GPSI - AUDIT DE PROJETS AGILE

CONTRIBUTEURS

Clément Boulard, Senior Manager I Tech Risks Financial Services - KPMG

Clément dispose de 12 ans d'expérience acquis dans la supervision d'audits IT ainsi que l'accompagnement au renforcement des dispositifs de maitrise des risques IT dans le secteur bancaire et de l'assurance. Diplômé du master en audit et conseil en Systèmes d'Information de l'Université Paris-Dauphine, Clément a précédemment évolué au sein d'un département d'audit interne d'un groupe bancaire international en France et aux Etats-Unis.

Valérie Schipman, Directrice Audit Interne IT & Digital – Renault Group

Membre du groupe professionnel systèmes d'information de l'IFACI et responsable du club des directeurs d'Audit interne IT de 18 grands groupes français, Valérie a plus d'une vingtaine d'années d'expérience IS/IT en France et à l'international, dans les secteurs bancaire et industriel.

Olivier Meyer, Directeur IT Group Internal Audit – The Adecco Group

Olivier est directeur de l'audit des systèmes d'information du groupe Adecco et membre du groupe professionnel systèmes d'information de l'IFACI. Diplômé de l'Institut Mines-Telecom Business School, Olivier a auparavant été senior manager au sein des équipes IT Advisory chez KPMG.

Guillaume Cuisset, Associé Technology Risks - KPMG

Guillaume dispose de 15 ans d'expérience dans le domaine des risques IT et est en charge des services d'assistance aux directions d'audit et contrôle interne IT pour KPMG France. Il est diplômé d'un master en audit et Conseil en Systèmes d'information à Paris Dauphine et membre du groupe professionnel systèmes d'information de l'IFACI.

Jean-Christophe Glise, Head of Digital Solutions Audit - Sanofi

Jean-Christophe dispose de 20 ans d'expérience dans le domaine des risques et de l'audit IT, de la gouvernance et de l'IT Advisory. Il a commencé sa carrière professionnelle chez Arthur Andersen. Aujourd'hui, Jean-Christophe est Directeur à l'Audit Interne de Sanofi – Digital & IT.

Olivier Sznitkies, CISA, CISM, CFE - Audiligence

Olivier dirige Audiligence, société de conseil et formation spécialisée en audit/contrôle interne, gestion des risques, prévention de la fraude et cybersécurité. Auparavant, Olivier a été Directeur de l'audit interne EMEA du Groupe LafargeHolcim et a également une expérience de plus de 15 ans comme Auditeur SI chez KPMG. Olivier a contribué à la rédaction de nombreuses publications comme le «Guide d'Audit de la Gouvernance des SI» et «Le Dirigeant Financier face à la Cyberfraude». Olivier est ingénieur informatique. Il est également auditeur de l'IHEDN session nationale «Souveraineté numérique et cybersécurité».

Philippe Hervias, Head of Digital & IT Audit – Sanofi

De formation ingénieur, Philippe cumule 35 ans d'expérience dans la cryptographie et la cyber sécurité puis dans l'audit des systèmes d'information. Philippe est également président du groupe professionnel systèmes d'information (GPSI) de l'IFACI.

Daniel Benveniste, Deputy Head of IS audit for Wholesale Banking perimeter – Société Générale

Daniel est adjoint au directeur de l'audit des Systèmes d'Information pour le périmètre Banque de Grande Clientèle et Solutions Investisseurs de la Société Générale. Diplômé du Master 242 de Dauphine, Daniel était auparavant directeur des audits applicatifs chez Sanofi.

Yohann Vermeren, Partner KPMG, Gouvernance, Risques et Conformité

Yohann accompagne les directions d'audit et du contrôle internes dans le cadre de l'exécution de leur mission et de leur transformation. Il travaille aussi sur les enjeux de maitrise des risques, de la transformation des organisations et la conformité réglementaire. Il est d'autre part Vice-Président du groupe professionnel système d'information (GPSI) de l'IFACI.

GPSI - Audit de projets agile

Table des matières

Intro 2
Qu'est-ce que la méthodologie « Agile » ? 3
1. Genèse de l' « Agile »: 3
2. Présentation des principales méthodologies et principes agiles 4
3. Principaux rôles autour de l'agile 5
4. Outillage 5
Agile vs Waterfall : quelles sont les principales différences 6
Les risques spécifiques aux démarches agiles 8
1. Facteur de risque non lié à un principe agile 9
Risque 1 – Appeler agile une méthodologie qui ne l'est pas 9
2. **Principe 1** : Nous valorisons les individus et leurs interactions plus que les processus et les outils 9
Risque 2 – Inadéquation des compétences 9
Risque 3 – Gestion inadéquate des ressources 10
Risque 4 – Outils non adaptés 10
3. **Principe 2** : Nous valorisons des logiciels opérationnels plus qu'une documentation exhaustive 11
Risque 5 – Absence de documentation de référence 11
4. **Principe 3** : Nous valorisons la collaboration avec les clients plus que la négociation contractuelle 11
Risque 6 – Manque de représentativité / légitimité du Product Owner 12
Risque 7 – Organisation non adéquate 12
5. **Principe 4** : Nous valorisons l'adaptation au changement plus que le suivi d'un plan 12
Risque 8 – Incompatibilité avec le type de projet 12
Risque 9 – Absence de périmètre et objectifs clairs du projet 13
Risque 10 – Dérive de budget 13
Risque 11 – Processus connexes au développement de projet non adapté à l'agilité 14
Synthèse / conclusion 14
Annexes 15
1. Annexe 1 : 12 principes du « Manifeste Agile » 15
2. Annexe 2: Agile à l'échelle 15
3. Annexe 3: Glossaire 15

Intro

Depuis une dizaine d'années, la méthodologie de gestion des projets et programmes liés aux systèmes d'information voit une recrudescence des démarches dites « Agile » notamment dans la mise en œuvre de transformation digitale.

Ces nouvelles démarches ont été définies et déclinées afin de pallier les limites des méthodologies projet dites « en cascade » (Waterfall), en particulier en termes de temps de mise à disposition d'une première version de la solution cible aux utilisateurs ou aux clients. Elles ont entrainé la définition de nouvelles méthodes, de nouvelles organisations, de nouveaux rôles et d'un nouveau cadencement.
Les équipes projets doivent donc changer d'état d'esprit, modifier fondamentalement leurs habitudes de travail, s'approprier et adapter des démarches et méthodes agiles. Cette adaptation, si elle est mal maitrisée, est peut-être propice à une insuffisance de contrôle et à la matérialisation de risques associés.

Que ce soit pour pénétrer de nouveaux marchés, développer de nouveaux produits ou gérer des alliances et des acquisitions, les organisations et consacrent des budgets considérables à leur transformation digitale. La mise en échec de projets agiles peut donc représenter un risque important pour les entreprises et institutions qui adoptent ce type de méthodologie.

