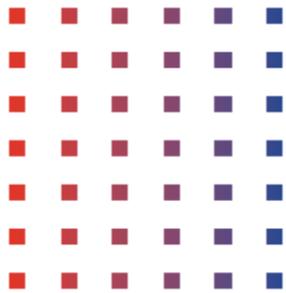




PREMIER MINISTRE



ACHETER DES PRESTATIONS DANS LE CADRE DE PROJETS AGILES SI AU SEIN DE L'ÉTAT

Préconisations de la DINSIC

Novembre 2017

BROUILLON

Ce guide est destiné aux acteurs de l'Etat souhaitant acheter des prestations de service informatique dans le cadre d'un projet ou d'une activité utilisant les méthodes agiles dédiées au système d'information (SI).

Les méthodes agiles se développent fortement dans le numérique et les systèmes d'information. Basées sur des cycles courts, elles facilitent la prise en compte progressive des besoins métier, tout en sécurisant la mise en place de logiciels et applications à des échéances assez courtes. La présente réflexion s'attache aux méthodes agiles SI structurées (SCRUM, Safe, Extreme Programming,...) et non aux concepts agiles (exemple du Lean start-up) qui peuvent également s'appliquer au numérique dans le cadre de start-up. Ce guide n'a pas non plus vocation à déterminer dans quelles conditions il faut utiliser des méthodes agiles dans un projet ou la maintenance SI. A ce sujet, de nombreuses réflexions restent en cours sur leur utilité dans des contextes variés, tels que les grands programmes ou les applications complexes. A l'heure actuelle, leur efficacité est cependant reconnue pour les développements de petites et moyennes applications (API, applications à front office web,...), ou les délais projets restent raisonnables (un à deux ans).

De nombreuses administrations utilisent des méthodologies agiles SI, soit pour des usages très spécifiques, soit dans une optique de généralisation à l'échelle d'une organisation d'une méthode de gestion de projet. Cependant, de nombreuses difficultés restent constatées à l'acquisition des compétences internes pour ces méthodes, qui nécessitent une expertise différente de celle plus classiquement nécessaire aux projets « cycle en V ». L'appel à des prestations de service externes est généralement la solution pour palier à ce manque de compétences ou de moyens.

Les sociétés de service informatique (ESN) ont développé depuis quelques années des prestations de services dédiées aux méthodes agiles, tant sur le conseil à la méthode, que sur le développement. Les centres de services des ESN développent fortement ces pratiques, notamment pour les activités de TMA. Aujourd'hui, le marché propose de nombreuses solutions et expertises sur le sujet. La mise en concurrence en est facilitée. Pour l'Etat, l'enjeu est d'effectuer le bon achat, afin que l'appel à ces sociétés se fasse dans les meilleures conditions. Ce guide fournira quelques conseils utiles pour sécuriser votre achat.

Les difficultés que rencontrent les prescripteurs de l'Etat sur les méthodes agiles sont diverses. D'une part, la découverte de nouveaux concepts, d'un nouveau vocabulaire, et d'une méthode de travail non habituelle. D'autre part, une appréhension dans l'achat de prestations dont le périmètre peut apparaître par essence comme plus difficile à cerner ou à circonstancier. Nous expliquerons ainsi dans ce guide qu'un dialogue serein entre acheteur, prescripteur et juristes peut être rétabli, dans la mesure où l'on évoque bien l'achat pour les méthodes agiles SI. En tous les cas, l'achat de prestations intellectuelles informatiques demande une expertise assez avancée, tant en connaissance du droit des marchés publics, qu'en méthode et technicité sur les systèmes d'information. Nous essayerons de partager dans ce guide quelques bonnes pratiques sur le sujet, en notant que l'apport d'un acheteur informatique expérimenté facilitera fortement votre démarche d'achat.

Prérequis à la bonne compréhension du guide :

- Une connaissance du droit des marchés publics
- Une initiation aux fondamentaux des méthodes agiles SI (un rappel des principales méthodes est précisé en annexe)

LES PRINCIPES A RETENIR	6
LE DROIT DES MARCHES PUBLIC, COMPATIBLE AVEC LES METHODES DE PROJETS AGILES SI?... 8	
1. Le droit des marchés public, une base solide pour la relation client-fournisseur	9
2. La maîtrise des techniques d’achat informatique, la clef du succès.....	11
BONNES PRATIQUES POUR L’ACHAT DE PRESTATIONS DANS LE CADRE DE PROJETS AGILES SI. 12	
1. Les entrants nécessaires au choix d’une ingénierie d’achat	13
2. L’ingénierie d’achat.....	14
3. La sélection des opérateurs économiques	17
3. Les recommandations pour l’élaboration du CCTP.....	18
4. L’importance du suivi de l’exécution du marché	21
ANNEXES	22
a. Bibliographie et documentation complémentaire.....	23
b. Présentation des principales méthodes agiles SI.....	24
c. Glossaire.....	31
d. Exemples de modalités de sélections des opérateurs économiques	38

BROUILLON

1

Les principes à retenir

Principe 1 : Il est possible d'acheter des prestations pour les méthodes agiles SI avec le droit des marchés publics. Le cadre contractuel peut être adapté.

