec les vérificateurs internes et doivent également respecter les niveaux de service convenus avec les clients qu'elle dessert. En raison de nombreux problèmes répertoriés en production, des changements dans la méthode de développement devaient être considérés.

2. MISE EN CONTEXTE

L'institution financière comporte un service TI de 3650 employés qui doivent développer de nouvelles applications et qui doivent maintenir plus de 1250 applications. La DST est responsable du développement, de l'installation et de l'évolution des logiciels tels que « Swap », « Gestion des Valeurs mobilières » et « Titrisation et obligations sécurisées », qui sont utilisés par 80 employés de la trésorerie. On y trouve une dizaine d'applications qui sont sous la responsabilité de la direction et le budget annuel, pour la maintenance applicative, est d'environ 1,5 million de dollars.

Depuis la réorganisation de la gouvernance informatique de cette institution financière en 2011, qui a mené à la centralisation des services informatiques au sein d'un seul groupe informatique, la DST a dû revoir son processus de développement pour adopter les pratiques recommandées par le nouveau groupe.

Le processus plutôt rigide se prêtait bien à la réalité de l'institution financière, mais cadrait difficilement au contexte de la trésorerie. En effet, l'évolution rapide

des marchés, des normes et des produits financiers a un impact majeur sur la maintenance et sur les projets de développement logiciel. Il devenait difficile d'établir la portée initiale des projets puisque les besoins changeaient rapidement en cours de réalisation. Ceci se traduisait par la création de nombreuses demandes de changement et de révisions de la conception.

Chaque année, la DST observe une augmentation importante de son carnet de commandes pour ajouter, corriger ou modifier des fonctionnalités reliées aux applications en utilisation.

En réponse à cette problématique, la DST a décidé d'adapter son processus de développement et d'appliquer l'agilité dans ses projets de développement. Suite au succès observé avec l'adoption de la méthode SCRUM pour la conduite de trois projets pilotes en 2014, la direction désirait standardiser la méthode au sein de la DST pour mieux répondre aux exigences de la conformité et souhaitait également améliorer sa productivité en prévision de la prochaine année. L'objectif de la DST était de vérifier la capacité d'adoption de l'agilité spécifiquement dans un contexte de maintenance du logiciel. La maintenance occupe en 2016 près de 23% du portefeuille de projets de la DST et nous devons déterminer si les méthodes utilisées dans le cadre des autres projets pilotes pouvaient être adaptées pour répondre au besoin spécifique de l'équipe de maintenance de la DST.

► A - Description du mandat

Le mandat proposé par le directeur de la DST consistait à analyser le processus de développement utilisé à la DST et de confirmer la faisabilité d'instaurer l'agilité au sein de la direction pour l'exécution de tous ses projets de maintenance et de développement. L'analyse des forces et des faiblesses du processus en place était basée sur les pratiques proposées par le profil Basique de la norme ISO/CEI 29110 [1] et le modèle CMMI® [2]. Nous avons sélectionné l'ISO 29110 parce qu'elle a été rédigée spécifiquement pour répondre aux besoins des très petits organismes (TPO) qui n'ont généralement pas les ressources nécessaires (p.ex. l'expertise ou le temps) pour documenter, à partir d'une norme existante, un processus adapté à leur contexte. Le processus de développement du groupe informatique est bien documenté, mais ce processus est difficilement applicable au contexte de la trésorerie. L'utilisation d'une norme internationale telle que l'ISO 29110 a également permis à la DST de questionner son propre processus face aux meilleures pratiques, de l'améliorer en ajustant ses activités tout en se dotant d'un outil plus simple pour vérifier sa conformité à d'autres normes connues en génie logiciel comme l'ISO/IEC/IEEE 12207 [3].

► B - Objectifs d'affaires de la DST

La DST visait principalement à diminuer le nombre d'incidents majeurs liés aux changements effectués sur les environnements de production tout en améliorant la productivité de ses équipes de développement et la qualité des artefacts produits.

3. PLAN D'INTERVENTION

La portée du projet ne cible que les besoins de la direction solutions trésorerie. Les résultats obtenus avec le nouveau processus ont été présentés à la direction principale (DP) de la DST. Ils pourront être utilisés dans le cadre d'autres projets d'amélioration.

Ce projet d'amélioration s'est échelonné sur une période de 10 mois et a été découpé en 4 phases:

► A - Analyse de la capacité d'adoption de l'agilité

L'objectif de cette étape était d'évaluer le processus de développement utilisé à la DST et de rédiger une liste de recommandations basées sur les pratiques proposées par l'ISO 29110. Nous avons évalué la capacité d'adoption de l'agilité de type SCRUM des développeurs et des clients ainsi que la promptitude au changement technologique de la direction afin de faciliter la transition vers le nouveau processus de travail.