L'audit interne peut réaliser une revue indépendante des projets, que ce soit sous l'angle de la conformité avec les bonnes pratiques, ou dans une dimension plus stratégique (gouvernance, dérive du planning et des budgets, respect du business case, et des objectifs initiaux du projet).

Ce document a pour objectif d'apporter aux équipes d'audit interne un éclairage sur les démarches agiles mises en œuvre par les entreprises et institutions dans le cadre de la gestion de leurs projets informatiques, les spécificités et nouveaux concepts de ces démarches, les différences entre les méthodologies en cascade et les démarches agiles ainsi que sur les risques afférents à ces démarches agiles dans le cadre de projet SI. Le champ de la gestion de projet agile étant extrêmement vaste, l'objet de ce guide est d'apporter un éclairage sur les spécificités de ces démarches et les risques induits par celles-ci, de donner des exemples de risques courants liés à la gestion de projet et d'étudier comment l'audit interne peut contribuer à limiter ces risques par à l'examen indépendant des projets agiles.

Ce document traite d'un projet conduit en mode agile, certaines organisations ont par ailleurs déployé ce concept agile sur l'ensemble de leur gouvernance et portefeuille projet. Ce concept d'« Agile à l'échelle » est présenté en annexe de ce document. Néanmoins, les risques et méthodologies d'audit des dispositifs d'agile à l'échelle ne sont pas abordés dans ce guide.

Qu'est-ce que la méthodologie « Agile » ?

1. Genèse de l'« Agile »:

Avant d'en apercevoir les spécificités, nous vous proposons de revoir la genèse ayant amené à la création des démarches agiles.

Création du « Manifeste Agile »

Le développement logiciel agile a été popularisé par le « Agile Manifesto », défini dans un objectif d'une créativité accélérée, moins bureaucratique que les méthodes traditionnelles de développement de logiciels en cascade. Le « Manifeste Agile » a été rédigé en février 2001 par un groupe de 17 spécialistes, dont les créateurs de Scrum, Extreme Programming (XP), Dynamic Systems Development Method (DSDM) et Crystal, une représentation du développement piloté par les fonctionnalités, et plusieurs autres leaders d'opinion de l'industrie du logiciel. Le « Manifeste Agile » a établi un ensemble commun de valeurs et de principes généraux pour toutes les méthodologies agiles. Cet ensemble de 4 valeurs et 12 principes, s'ils étaient acceptés et mises en œuvre, entraîneraient un passage des méthodologies de développement traditionnelles à des méthodologies plus flexibles dans le monde informatique.

> « Nous découvrons comment mieux développer des logiciels par la pratique et en aidant les autres à le faire. Ces expériences nous ont amenés à valoriser :
>
> 1. **Les individus et leurs interactions** plus que les processus et les outils
> 2. **Des logiciels opérationnels** plus qu'une documentation exhaustive
> 3. **La collaboration avec les clients** plus que la négociation contractuelle
> 4. **L'adaptation au changement** plus que le suivi d'un plan
>
> Nous reconnaissons la valeur des seconds éléments, mais privilégions les premiers. »[1]

L'« Agile Alliance » a été officiellement créée à la fin de 2001 en tant qu'organisation à but non lucratif reconnue par l'industrie et engagée à soutenir la communauté agile. Cette alliance vise à aider à l'adoption des pratiques agiles et à améliorer la pratique agile ainsi qu'à explorer et partager les idées et les expériences.

Adoption devenue grand public

Les équipes et les organisations ont commencé à adopter l'agilité, principalement dirigées par des personnes effectuant le travail de développement au sein des équipes. Progressivement, les managers de ces équipes ont également commencé à introduire des approches agiles dans leurs organisations. Ces méthodologies ont mis l'accent sur une collaboration étroite entre l'équipe de développement et les parties prenantes de l'entreprise ; livraison fréquente de valeur commerciale, équipes serrées et auto organisées ; et des moyens intelligents de créer, de confirmer et de livrer du code. Des frameworks tels que Scrum, Extreme Programming, Feature-Driven Development (FDD) et Dynamic Systems Development Method (DSDM), entre autres, ont commencé à apparaître. La diffusion des idées à cette époque était très organique, et toutes ces différentes approches ont commencé à se développer de manière très locale. Les gens ont emprunté les cadres originaux et les ont modifiés avec différentes pratiques pour les adapter à leurs propres contextes.

Au cours des dernières années, les méthodes agiles sont passés du domaine informatique aux parties non technologiques de l'entreprise, soutenant le mandat plus large de l'agilité à l'échelle de

[1] **Source**: Agile Manifesto, 2001 - https://agilemanifesto.org/iso/fr/manifesto.html

l'organisation, en mettant l'accent sur la culture et l'état d'esprit pour atteindre les objectifs commerciaux. Avec l'adoption du Cloud et du DevOps, les méthodes agiles ont été mélangées au développement itératif.

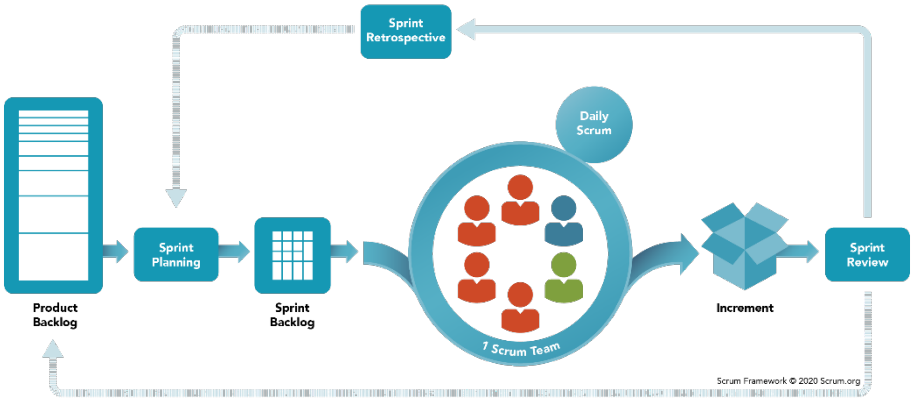
2. Présentation des principales méthodologies et principes agiles

La démarche agile a été déclinée selon différentes formes et méthodologies que l'auditeur interne devra connaitre et distinguer afin d'en percevoir les risques sous-jacents.

S'appuyant sur les valeurs et principes, de nombreuses méthodologies et cadres ont été développés, allant de Scrum et Kanban, à SAFe, LeSS et le modèle Spotify.