► B - Démarrage et conduite d'un projet pilote

L'objectif de cette étape était d'expérimenter la méthode SCRUM dans le cadre d'un projet de maintenance à la DST. Une formation de 15 heures a été dispensée à tous les participants et nous avons mis en œuvre la stratégie de gestion de changement technologique établie à l'étape d'analyse préliminaire.

► C - Expérimentation et amélioration du processus

L'objectif de cette phase était de mettre en pratique l'approche agile de type SCRUM dans un contexte de maintenance logicielle et d'intégrer des activités complémentaires qui permettraient d'améliorer la productivité globale de l'équipe et de bâtir ainsi une trousse de déploiement de qualité, basée sur l'expérimentation de la nouvelle méthode.

► D - Documentation finale

L'objectif de cette phase était de documenter la trousse de déploiement « ISO 29110 – Scrum/DST » ainsi que le journal de bord décrivant les étapes ayant mené à l'obtention du produit final.

4. ÉVALUATION DE LA CAPACITÉ D'ADOPTION DE L'AGILITÉ ET DU PROCESSUS DE DÉVELOPPEMENT LOGICIEL

Pour vérifier la capacité d'adoption de l'agilité, nous avons débuté par l'analyse du modèle d'affaires et nous avons vérifié la promptitude au changement technologique de la DST. Un modèle d'affaires « énonce comment un organisme gagne de l'argent en précisant où (et comment) il

se positionne dans son ou ses marchés » [5]. Ces éléments nous ont permis d'établir si l'environnement de travail de cette direction était propice à la mise en œuvre d'un changement et à l'utilisation de l'agilité.

Suite à l'analyse des résultats, nous avons confirmé la capacité d'adoption de l'agilité de la direction et proposé des recommandations quant à la façon d'intégrer l'agilité au sein du département. Nous avons indiqué les éléments à améliorer dans le processus de développement actuel et documenté la façon dont l'agilité pouvait répondre à ces éléments d'amélioration.

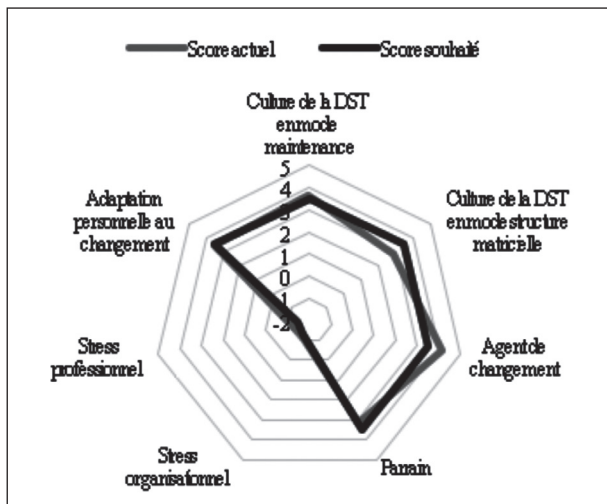
► A - Analyse du modèle d'affaires de la DST

La DST effectue des projets de développement logiciel à l'interne qui répondent aux besoins de la trésorerie de l'institution financière. La direction contribue parfois à la réalisation de projets qui s'adressent à ses clients mais cette situation ne représente pas la majorité des opérations régulières de la DST. Le choix de la méthode de développement du logiciel est largement influencé par le modèle d'affaires de l'organisation. En effet, les facteurs situationnels, qui sont associés à ce modèle d'affaires, nous ont permis de déterminer les contraintes et les priorités de l'entreprise qui devaient absolument être prises en compte lors d'un changement organisationnel.

► B - Évaluation de la promptitude au changement

La société américaine IMA [4] propose des outils qui nous permettent d'évaluer adéquatement les éléments qui peuvent influencer positivement ou négativement le succès de la mise en place d'un changement au sein d'une organisation (p.ex. les compétences des agents de changement et le niveau d'engagement du parrain). La figure 1 présente le graphique des pointages obtenus en comparaison avec le score souhaité pour chaque type d'évaluation de la promptitude au changement de la DST.

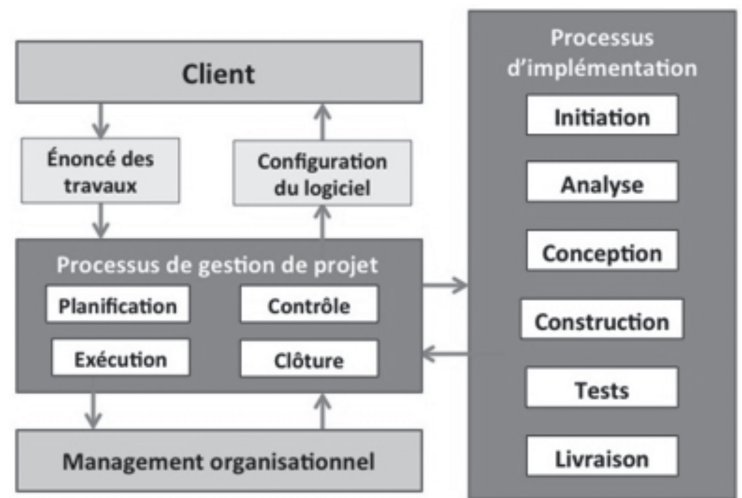
▼ *Figure 1 : Évaluation de la promptitude au changement de la DST*