Méthodologies agiles :

- **KANBAN :** Contrairement à d'autres frameworks agiles, le développement Kanban ne nécessite pas d'itérations fixes. Le travail progresse tout au long du processus de développement sous la forme d'un flux continu d'activités. Kanban se concentre sur la visualisation du travail, la limitation du travail en cours et la garantie que le travail continue de s'écouler.
- **SCRUM :** Scrum est un cadre de gestion léger avec une large applicabilité pour la gestion et le contrôle de projets itératifs et incrémentiels de tous types. Scrum est un cadre agile pour la gestion de projets où le travail est engagé pour chaque itération, appelés sprints. Les équipes Scrum livrent souvent, se concentrent sur des produits prêts et ont des rôles clairement définis tels qu'un propriétaire de produit ou un scrum master.[2]



	Description	Créateur	Date de création
SAFe (Scaled Agile Framework)	Définit les pratiques Lean-agile au niveau de l'entreprise en définissant trois segments : Équipe, Programme et Portefeuille	Dean Leffingwell (Scaled agile Inc.)	2011
LeSS (Large-Scale Scrum)	Est un cadre pour faire évoluer Scrum jusqu'à huit équipes ou jusqu'à quelques milliers de personnes pour un produit (LeSSHuge)	Greg Larman& Bas Vodde (The LeSSCompany B.V.)	2006
DAD (Disciplined Agile Delivery)	Est une boîte à outils organisée en quatre couches de portée qui visent à aider les équipes des organisations à rationaliser leurs processus de manière contextuelle ;	Scott Ambler & Mark Lines (Disciplined Agile Consortium)	2012

[2] **Source**: Scrum Framework, 2020 - https://www.scrum.org/resources/

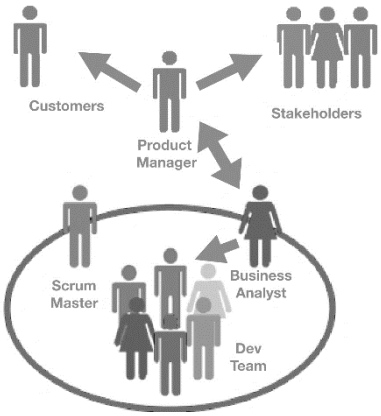
Scrum@ Scale	Vise à mettre en place une « bureaucratie minimale viable » via une architecture « sans échelle »	Jeff Sutherland	2018
Scrum Nexus	Est un ajout (et recouvre) le cadre Scrum agile existant	Ken Schwaber (Scrum.org)	2015
Modèle "Spotify"	Est une approche autonome et axée sur les personnes pour déployer l'agilité à grande échelle qui met l'accent sur l'importance de la culture et du réseau.	Henrik Kniberg et Anders Ivarsson (Scaling Agile @ Spotify)	2012

3. Principaux rôles autour de l'agile

Les démarches agiles ont engendré la création de rôles propres et spécifiques dont l'auditeur devra percevoir les objectifs et responsabilités.

La méthode Scrum définit trois rôles distincts : **Product Owner**, **Scrum Master** et **équipe opérationnelle**. L'établissement d'une relation de confiance permet de fédérer tout un chacun vers l'accomplissement d'un but unique et commun : l'aboutissement du projet qui est confié à l'équipe.

-
- **Le Product Owner** est le représentant des clients et utilisateurs. C'est lui qui définit l'ordre dans lequel les fonctionnalités seront développées et qui prend les décisions importantes concernant l'orientation du projet.
- **Le Scrum Master** est le garant de la méthodologie Scrum auprès de l'équipe fonctionnelle. Facilitateur, animateur, isolateur… Autant de rôles qu'il doit tenir pour s'assurer que le travail progresse correctement. Un Scrum Master n'est en aucun cas un chef d'équipe. Il est le garant de la bonne atteinte des objectifs par l'équipe et du bon respect des jalons.
- **L'équipe opérationnelle** est auto-gérée et n'a aucune notion de hiérarchie interne : toutes les décisions sont prises ensemble et personne ne donne d'ordre à l'équipe sur sa façon de procéder. Elle est composée de 3 types de profils : les développeurs, le testeur technique, le testeur métier.

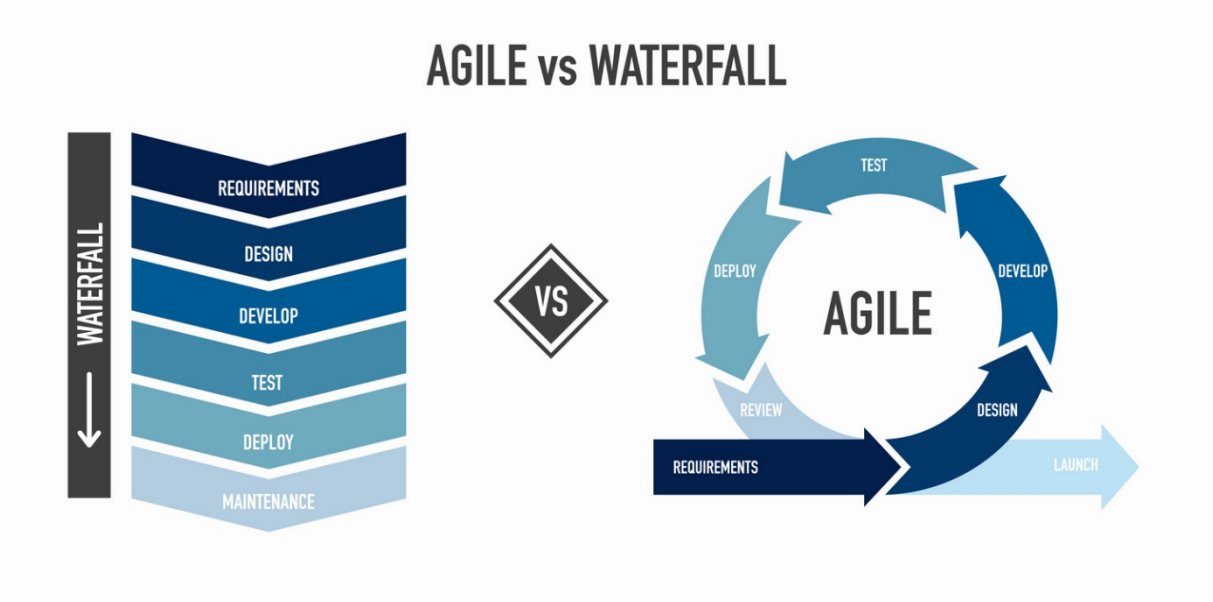


4. Outillage

Dans le cadre de la gestion des démarches agiles différents types d'outil ont été créés :

- Outils agiles en gestion de projet ;
- Outils d'analyse des indicateurs de performance ;
- Outils de brainstorming, roadmaps et autres ;
- Logiciels de collecte des feedbacks utilisateurs ;
- Outils de collaboration des équipes.

Agile vs Waterfall : quelles sont les principales différences



	Approche traditionnelle (Waterfall)	Approche agile
Principe fondamental	Méthode séquentielle et linéaire. L'approche traditionnelle repose sur un cycle de développement séquentiel linéaire	Méthode itérative, incrémentale et adaptative. L'approche agile repose sur des itérations continues de développements et de tests
Planification	Une définition des spécifications fonctionnelles et techniques détaillées est réalisées Rédaction d'un cahier des charges précis.	Définition des fonctionnalités à développer lors de chaque phase de développement.
Phases du projet	Dans le cadre d'un projet traditionnel, les phases se succèdent sans retour en arrière.	Les mêmes phases peuvent se reproduire dans le cadre d'un projet agile.
Calendrier	Le calendrier est défini à l'avance de manière précise dès l'initiation du projet.	Le calendrier est une suite de phase de développement et de livraison de fonctionnalités opérationnelles.
Participation des clients	Une fois que l'objectif final est établi, l'approche traditionnelle implique peu le client du projet pendant le processus de développement à l'exception de contrôles spécifiques ou pour les livrables. Le déroulement du projet est défini dès le départ, les commentaires des clients sont intégrés à l'issue des développements (validations intermédiaires ou finales)	Un élément fondamental de l'approche agile est d'inclure les clients dans le développement du projet à chaque étape. Les clients sont impliqués et donnent un retour à l'équipe de développement au fur et à mesure qu'ils progressent dans les différentes phases du projet.
Budget	Le budget des projets utilisant l'approche traditionnelle est généralement fixe.	L'approche agile est ouverte à l'adaptation, encourage l'expérimentation et accueille les changements de direction, même dans les phases ultérieures du projet. De ce fait, le budget a tendance à être plus flexible.
Phase de tests	Située à la fin des différentes phases elles sont limitées dans le temps.	Elles prennent place tout au long du projet, lors de chaque itération.

	Approches traditionnelles	Approches agiles
Points forts	**Un projet clair et planifié** Le projet est défini par un cahier des charges et un calendrier précis. Le périmètre est défini et validé lors des phases amont du projet et les livrables attendus lors du projet sont connus. **Une responsabilité centralisée permettant un niveau de contrôle élevé** La responsabilité du projet est détenue par le chef de projet ; il est responsable du planning, du budget et de l'efficacité du projet. **Une documentation complète** La production d'une documentation importante et détaillée en amont du projet (recueil de besoins, cahier des charges...) et en aval (documentation technique) constitue une base de l'approche traditionnelle.	**Implication forte du client / utilisateur** Le client/utilisateur acquiert un fort sentiment d'appartenance en travaillant étroitement et directement avec l'équipe de projet tout au long du projet. Grâce au modèle itératif, il peut faire des retours sur les évolutions de manière extrêmement régulière permettant d'assurer des livraisons de hautes qualités et d'accroître la satisfaction des clients / utilisateurs. **Time to market / Flexibilité accrue** L'approche agile est par nature flexible, composé de sprints de durée réduite ; elle permet de livrer plus rapidement des produits adaptés à l'évolution des besoins des clients. **Transparence** L'ensemble de l'équipe est au courant de l'avancement du projet. Dans cette approche, toute l'équipe décide du plan ensemble et partage la propriété du projet ce qui évite la rétention d'informations.
Limites	**Une définition rigide du projet** L'approche traditionnelle suit un ensemble d'étapes séquencées. Si des changements interviennent pendant le projet, l'adaptation pourra être difficile et coûteuse en temps. **Des phases de test tardives** Les tests techniques et fonctionnels interviennent à la fin de la phase de développement. Des résultats insatisfaisants peuvent déclencher des révisions importantes et onéreuses. **Définitions complexes des besoins** Les clients des projets peuvent avoir du mal à présenter des besoins exhaustifs et détaillés voire faire face à de nouveaux concepts. Il peut en résulter des propositions finales ne correspondant pas totalement aux réels besoins.	**Nécessité une appropriation rigoureuse des concepts par l'équipe Agile.** Il peut en découler des incohérences dans les processus et les pratiques Agile mise en œuvre. **Nécessite une forte contribution et implication des utilisateurs / clients** Le projet Agile peut faire face à la résistance au changement notamment des clients et utilisateurs. L'implication du client, très importante pour le projet, peut s'avérer insuffisante présentant des risques de dérives pour le projet **Difficulté à découper et structurer les besoins pour respecter la logique incrémentale** Le découpage des besoins clients / utilisateurs en vue de leur acceptation, priorisation et inscription dans un sprint est une étape importante mais complexe et pouvant être mal maitrisée.

Les risques spécifiques aux démarches agiles

Les particularités et les principaux bénéfices des méthodologies agiles ont été décrits dans les parties précédentes. Elles permettent aux organisations d'améliorer à plusieurs niveaux la gestion et la livraison des projets.

Les spécificités des méthodologies agiles n'apportent cependant pas que des bénéfices. Les projets gérés selon une méthodologie agile sont soumis à des risques spécifiques qui viennent s'ajouter aux risques standards liés à l'exécution de tout projet. Les principaux inconvénients et challenges liés à la démarche agile sont les suivants :

1) L'incompréhension ou mauvaise compréhension de l'agilité
2) La minimisation de la documentation dans un développement agile
3) La complexité de l'adoption de la culture agile
4) Le manque de prévisibilité de l'approche agile
5) La difficulté d'implémentation pour des projets complexes et des entreprises à grande échelle

Les méthodologies agiles se basent sur 4 principes de base édictés dans le manifeste agile. Si ces principes permettent effectivement d'apporter un nombre important de bénéfices lorsqu'ils sont utilisés à bon escient, ils entrainent également des risques spécifiques sur lesquels une équipe d'audit interne doit pouvoir apporter une assurance. Les facteurs de risques listés dans la suite du document sont ceux spécifiques aux méthodologies agiles, les facteurs de risques standards de tout projet restent un point d'attention nécessaire mais ne seront pas développés dans ce document.

Le tableau ci-dessous synthétise les principaux facteurs de risques liés aux principes agiles qui seront développés dans ce chapitre.

Principes agile	#	Principaux Facteurs de risque
Facteur de risque non directement lié à un principe agile	1	Appeler agile une méthodologie qui ne l'est pas
Nous valorisons les individus et leurs interactions plus que les processus et les outils	2	Inadéquation des compétences
	3	Gestion inadéquate des ressources
	4	Outils non adaptés
Nous valorisons des logiciels opérationnels plus qu'une documentation exhaustive	5	Absence de documentation de référence
Nous valorisons la collaboration avec les clients plus que la négociation contractuelle	6	Manque de représentativité / légitimité du Product Owner
	7	Organisation non adéquate
Nous valorisons l'adaptation au changement plus que le suivi d'un plan	8	Incompatibilité avec le type de projet
	9	Absence de périmètre et objectifs clairs du projet
	10	Dérive de budget
	11	Processus connexes au développement de projet non adapté à l'agilité

1. Facteur de risque non lié à un principe agile

Risque 1 – Appeler agile une méthodologie qui ne l'est pas

Le premier risque à mettre en exergue sur les méthodologies agiles est justement que « l'agilité » appliquée dans de nombreuses organisations s'éloigne fortement des standards. Ce risque n'est donc pas directement lié aux principes agiles, mais au fait qu'ils sont parfois mal ou peu appliqués.