Les évaluations effectuées nous montraient qu'il existait une bonne ouverture au changement, ce qui favorisait la mise en œuvre de notre projet au sein de la direction.

► C - Analyse du processus de développement actuel

L'objectif de cette section est de présenter l'analyse du processus existant au début de notre projet d'amélioration. Nous indiquerons ensuite des recommandations basées sur nos observations et nous comparerons le processus existant aux deux processus de l'ISO 29110 tel qu'illustré à la figure 2: un processus de gestion de projet (PM) et un processus de mise en œuvre du logiciel (SI).



▲ *Figure 2 : Les processus et les activités du profil Basique de la norme ISO/CEI 29110*

Pour dresser un portrait complet de la situation initiale, nous avons évalué, à haut niveau, les activités en lien avec ces deux processus et nous avons déterminé les améliorations à apporter pour assurer le développement d'une méthode SCRUM optimale pour la DST. Le processus de développement utilisé chez le groupe informatique est le processus PGP (c.-à-d. le Processus de gestion de Projet de l'institution financière). Ce processus rassemble à la fois les phases du processus de gestion de projet et celles du processus de mise en œuvre du logiciel. Le processus de gestion de projet influence les activités de l'ensemble des directions TI puisque la liste des points de contrôle utilisée par les vérificateurs internes et externes à la DST est basée sur la documentation du processus de gestion de projet.

Nous avons débuté par une analyse du PGP. Lorsque nous avons consulté la documentation du PGP, nous avons constaté que la gestion de projet est déjà à un stade très mature. Cette situation était en partie attribuable à l'existence du bureau de projet. Nous avons aussi constaté que les activités du processus PM documentées dans l'ISO 29110 peuvent toutes être associées à au moins une activité du processus PGP du groupe informatique.

Poursuivons avec l'évaluation du processus de gestion de projet de la DST en utilisant le profil Basique de l'ISO 29110 comme référentiel. Pour présenter un portrait plus juste de la situation, nous avons distingué les deux types de projets effectués à la DST:

- Projet de développement logiciel avec ou sans la présence du bureau de projet
- Projet de maintenance de logiciels

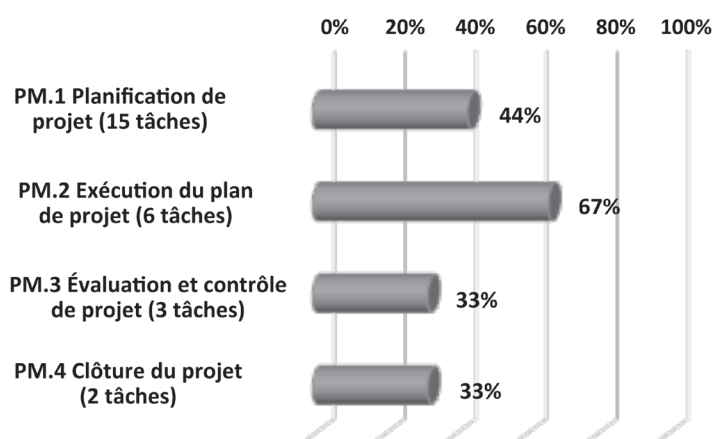
Les résultats de l'évaluation du processus PM de développement de la DST confirmaient que, mis à part quelques éléments mineurs, le processus était mature et bien maîtrisé.

La figure 3 présente les résultats de l'évaluation du processus PM de maintenance à la DST. On peut noter que l'exécution du processus fait défaut pour la plupart des activités recommandées par le profil Basique de l'ISO 29110. Ceci n'est guère surprenant puisque ce processus de gestion de projet n'est pas bien adapté à la maintenance logicielle de la DST. La portée du projet de maintenance n'était pas connue en début de projet et ceci avait un impact important sur la planification des activités et des ressources. Le carnet de commandes est typiquement composé de deux types de demande:

- Des demandes de correction de défauts nécessitant des interventions urgentes en production.

- Des demandes d'évolution de petite, moyenne et grande envergure (2 j/p à 50 j/p).