En effet, si les méthodologies agiles permettent une grande flexibilité, elles ne sont cependant pas sans structuration. Le cadre de développement SCRUM, qui est le plus répandu, a des critères très précis pour définir ce qu'est un projet agile/SCRUM. Il est donc essentiel lors d'un audit de projet agile de s'assurer qu'une gouvernance projet est bien en place et que les objectifs et contrôles clés de la méthodologie sont bien appliqués. En effet, la démarche agile ne peut pas être un prétexte pour échapper aux contrôles.

Les éléments clés du cadre de développement SCRUM qu'il conviendra de vérifier sur un projet ayant adopté une démarche agile sont les suivants :

- Les rôles clés de la méthodologie agile sont alloués et les responsabilités associées sont clairement définies, particulièrement pour le product owner et le scrum master,
- Le product backlog liste les exigences du projet et est mis à jour de manière dynamique au cours du projet selon un processus défini. Chaque élément du backlog a des conditions claires pour définir quand il est livré,
- Chaque sprint fait l'objet d'un Sprint Planning qui définit les tâches à accomplir durant le sprint (ce sprint planning est alimenté par le product backlog),
- Une réunion d'équipe quotidienne – le daily scrum, permet à l'équipe projet de faire le point sur le progrès du sprint,
- Des réunions de sprint review et sprint retrospective sont organisées pour présenter les résultats et obtenir des feedbacks sur le sprint pour la première, et pour revoir ce qui peut être amélioré dans les processus pour la seconde.

Ces éléments et plus largement la méthodologie agile appliquée doivent faire l'objet d'une documentation dans une méthodologie projet.

2. Principe 1 : Nous valorisons les individus et leurs interactions plus que les processus et les outils

Le premier principe du « manifeste Agile » précise que les individus et leurs interactions ont plus de valeur que les processus et les outils. Ce principe entraine un risque que les projets soient délivrés sans suivre des processus clairs et structurés et place la gestion des équipes projet au cœur des préoccupations. Ce risque est d'autant plus important que de nombreuses organisations ont adapté les méthodologies agiles à leur contexte, en prenant des libertés par rapport aux objectifs initiaux de la démarche agile afin de garder un maximum de flexibilité.

Risque 2 – Inadéquation des compétences

La grande dépendance des projets agiles sur les individus va également entrainer une nécessité que l'ensemble des membres de l'équipe projet aient les compétences requises sur les méthodologies agiles. L'accélération des projets digitaux entraine dans les entreprises une accélération de l'utilisation de méthodologie, qui ne peut se faire de manière efficace sans une gestion du changement adéquat.

Un facteur aggravant des risques liés au principe 1 est donc le manque de formation des équipes en charge des projets agiles.

Il semble indispensable que les équipes projets puissent suivre un cursus de formation agile avant de débuter leur implication dans tout projet, particulièrement pour les positions clés comme le Scrum Master et plus particulièrement le Product Owner qui ne provient généralement pas d'équipes spécialisés dans les projets ou même de l'informatique. En complément, des programmes de coaching ou d'accompagnement par des mentors plus expérimentés sont de bonnes pratiques qui peuvent contribuer à mitiger le risque.

Risque 3 – Gestion inadéquate des ressources

Le fait de valoriser les individus et les interactions plutôt que les processus et les outils permet plus de réactivité face aux changements et une meilleure prise en compte par tous les acteurs des objectifs et des contraintes associées au projet, mais il présente également des risques.
Le principe est que l'équipe agile s'auto-organise avec l'aide de son « scrum master », et adopte des modes de fonctionnement sur-mesure, qui s'améliorent au fil des « rétrospectives » pour garantir la productivité des développements.
Mais un turn-over trop élevé pourrait mettre en péril ce principe : s'approprier des pratiques peu ou pas formalisées devient plus complexe, et la logique d'amélioration continue peut même s'avérer contre-productive si elle ne repose plus sur une expérience partagée par les membres de l'équipe au fil des « sprints ». Il est donc important de s'assurer des dispositifs mis en place pour gérer ce risque (par exemple, adaptation du niveau de spécificité des modes de fonctionnement, du degré de formalisation, etc. en fonction du niveau de stabilité de l'équipe).
La taille de l'équipe joue également un rôle important : l'expérience montre que ce type de fonctionnement n'est pas recommandé pour des structures de plus de 9 personnes. Au-delà, un découpage en sous-produits et des mécanismes de coordination entre les différentes équipes agiles s'avère nécessaire.
D'autre part, le fonctionnement agile, en induisant une plus grande flexibilité et des interactions plus fréquentes entre membres de l'équipe, peut générer quelques ajustements pour garantir la maîtrise de la relation avec les sous-traitants : il s'agira notamment de s'assurer que les livrables et les modalités de suivi définis dans le cadre de la prestation sont compatibles avec le fonctionnement par itérations, et que les interactions avec les intervenants externes se font bien dans le respect du droit du travail (risque de prêt illicite de main d'œuvre).

Risque 4 – Outils non adaptés

Les méthodologies agiles ne nécessitent pas un nombre important d'outils, ce qui est parfaitement aligné avec le premier principe. Les organisations les plus importantes pourront utiliser des outils dédiés pour faire le suivi des éléments clés des projets, mais ces outils peuvent être remplacés par des suivis sous Microsoft Excel dans les organisations plus modestes. Il existe également une offre large de solutions cloud permettant de gérer tout ou une partie du projet.

Il reste néanmoins important qu'un minimum d'outillage soit utilisé pour s'assurer du bon fonctionnement du projet. Ces outils doivent faire l'objet d'une définition claire dans la méthodologie agile et être utilisés par les équipes sur chacun des projets. Il existe deux principaux types d'outils pouvant supporter un projet agile : les outils de gestion des tâches et les outils de collaboration, par exemple :

- Un outil pour gérer le product backlog doit permettre de faire un suivi de l'ensemble des activités à livrer lors du projet et de leur statut / affectation aux différents Sprint,
- Un tableau Kanban ou SCRUM afin de supporter les cérémonies scrum quotidiennes. Lorsque les équipes sont sur site, un tableau blanc peut fonctionner, mais la digitalisation peut être nécessaire.

En complément, dans le contexte actuel, un outil de gestion de la communication à distance est, comme pour tout projet, indispensable.

3. Principe 2 : Nous valorisons des logiciels opérationnels plus qu'une documentation exhaustive

Si la focalisation sur la livraison de logiciels opérationnels est bien évidemment louable, la relégation de la documentation au second plan des priorités préconisée par ce principe renforce les risques liés à l'absence de documentation pour tout projet.

Risque 5 – Absence de documentation de référence

Afin d'assurer la pérennité d'une solution informatique, il est nécessaire de disposer d'un certain nombre de documents de référence. Dans le cadre d'un projet de type Waterfall, il existe en général différents points de contrôle dans le projet pour s'assurer que l'ensemble des documents requis ont bien été livrés par l'organisation projet.
Le second principe du manifeste agile est de privilégier un produit fonctionnel plutôt que la documentation. Le manque de documentation de référence est donc un risque inhérent pour les projets agiles. Pour ne pas tomber dans cet écueil, il est indispensable d'intégrer les documents requis pour les projets agiles dans le backlog, au même titre que les exigences non fonctionnelles. Ces exigences doivent être portées par le Product Owner qui doit s'assurer de les intégrer dans la priorisation du Backlog et dans les objectifs des différents sprints du projet.

Par nature, la documentation supportant les projets agiles est très dynamique pour permettre l'adaptabilité et l'évolutivité désirée. Ce dynamisme ne doit cependant pas empêcher la formalisation d'un socle de base documentaire pour le projet. La méthodologie de projet agile doit préciser quels sont les documents projets requis lors de chaque projet.

4. Principe 3 : Nous valorisons la collaboration avec les clients plus que la négociation contractuelle

Ce principe fait écho à l'une des principales différences entre agile et Waterfall : plutôt que de partir sur une expression des besoins initiaux « ferme et définitive », il y a une collaboration continue tout au long du projet entre les équipes de développement et leurs clients, représentés par le product owner. Cette collaboration qui est une des clés de la réussite des projets agiles, va entrainer un certain nombre de risques spécifiques.

Risque 6 – Manque de représentativité / légitimité du Product Owner

Dans l'organisation agile, le product owner est la voix de l'ensemble des utilisateurs futurs de la solution, c'est-à-dire des clients du projet. C'est un membre clé de l'organisation projet qui possède énormément de pouvoir sur la conception de la solution en développement. Par conséquent, un mauvais casting pourrait avoir des impacts très négatifs sur le projet, voir une remise en cause par les utilisateurs de ce qui aura été fourni. Il est donc indispensable de s'assurer que le Product Owner dispose d'une légitimité et d'une capacité à représenter l'ensemble des utilisateurs finaux. Dans une organisation plus conséquente, ou un projet transverse à plusieurs équipes / départements / business units, il sera indispensable de s'assurer que des processus sont en place pour que le Product Owner collecte l'ensemble des besoins et communique sur les avancées du projet. Des démonstrations fréquentes en cours de projet aux utilisateurs finaux est une bonne pratique à privilégier.

Risque 7 – Organisation non adéquate

La méthodologie agile change de manière significative la manière dont les utilisateurs sont impliqués au cours du projet, lorsqu'on compare leur implication avec des méthodologies plus traditionnelles. Le monde de la gestion de projet Waterfall nécessitait d'avoir une implication des utilisateurs métiers en début de projet, pour définir les besoins, et en fin de projet pour les tests de validation de la solution délivrée.
Lors d'un projet agile, les utilisateurs métiers, et leur représentant dans l'organisation projet (le product owner) vont être amenés à participer à l'ensemble des étapes du projet agile. Il est donc nécessaire que l'organisation des plannings de chaque partie prenante soit adaptée. Encore une fois, ce risque est plus présent dans les organisations moins matures sur la gestion des projets agiles.

Ce risque est particulièrement significatif pour le Product Owner qui joue un rôle clé sur l'ensemble de la durée du projet. Il est le relais entre les utilisateurs et l'équipe projet, et aura la charge de la validation de l'ensemble des besoins et des livraisons. Il est donc nécessaire de s'assurer d'une disponibilité adéquate sur l'ensemble du projet.

5. Principe 4 : Nous valorisons l'adaptation au changement plus que le suivi d'un plan

Ce dernier principe constitue l'essence de tout projet agile : l'adaptabilité aux changements, la flexibilité sur les objectifs ou en tout cas sur le moyen d'atteindre les objectifs du projet. C'est l'une des différences majeures avec les méthodologies traditionnelles de projet, et par conséquent une source de risques spécifiques.

Risque 8 – Incompatibilité avec le type de projet

La méthodologie agile est particulièrement efficace dans les contextes permettant des constructions itératives et des ajustements au fil du projet du contenu du produit. Les projets pour lesquels le but à atteindre est clair, et pour lesquels il n'y a que peu de tolérance au changement, ne seront pas de bons candidats pour la méthodologie agile. Par exemple, les implémentations d'ERP rentrent typiquement dans cette situation : l'objectif est clairement défini, les processus à couvrir sont connus et maitrisés et il est peu probable que le projet subisse des changements majeurs en cours de route. Les avantages de la méthodologie agile, et ce quatrième principe, ne s'appliquent donc pas dans ce contexte. A l'inverse, les projets d'innovation seront très adaptés à agile, notamment les projets de transformation digitale.

Pour mitiger ce risque, il peut être utile de documenter dans la méthodologie de gestion de projet d'entreprise les critères afin de choisir la méthodologie la plus adaptée à chaque projet. Ces critères pourront être regroupés dans une checklist permettant de s'assurer de l'adéquation de la méthodologie utilisée, ou bien faire l'objet d'une documentation dans la charte de projet / note de cadrage du projet.

Risque 9 – Absence de périmètre et objectifs clairs du projet

Le principe est de maximiser la valeur apportée par le projet, grâce à des livraisons plus fréquentes et une plus grande flexibilité pour s'adapter aux évolutions de contexte, en particulier dans des environnements très concurrentiels. Néanmoins, le contenu du produit n'étant pas figé dès les phases amont comme dans le cycle de vie « Waterfall », comment s'assurer que le projet va réussir à maintenir le cap et atteindre ses objectifs ?
Quelle que soit la forme que peut prendre la valeur attendue du projet (gains, « time to market », satisfaction client, conformité réglementaire, etc.), des indicateurs précis et en relation directe avec les livrables du projet doivent être définis pour l'évaluer. Les différentes fonctionnalités envisagées doivent être priorisées en fonction de la valeur qu'elles apportent, et l'analyse de la valeur acquise lors de chaque livraison doit être prise en compte pour déterminer le contenu des itérations à venir. C'est également cette analyse qui permettra de déterminer que le projet a atteint ses objectifs et peut donc se terminer.
On l'aura compris, les contrôles associés à la définition et au suivi de la réalisation des objectifs du projet, nécessaires quel que soit le type de projet, revêtent une importance particulière lorsque l'on souhaite s'assurer que l'on tire tous les bénéfices attendus de l'utilisation de méthodes agiles.

Risque 10 – Dérive de budget

Si tous les projets, quelle que soit la méthodologie adoptée, sont susceptibles de faire face au risque de dépassement de budget, les spécificités des projets agiles peuvent générer des risques particuliers. En effet, si les organisations utilisent de plus en plus les méthodologies agiles, les processus d'allocation budgétaire restent dans la plupart des cas inchangés : traditionnellement, les coûts sont définis, validés et fixés en début de projet sur la base d'estimations prises sur une vue globale du projet (évaluation qui peut s'avérer in fine plus ou moins juste). Lorsque l'on fonctionne par itérations successives pour adapter le contenu du produit aux changements de contexte, la visibilité sur les coûts en début de projet est nécessairement limitée.
Ce mode de fonctionnement peut faciliter l'approche « design to cost » en permettant d'adapter le contenu au budget fixé, mais il peut aussi entraîner des dérives (coûts s'avérant finalement plus élevés que prévu, incapacité à savoir s'arrêter). Comme pour la mesure de l'atteinte des objectifs, une synchronisation avec le rythme des livraisons permet de limiter ces risques.
Lorsque l'on fait appel à des fournisseurs pour réaliser tout ou partie des travaux du projet, le fonctionnement en mode agile peut également introduire des risques de surcoût si les dispositions contractuelles ne prennent pas en compte la flexibilité induite par les méthodes agiles (gestion des évolutions par exemple).