▼ Figure 3 : Évaluation du processus gestion de projet



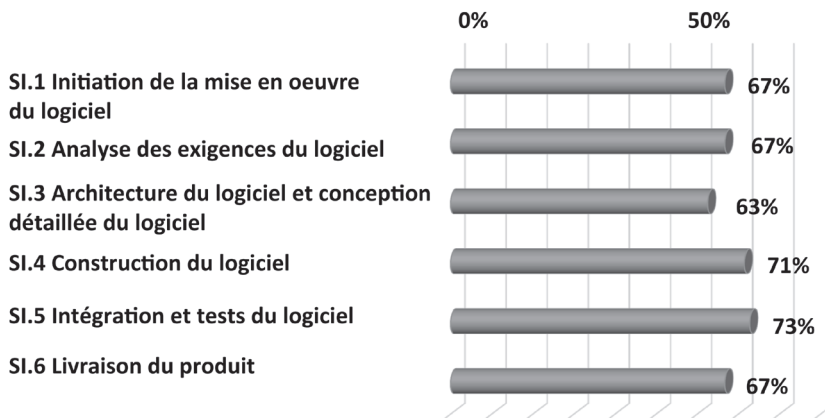
Le tableau 1 présente la liste des observations qui expliquent la faiblesse des résultats de l'évaluation.

Tâches de l'ISO 29110	Description de l'observation
PM.1.1 PM.1.2	Le client demande l'ouverture de plusieurs projets de maintenance sans indiquer précisément la nature et les objectifs de chacun des projets. Il n'y a pas de cahier des charges pour ce type de projet. La définition des travaux à effectuer pour chaque demande n'est pas documentée. Ainsi, pour une tâche en particulier, l'évaluation du travail à faire peut varier d'un programmeur à l'autre.
PM.1.3	Il n'est pas possible d'établir un plan de référence (<i>baseline</i>) à l'avance pour contrôler l'évolution des activités. Les demandes sont inscrites à la pièce et ne sont pas documentées avant le début des travaux. Comme les priorités ne sont pas établies à l'avance, le chef de projet est sollicité constamment par les ressources du projet (programmeurs, analystes et analystes en assurance qualité (AQ) pour déterminer quelles sont les prochaines tâches à effectuer.
PM.1.4 PM.1.7 PM.3.1 PM.3.2	L'effort nécessaire pour réaliser une demande est évalué avant le début des travaux. Mais l'évaluation quant à la durée de la tâche n'est pas spécifiée (date de début et date de fin).
PM.1.9	Les risques sont documentés, mais cette liste n'est pas mise à jour régulièrement en fonction de l'évolution du projet.
PM.1.10 PM.1.15	Il n'y a pas de stratégie documentée pour le contrôle des versions.
PM.1.11 PM.1.12 PM.1.13 PM.1.14	Le plan de projet est découpé selon les demandes (requêtes), mais il n'est pas détaillé en fonction des tâches à réaliser. Il n'est donc pas possible d'obtenir des mesures adéquates pour déterminer les efforts dépensés en fonction de la nature de la tâche.
PM.2.2 PM.3.3	Il n'existe pas un processus de gestion de changement clair.
PM.3.1	Le client n'est pas informé du statut de ses demandes et il ne sait pas à quel moment sa demande sera déployée en production.
PM.4.1 PM.4.2	Il n'existe pas un processus de clôture du projet pour les projets de maintenance.

Tableau 1 : Observations et évaluation du processus de gestion de projet en mode maintenance par rapport à l'ISO 29110

Nous avons ensuite évalué le processus de développement de la DST en utilisant le processus de mise en œuvre de l'ISO 29110 comme référentiel. La figure 4 présente les résultats de l'évaluation de ce processus.

▼ Figure 4 : Évaluation du processus de développement en mode maintenance



► D - Recommandations

Nous présentons dans cette section les recommandations que nous avons faites suite à notre analyse de faisabilité. Les résultats de l'analyse de la promptitude au changement indiquaient que le contexte favorisait la mise en place d'un changement au sein de la direction.

À première vue, l'utilisation de la méthode SCRUM pouvait aider à améliorer le processus de gestion de projet en mode maintenance et cadrerait bien à un contexte de projet où la portée n'est pas totalement connue à l'avance. Les activités du processus PM de l'ISO 29110 pouvaient être exécutées selon l'approche SCRUM. Ceci faciliterait l'implantation d'un cadre permettant une meilleure planification des activités.

Nous avons recommandé la mise en place d'un processus de déploiement pour diminuer le nombre d'incidents majeurs, pour faciliter la coordination entre les équipes de maintenance et pour répondre aux exigences des vérificateurs et à celles du département de gestion de changement.

La mise en place de la méthode SCRUM a été accompagnée par l'ajout d'activités au processus de développement utilisé à la DST. Nous y avons ajouté ces 2 tâches de vérification et de validation de l'ISO 29110 pour prévenir l'injection de défauts et pour corriger les erreurs dès la phase de développement:

- Documentation d'une liste de vérification permettant de vérifier chacun des artefacts produits dans le cadre du processus de développement.
- Ajout d'une étape de validation/vérification de tous les artefacts produits dans le cadre du processus de développement.