Risque 11 – Processus connexes au développement de projet non adapté à l'agilité

Le développement itératif de la méthodologie agile va déboucher sur un nombre de déploiements plus important que les méthodologies classiques. Par conséquent, il est important que les processus de déploiement / mise en production permettent de soutenir ce rythme pour éviter que la mise en production ne soit un goulot d'étranglement qui limiterait très fortement les bénéfices apportés par la multiplicité des itérations. Les pratiques DevOps visent notamment à unifier les pratiques entre le développement et les opérations pour fluidifier les processus d'implémentation.

Dans les cas où les processus de mise en production restent manuels, Pour chaque projet, il sera nécessaire que le programme de déploiement ait été communiqué et validé avec les équipes de production pour s'assurer de leur capacité à mettre en œuvre les changements.
Dans le cas de processus d'intégration/de déploiement continus (automatisation du déploiement), il conviendra de s'assurer de l'existence et de la pertinence de critères permettant de bloquer le déploiement, en lien avec des outils d'audit de code ou de test par exemple.

Synthèse / conclusion

Pour conclure, si certains facteurs de risques sont plus ou moins importants sur des projets en mode agile que dans des projets traditionnels, l'univers et la typologie des risques à couvrir restent sensiblement les mêmes. L'importance d'auditer ces projets qui peuvent être majeurs, ne change pas non plus. L'auditeur interne doit s'approprier ces nouvelles méthodologies agiles, comprendre la déclinaison qui en est faite au sein de son organisation et adapter son plan d'audit et les programmes de travail afin de couvrir les risques afférents.
L'adaptation des programmes de travail devra permettre que les questions de l'auditeur soient comprises de l'équipe projet. Plus généralement, un effort important d'appropriation du jargon agile est primordial pour la bonne collaboration entre les auditeurs et les audités, assurant l'obtention de la documentation et des informations pertinentes pour la bonne réussite de la mission.
Enfin, nous espérons avant tout que ce document permettra aux équipes d'Audit Interne de mieux appréhender, préparer et conduire des missions d'audit de projets agiles.

Annexes

1. Annexe 1 : 12 principes du « Manifeste Agile »

Les 12 principes suivants sont basés sur le « Manifeste Agile »[3] :

1. Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.
2. Accueillez positivement les changements de besoins, même tard dans le projet. Les processus agiles exploitent le changement pour donner un avantage compétitif au client.
3. Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.
4. Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.
5. Réalisez les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés.
6. 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.
7. Un logiciel opérationnel est la principale mesure d'avancement.
8. 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.
9. Une attention continue à l'excellence technique et à une bonne conception renforce l'Agilité.
10. La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.
11. Les meilleures architectures, spécifications et conceptions émergent d'équipes autoorganisées.
12. À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.

2. Annexe 2 : Agile à l'échelle

La forme la plus simple de développement de produits agiles se situe au niveau de l'équipe - qui est la base. Lorsqu'il est appliqué à un public plus large au-delà du développement informatique de l'organisation, l'interprétation du concept d'agilité devient de plus en plus complexe et envahissante. Afin de créer une organisation agile holistique, les pratiques, les valeurs, les principes et les mentalités de Lean-Agile doivent être appliqués au-delà des équipes - aux programmes, aux portefeuilles et au-delà de l'informatique également.

3. Annexe 3 : Glossaire

Approches agiles de développement logiciel

DEVOPS : DevOps est une méthodologie de développement logiciel qui combine le développement logiciel (Dev) avec les opérations de technologie de l'information (Ops) participant ensemble à l'ensemble du cycle de vie du service, de la conception au processus de développement jusqu'au support de production.

DYNAMIC SOFTWARE DEVELOPMENT METHOD (DSDM) : MÉTHODE DE DÉVELOPPEMENT DYNAMIQUE DU LOGICIEL (DSDM) : est une méthode agile qui se concentre sur le cycle de vie complet du projet. Il a été créé en 1994, après que les chefs de projet utilisant le RAD (Rapid Application Development) aient cherché plus de gouvernance et de discipline à cette nouvelle façon itérative de travailler. Son succès est dû à la philosophie "que tout projet doit être aligné sur des objectifs stratégiques clairement définis et se concentrer sur la livraison rapide d'avantages réels à l'entreprise". À l'appui de cette philosophie, huit principes permettent aux équipes de rester concentrées et d'atteindre les objectifs du projet.

[3] **Source**: Agile Manifesto, 2001 - https://agilemanifesto.org/iso/fr/manifesto.html

EXTREME PROGRAMMING (XP) : PROGRAMMATION EXTRÊME (XP) : XP s'articule autour de petits cycles de développement pour augmenter la concentration et la productivité des développeurs.

INCREMENTAL DEVELOPMENT : DEVELOPPEMENT INCREMENTAL : Presque toutes les équipes agiles privilégient une stratégie de développement incrémental ; dans un contexte agile, cela signifie que chaque version successive du produit est utilisable et que chacune s'appuie sur la version précédente en ajoutant des fonctionnalités visibles par l'utilisateur.

ITERATIVE DEVELOPMENT : DEVELOPPEMENT ITERATIF : Les projets agiles sont itératifs dans la mesure où ils permettent intentionnellement de "répéter" les activités de développement logiciel, et de potentiellement "revisiter" les mêmes produits de travail.

KANBAN : Contrairement à d'autres frameworks agiles, le développement Kanban ne nécessite pas d'itérations fixes. Le travail progresse tout au long du processus de développement sous la forme d'un flux continu d'activités. Kanban s'appuie sur des systèmes de tickets ou cartes qui permettent de visualiser les tâches en attente, en cours ou déjà réalisées. Son objectif est de limiter le gaspillage de ressources et d'efforts en créant des tâches sur la base de la demande.

SCRUM : Scrum est un cadre de gestion léger avec une large applicabilité pour la gestion et le contrôle de projets itératifs et incrémentiels de tous types. La méthode Scrum s'articule autour d'itérations appelées « sprints », durant une période définie, et avec pour objectif de maximiser le temps de développement d'un produit.