Nous avons recommandé de documenter et de mettre en place les éléments d'amélioration dans le cadre d'un projet pilote en maintenance de petite envergure (4-5 participants). L'exécution de ce projet nous a permis de recueillir les

commentaires des participants à chaque itération et d'améliorer graduellement le contenu du processus « ISO 29110 – Scrum/DST ».

5. MISE EN OEUVRE DES RECOMMANDATIONS

Nous présentons ici les étapes réalisées pour permettre la mise en place progressive de nos recommandations dans le cadre d'un projet pilote en maintenance. Nous présentons les étapes que nous avons réalisées pour introduire la méthode SCRUM et développer un nouveau processus ISO 29110 - Scrum/DST dans le cadre d'un projet pilote en maintenance.

► A - L'itération de démarrage du projet pilote

Le tableau 2 présente un diagramme utilisant la notation ETVX (Entry, Task, Validation, eXit) développé chez IBM [6] montrant les tâches ainsi que la liste des artefacts que nous avons réalisés dans le cadre de l'itération de démarrage du projet.

Itération de démarrage		
Document d'entrée	Tâches	Documents de sortie
• Budget autorisé	<ul style="list-style-type: none"> • Constituer l'équipe • Former l'équipe • Créer la charte de projet agile • Créer le carnet de produit • Prioriser et évaluer le carnet de produit • Présenter au Comité directeur • Créer le tableau de suivi • Planifier les déploiements et des activités SCRUM • Créer la définition de terminé 	<ul style="list-style-type: none"> • Charte de projet • Carnet de produit priorisé • Tableau de suivi des demandes • Calendrier de déploiement • Définition de terminé • Structure de répertoire (documentation) • Plan de projet
	<p>Validation</p> <ul style="list-style-type: none"> • Révision par les pairs de tous les documents de sortie 	

Tableau 2 : Processus de l'itération de démarrage

Le tableau 3 présente un sommaire des éléments que nous avons introduits pendant l'itération de démarrage pour contribuer à l'amélioration du processus de développement en mode maintenance. Nous indiquons les éléments de l'ISO 29110 visés par cette amélioration ainsi que la tâche où l'artefact est introduit dans le processus.

Tâches de l'ISO 29110	Description de l'amélioration apportée
PM.1.1 PM.1.2	L'introduction de la notion de « carnet de produit » permet au client de rassembler ses besoins et ses demandes à un seul endroit. Ce carnet de produit correspond à l'énoncé des travaux de l'ISO 29110 et permet de présenter la portée initiale du projet. Le carnet de produit initial est maintenant présenté à l'ensemble de l'équipe de développement. Cette activité a pour objectif de réviser le contenu, évaluer les tâches à effectuer et à commencer la planification des travaux. La création d'une « définition de terminé » indique clairement le travail à faire pour compléter une demande.
PM.1.3	Le carnet de produit initial permet de prioriser et d'évaluer les demandes. Ainsi, les membres de l'équipe connaissent les priorités en tout temps et ne dépendent plus du chef de projet pour planifier les activités.
PM.1.9	Les risques sont maintenant documentés dans la charte de projet « agile » et la liste des risques sera mise à jour lors de la présentation au comité directeur.
PM.1.10 PM.1.15	La stratégie de contrôle des versions est maintenant documentée dans la charte de projet « agile ».
PM.1.11 PM.1.12 PM.1.13 PM.1.14	La nouvelle charte de projet « agile » comprend maintenant l'ensemble des sections recommandées par l'ISO 2910.

Tableau 3 : Présentation des améliorations apportées avec l'introduction d'une itération de démarrage

► B - Les itérations de développement

Le tableau 4 présente un diagramme ETVX montrant les tâches que nous avons ajoutées dans le cadre des itérations de développement.

Itération de développement		
Documents d'entrée	Tâches	Documents de sortie
<ul style="list-style-type: none"> • Récits documentés • Carnet de produit évalué et priorisé • Définition de terminé 	<ul style="list-style-type: none"> • Évaluer la vélocité et préparer le carnet de l'itération • Créer le tableau de suivi des demandes • Exécuter les mêlées quotidiennes • Réaliser les demandes • Préparer les critères d'acceptation et les cas d'essais 	<ul style="list-style-type: none"> • Nouvel incrément du produit logiciel • Nouvelle version améliorée du processus « ISO/CEI 29110 – Scrum/DST »
	<p style="text-align: center;">Validation</p> <ul style="list-style-type: none"> • Exécution des cas de test par un pair • Révision par les pairs des critères d'acceptation • Approbation des critères d'acceptation par le client • Vérification des éléments de la définition de terminé • Revue de l'itération avec les parties prenantes 	

Tableau 4 : Processus d'une itération de démarrage

Le tableau 3 présente un sommaire des éléments de la méthode SCRUM que nous avons introduits pendant les itérations de développement et qui ont contribué à l'amélioration du processus de développement en maintenance de la DST. Nous indiquons les éléments du profil Basique de l'ISO 29110 visés par cette amélioration ainsi que l'activité ou l'artéfact introduit dans le processus.

Tâches du profil Basique	Description de l'observation
PM.1.2	La création d'un processus de déploiement mensuel a réduit le nombre de déploiements, a réduit les risques d'erreurs et a amélioré les directives de livraison en indiquant spécifiquement les composants qui seront livrés.
PM.1.4	Avec une planification par itération, le client est en mesure de déterminer facilement à quel moment une demande sera livrée.
PM.2.2 PM.3.3	Il existe maintenant un processus de gestion du changement. Pour ajouter ou retirer des éléments de l'itération courante, le client doit préalablement communiquer avec le « Scrum master ». L'équipe et le client ont été sensibilisés au processus de gestion de changement. Un changement dans un récit ayant un impact sur l'évaluation initiale de l'équipe doit maintenant être réévalué avant d'être réalisé.
PM.3.1	Le client est informé quotidiennement de l'avancement des travaux.
SI.1.2	L'équipe de maintenance peut maintenant utiliser un environnement dédié à ses projets (DEV et AQ).
SI.2.3 SI.2.6 SI.3.4 SI.3.6 SI.5.8 SI.6.4	Avec l'intégration d'une tâche de revue par les pairs et l'ajout de listes de vérification permettant de vérifier les artefacts, l'équipe est maintenant sensibilisée à l'importance d'assurer la qualité de tous les éléments développés.
SI.3.7 SI.5.4 SI.5.6	Dorénavant, les défauts sont tous enregistrés dans le carnet d'itération (JIRA) et il est possible de faire le suivi des tests en temps réel grâce à l'intégration du tableau KANBAN de l'AQ.
SI.3.4 SI.3.7 SI.5.6	Nous avons ajouté des tâches permettant la traçabilité des exigences: <ul style="list-style-type: none"> • L'intégration des critères d'acceptation. • La création de la définition de « terminé » pour un récit. • Le tableau de suivi de l'itération
SI.5.7 SI.5.8 SI.6.4	L'intégration d'une définition de « terminé » pour une itération (et pour un récit) de même que l'ajout des activités de vérification, permet dorénavant d'assurer un maintien adéquat de la documentation existante.
SI.4.4	Les tests unitaires ont été ajoutés à la définition de « terminé » des récits. Lorsqu'un programmeur effectue une modification dans une région de code qui ne contient pas de tests unitaires, il doit maintenant ajouter un test supplémentaire. Cependant, si l'ajout d'un test unitaire dans du code existant est plutôt complexe et nécessite une modification au niveau de la conception, il ajoutera une tâche dans le carnet de produit. Cette tâche sera priorisée par le responsable du carnet de produit et évaluée par l'équipe.
SI.6.6 SI.5.3	Nous avons réduit le nombre de déploiements en production et réduit considérablement le nombre d'incidents liés aux changements en production.
PM.1.15	L'intégration d'un analyste AQ responsable de la vérification des artefacts ainsi que l'ajout d'une « définition de terminé » pour les itérations (et pour les récits), facilite grandement le processus de vérification interne.

Tableau 5 : Présentation des améliorations apportées au cours des itérations de développement

6. RÉSULTATS DU PROJET PILOTE

L'intégration des principaux éléments de la méthode SCRUM au sein de l'équipe de maintenance de la DST peut définitivement être considérée comme un succès. Après 6 mois d'expérimentation, l'équipe est beaucoup mieux structurée et elle nécessite moins d'intervention et de contrôle de la part du chef de projet (c.-à-d. que l'équipe est auto-organisée).

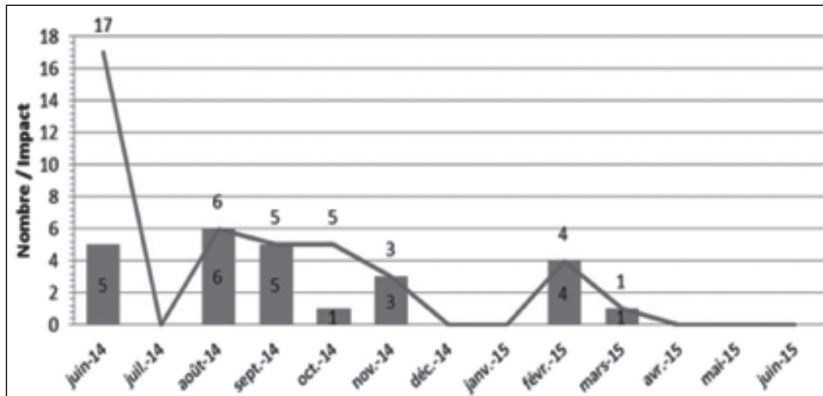
Toutes les améliorations que nous avons apportées au processus de développement et que nous avons documentées

dans le processus «ISO 29110 – Scrum/DST» ont contribué à diminuer significativement le nombre d'incidents majeurs en production et à augmenter la satisfaction du client et du groupe de gestion des changements.