Membres de l'équipe de développement agile

BUSINESS ANALYST : ANALYSTE FONCTIONEL L'analyste est à la fois actif dans le soutien des rôles au niveau du projet et entièrement intégré à l'équipe de développement de solutions. L'analyste commercial facilite la relation entre les rôles commerciaux et techniques, en veillant à ce que des décisions précises et appropriées soient prises au jour le jour sur la solution évolutive. L'analyste d'affaires s'assure que les besoins de l'entreprise sont correctement modélisés et analysés et sont correctement reflétés dans les conseils dont l'équipe a besoin pour générer la solution.

BUSINESS STAKEHOLDERS : PARTIES PRENANTES DE L'ENTREPRISE responsables de la réalisation de tests utilisateurs continus et de la fourniture rapide de commentaires à l'équipe de projet.

DOMAIN EXPERT/TECHNICAL ADVISOR : EXPERT DU DOMAINE/CONSEIL TECHNIQUE : soutient l'équipe en apportant une contribution technique spécifique et souvent spécialisée au projet, souvent du point de vue des responsables de la conduite du changement opérationnel, du support opérationnel, de la maintenance continue de la solution, etc.

PRODUCT OWNER : PROPRIÉTAIRE DU PRODUIT : Responsable de combler le fossé entre le client, les parties prenantes de l'entreprise et l'équipe de développement en gérant quotidiennement la clarification des besoins des utilisateurs et la fonctionnalité de l'application.

RELEASE MANAGER : coordonne et planifie le processus d'assurance qualité et de production, gère le cycle de publication, les outils d'automatisation et surveille les activités de publication.

SCRUM MASTER : Dirige l'équipe de développement, élimine les obstacles organisationnels et maintient la cohérence du processus agile. Agit à la fois comme facilitateur et chef de projet.

SOLUTION INFRASTRUCTURE MANAGER : GESTIONNAIRE DE L'INFRASTRUCTURE DE LA SOLUTION : coordonne la mise en œuvre des environnements de développement, de test, d'assurance qualité et de production.

THE COMPONENT TEAM : L'ÉQUIPE DE COMPOSANTS : Les équipes qui travaillent sur un composant ou une couche particulière, et leurs responsabilités restent dans les limites du composant ou de la couche qu'elles sont chargées de développer.

THE FEATURE TEAM : L'ÉQUIPE FEATURE : Équipe composée de 5 à 9 personnes aux compétences transversales qui font le travail et sont autonomes. Cette équipe est responsable de toutes les tâches du sprint d'itération, y compris la planification, la construction, les tests et les commentaires sur les user stories.

Mécanismes de suivi, d'assurance qualité et de test des systèmes

BACKLOG : Le Backlog liste l'ensembles des fonctionnalités, exigences, améliorations et corrections qui constituent des modifications à apporter au produit dans les versions futures.

DAILY MEETING : Chaque jour à la même heure, l'équipe se réunit afin de faire le point sur les informations indispensables à la coordination : chaque membre de l'équipe décrit brièvement les contributions « réalisées » et les obstacles qui se dressent sur leur chemin.

DEFINITION OF READY (DOR) : la liste d'éléments attendus, qu'une user story doit rassembler pour être proposée au développement.

DEFINITION OF DONE (DOD) : la liste de critères à vérifier, afin de déterminer si les user stories ou tickets sont vraiment terminés.

LIGHTWEIGHT MILESTONE REVIEW: EXAMEN DES JALONS LÉGERS : Les jalons marquent des points de progression spécifiques sur le calendrier de développement, et ils peuvent être inestimables pour mesurer et surveiller l'évolution et le risque d'un programme. Il peut y avoir trois types de jalons : incréments de programme (PI), jalons à date fixe et jalons d'apprentissage.

MILESTONE RETROSPECTIVE : EXAMEN DES JALONS : Lorsqu'un projet est en cours depuis un certain temps, ou à la fin du projet, tous les membres permanents de l'équipe (pas seulement les développeurs) investissent de un à trois jours dans une analyse détaillée des événements marquants du projet.

SPRINTS : Période courte délimitée dans le temps, durant laquelle les différents collaborateurs d'un projet vont travailler sur un objectif détaillé et atteignable.

USER STORIES : En concertation avec le client ou le product owner, l'équipe divise le travail à faire en incréments fonctionnels appelés « user stories ». On s'attend à ce que chaque histoire d'utilisateur apporte une contribution à la valeur du produit global.

Agile Project Metrics

CUSTOMER SATISFACTION SURVEY : ENQUÊTE DE SATISFACTION CLIENT : Il existe plusieurs mesures bien connues utilisées pour mesurer la satisfaction client. L'utilisation d'une métrique de satisfaction client cohérente et sa mesure pour chaque version indiquent si l'équipe Scrum atteint son objectif final : apporter de la valeur aux clients.

SCRUM TEAM SATISFACTION SURVEY : ENQUÊTE DE SATISFACTION DE L'ÉQUIPE SCRUM : Sonder périodiquement l'équipe Scrum pour voir dans quelle mesure elle est satisfaite de son travail peut fournir des signaux d'avertissement concernant des problèmes de culture, des conflits d'équipe ou des problèmes de processus.

BUSINESS VALUE BURN UP : BUSINESS VALUE BURN UP : équivalent au « nombre total de user stories + valeur unitaire de chaque user story/durée du sprint »

TEAM VELOCITY : VÉLOCITÉ DE L'ÉQUIPE : équivalent au « nombre d'unités d'effort réalisées/durée du sprint »

TECHNICAL DEBT : DETTE TECHNIQUE : équivalente au « nombre total d'unités (story points) du backlog de produit/durée du sprint »

RETROSPECTIVE PROCESS IMPROVEMENT/SPRINT RETROSPECTIVE : RETROSPECTIVE PROCESS AMELIORATION/SPRINT RETROSPECTIVE : Une réunion de synthèse à la fin du sprint pour partager ce qui s'est bien passé, ce qui s'est moins bien passé, et des idées d'amélioration. Fournit des informations qualitatives qui peuvent aider à évaluer la santé de l'équipe Scrum et du processus.

À propos

Porte-paroles de la profession d'audit interne, l'Institut Français d'Audit et de Contrôle internes (IFACI) et l'Institut des auditeurs internes (IIA) sont des autorités reconnues dans le domaine de la formation et de la formulation des normes, lignes directrices et certifications.

Copyright

Le copyright de ce guide pratique est détenu par l'IFACI.
Pour l'autorisation de reproduction, prière de contacter l'IFACI à l'adresse suivante :
institut@ifaci.com

IFACI

98bis, Boulevard Haussman
75008 PARIS, France
Téléphone : 01 40 08 48 00
Fax : 01 40 08 48 20
Site Web : www.ifaci.com
Courrier électronique : institut@ifaci.com

institut français de l'audit et du contrôle internes

GROUPE PROFESSIONNEL
SYSTÈMES D'INFORMATION

GPSI - AUDIT DE PROJETS AGILE

Institut Français de l'Audit et du Contrôle Internes -
98bis, Boulevard Haussmann 75008 Paris - Tél. : +331 40 08 48 00 - institut@ifaci.com