► Diminution du nombre d'incidents majeurs

La figure 5 montre la forte diminution du nombre d'incidents majeurs causés par les changements en production. Depuis la mise en place du processus en 2015 et après l'exécution de 7 déploiements, il n'y a eu aucun incident majeur

répertorié en production causé par les changements de l'équipe de maintenance. Les colonnes indiquent le nombre de services impactés et la ligne montre l'impact des incidents. L'impact est évalué en fonction de la criticité, sur une échelle de 1 à 5 (où le chiffre 5 représente le plus gros impact), pour le service touché.



▲ Figure 5 : Historique du nombre de services de la trésorerie impactés

Les incidents majeurs provoqués en 2015 ne sont pas reliés à des déploiements applicatifs.

► Évaluation de la satisfaction de la clientèle

Nous avons conduit un sondage de la clientèle pour mesurer le niveau de satisfaction de notre principal client suite à la mise en place de la méthode SCRUM.

Avant la mise en œuvre du nouveau processus agile-ISO 29110, nos clients avaient soulevé des problématiques au sujet du processus de développement pour les projets de maintenance:

- Impossibilité de connaître le statut d'avancement pour une demande particulière.
- Il y a presque toujours un incident déclaré lors d'un changement en production.
- Il y a toujours un très grand nombre d'anomalies répertoriées en AQ.

Suite à l'analyse des résultats du sondage, nous avons constaté une nette amélioration de la satisfaction du client à plusieurs niveaux: la confiance envers le processus de livraison a augmenté, le nombre d'anomalies répertoriées en AQ a diminué et le client est beaucoup mieux informé quant au statut d'avancement des demandes.

► Augmentation de la satisfaction du groupe de gestion des changements

Le résultat du sondage de satisfaction indique un succès sur toute la ligne. Antérieurement, les relations entre l'équipe de développement et l'équipe de gestion de changement étaient plutôt tendues. Le nombre de demandes de mise en production (MEP) rejetées par l'équipe de changement était

très élevé et ceci avait contribué à la baisse de productivité de l'équipe puisque les analystes/programmeurs devaient continuellement réviser leurs demandes.

Depuis la mise en place du processus de déploiement, nous avons noté une diminution du nombre de demandes

rejetées et une augmentation de la qualité des demandes de déploiement. Les équipes de gestion du changement sont avisées des MEP bien avant les délais d'annonce et le système de coordination entre les équipes a contribué à diminuer significativement le nombre de changements nécessaires.

7. LEÇONS APPRIS

Notre projet d'amélioration du processus de maintenance nous a permis de constater qu'il est possible combiner l'approche

SCRUM et les activités et les tâches du profil Basique de l'ISO 29110. Dans un contexte favorable au changement et à l'application de l'agilité, nous avons démontré que l'approche SCRUM est tout à fait indiquée pour la conduite de projet de maintenance. L'agilité ne doit cependant pas être considérée systématiquement comme une solution à tous les problèmes et l'approche SCRUM doit être appliquée de manière rigoureuse afin d'éviter de tomber dans ce que nous pourrions appeler le piège de l'agilité. En effet, l'approche SCRUM est un guide qui propose des activités de gestion de projet qui répondent bien au contexte de maintenance. Mais, le succès de son application dépend d'un certain nombre de facteurs qu'il ne faut pas négliger. Par exemple, il ne faut pas oublier que le client a un rôle prépondérant à la réussite d'un projet agile. Si le responsable du carnet de produit n'a pas la connaissance adéquate du domaine d'affaires, il est clair que l'instauration de l'agilité se traduira par un facteur de risque pour le projet.

La maturité du processus de développement utilisé pour la réalisation des itérations est très importante et c'est pourquoi l'utilisation de l'ISO 29110 a été très utile lors de notre transition vers l'agilité. Comme l'évaluation des activités de développement est indépendante de l'approche utilisée, il est possible d'améliorer le processus de développement et d'intégrer ensuite l'approche SCRUM.

Il a été essentiel d'adapter le cadre de développement SCRUM en fonction du contexte et des contraintes que l'on retrouvait au sein de l'institution financière. Toutes les adaptations apportées à la méthode ont été documentées dans le nouveau processus ISO 29110 - SCRUM/DST.

8. CONCLUSION

Nous avons développé une approche agile tout en mettant en œuvre les deux processus du profil Basique de l'ISO 29110. Nous avons appris aussi qu'il est difficile d'appliquer systématiquement l'agilité pour la conduite de tous les

projets logiciels de la DST. La combinaison de l'approche SCRUM et de l'ISO 29110, a permis à la DST d'améliorer significativement son processus de maintenance et de livrer des logiciels de qualité qui répondent parfaitement aux besoins des clients. Depuis 2016, l'équipe de 5 personnes, qui est responsable du support des applications de la trésorerie, utilise la même approche que l'équipe de développement pour le traitement des demandes non-urgentes (p.ex. les problèmes récurrents de production). Un seul carnet de produit est utilisé pour prioriser les demandes (incluant les problèmes de production) des clients. Cette initiative nous a permis de répondre aux éléments qui ne sont pas seulement prioritaires pour un individu, mais pour tous les groupes de la trésorerie de la grande institution financière canadienne.

RÉFÉRENCES

- [1] ISO/CEI TR 29110-5-1-2 :2011, Ingénierie du logiciel - Profils de cycle de vie pour très petits organismes (TPO) - Partie 5-1-2 : Guide de gestion et d'ingénierie : Groupe de profil générique: Profil Basique, Disponible gratuitement de l'ISO: <http://standards.iso.org/itf/PubliclyAvailableStandards/index.html>
- [2] CMMI®-DEV, « CMMI® pour le développement, Version 1.3 », Software Engineering Institute. CMU/SEI-2010-TR-033, 2010.
- [3] ISO/IEC/IEEE 12207:2008, « Systems and software engineering -- Software life cycle processes », Organisation de normalisation internationale, Genève, Suisse.
- [4] Implementation Management Associates Inc., 13949 West Colfax Avenue Bldg 1 Suite 108 Lakewood, CO 80401, États-Unis. <http://www.imaworldwide.com>
- [5] Claude Y. Laporte et Alain April : *L'assurance qualité logicielle 1*, Lavoisier, Paris, 383 p., 2011
- [6] R. A. Radice, N. K. Roth *et al.* : A Programming Process Architecture ; IBM Systems Journal, 24(2), p. 79-90, 1985.

BIOGRAPHIES



Francis Plante est, depuis avril 2016, directeur de la direction solution trésorerie (Groupe technologie d'une grande institution financière canadienne). Il travaille au sein du département informatique de cette institution financière depuis 15 ans à titre d'analyste fonctionnel et de chef de projets. Il a obtenu une maîtrise (M.Ing) de l'École de technologie supérieure (ÉTS) en 2015.



Claude Y. Laporte est, depuis l'an 2000, professeur agrégé de génie logiciel à l'École de technologie supérieure (ÉTS). L'ÉTS est une école d'ingénieurs, de plus de 11 000 étudiants. Il a effectué ses études de premier cycle en physique et mathématique au Collège militaire royal de Saint-Jean. Il a obtenu un doctorat (Ph.D.) de l'Université de Bretagne Occidentale. La faculté d'ingénierie et d'architecture de l'Universidad de San Martin de Porres, de Lima au Pérou, lui a décerné un Doctorat Honoris Causa. Il est l'éditeur du projet de normalisation ISO/CEI 29110 depuis 2005. Il est membre de l'Ordre des ingénieurs du Québec, de l'IEEE Computer Society, du PMI et de l'INCOSE.

Il est co-auteur de deux ouvrages sur l'assurance qualité logicielle : l'assurance qualité logicielle 1 - Concepts de base et l'assurance qualité logicielle 2 - Processus de support, publiés chez Hermès-Lavoisier en 2011. Il est co-auteur d'un ouvrage en anglais sur l'assurance qualité logicielle. Cet ouvrage sera publié en 2017 par l'IEEE Press-Wiley.

A PRIMER FOR MODEL-BASED SYSTEMS ENGINEERING

David Long et Zane Scott

La société Vitech Corporation annonce la parution de la seconde édition de son « abécédaire » d'ingénierie de systèmes guidée par les modèles. Il est disponible gratuitement en version électronique.

Contents: The primer addresses the basic concepts of model-based systems engineering. It covers the mode, language, behavior, process, architecture, and verification. It is a call to the consideration of the foundational principles behind those concepts.

It is not designed to present novel insights into MBSE so much as to provide a guided tour of the touchstones of systems design. It is suitable for use by a systems engineers new to the practice of "model-based" systems engineering, an experienced systems engineer who has been introduced to "model-based" concepts in an ad hoc fashion or by any professional knowledgeable of systems thinking and practice. It is not intended as a comprehensive guide or as practice handbook.

www.vitechcorp.com/ressources/mbse.shtml