Principe 2 : Les échanges préalables avec les opérateurs économiques (sourcing) et le travail sur les critères de sélection des candidats sont des opérations de l'achat permettant d'optimiser l'apport d'opérateurs économiques compétents.

Principe 3 : Un marché pour les méthodes agiles est adapté à un contexte, un besoin et à un environnement, il est difficilement générique.

Principe 4 : Les exigences des méthodes agiles nécessitent une souplesse dans l'exécution. L'ingénierie d'achat doit pouvoir s'adapter à la majeure partie des situations du projet.

Principe 5 : L'exécution d'un marché pour les méthodes agiles nécessite un pilotage fort de la part de l'entité bénéficiaire et une compétence technique interne avérée.

2

Le droit des marchés public,
compatible avec les méthodes
de projets agiles SI?

Parle-t-on d'agilité des marchés ou d'achat pour méthodes agiles ?

Le débat sur l'agilité des marchés public, rendre les procédures plus accessibles, favoriser l'accès à toutes les entreprises, quel que soit leur taille, accélérer les processus administratifs sont des éléments essentiels pour encourager un meilleur accès aux entreprises compétentes et assurer la performance de l'action publique. Ce débat dépasse cependant la problématique de contractualisation avec des fournisseurs dans le cadre de projets informatiques utilisant les méthodes agiles. Le présent guide ne traite ainsi pas le sujet d'amélioration des processus de la commande publique. Bien entendu, l'amélioration et la modernisation de la commande publique seront des facilitateurs aux pratiques des informaticiens. Nous prenons dans ce document le parti de développer les bonnes pratiques à partir de l'état actuel du droit des marchés publics et des méthodes/pratiques de gestion associée. Les propositions visent à être pragmatiques et opérationnelles.

1. Le droit des marchés public, une base solide pour la relation client-fournisseur

Le droit des marchés publics propose un ensemble de modalités de contractualisation avec les fournisseurs, sécurisé, complet et surtout adapté aux pratiques de l'administration. Contrairement à beaucoup d'idées reçues, il n'impose pas de freins à l'utilisation de pratiques de gestion de projet nouvelles telles que celles sur les méthodes agiles.

Plusieurs éléments fondamentaux du droit sont des vecteurs rassurants pour l'administration et les fournisseurs dans les marchés :

- La mise en concurrence, nécessaire à partir d'un montant de prestation, impose d'étudier et de choisir des offres sur la base d'un besoin exprimé et de critères de sélections réfléchis. La contractualisation avec un opérateur économique permet ainsi de poser les bases d'un accord commun de travail, avec ses limites et son périmètre ;
- L'assurance d'un dialogue de facturation et paiement clair, avec les possibilités offertes par le droit (avances, acomptes, paiement direct des sous-traitants, etc.).

L'article 30 de l'Ordonnance 2015-899 du 23 juillet 2015 relative aux marchés publics précise « La nature et l'étendue des besoins à satisfaire sont déterminées avec précision avant le lancement de la consultation en prenant en compte des objectifs de développement durable dans leurs dimensions économique, sociale et environnementale ».

De nombreux acteurs citent cet article afin de justifier que les méthodes agiles ne sont pas compatibles avec le droit, du fait qu'elles permettent une définition des fonctionnalités progressive. Nous tenons à clarifier la pratique sur le sujet :

- La notion de « besoin » est à interpréter de manière différente en fonction des contextes. Un marché agile ne peut, contrairement à certaines idées reçues, être réalisé à partir d'une feuille blanche. Un cadrage initial du projet est obligatoire et s'impose en amont du lancement du marché. Il définit généralement les éléments suivants : objectifs métier, macro-fonctionnalités (services rendus à un ensemble d'utilisateurs finaux), exigences non fonctionnelles (disponibilité, volumétrie, temps de réponse...), solutions candidates et coûts associés, risques du projet et plan d'action associé. Ces éléments correspondent à des besoins explicites, permettant aux soumissionnaires de présenter l'offre la plus adaptée.

- Le besoin d'un marché « Agile » est correctement exprimé à partir du moment où la nature des prestations à réaliser est clairement définie dans le marché. Un marché qui se base sur les méthodes de réalisation agiles doit prévoir l'ensemble des natures de prestations : conception, rédaction de spécifications, réalisation de tests,... prestations généralement présentes dans des marchés informatiques utilisant des unités d'œuvre.

Il est vivement conseillé une bonne maîtrise du droit des marchés publics avant de se lancer dans l'élaboration d'un marché agile. Le site internet de la DAJ (Direction des Affaires Juridiques, Ministères économiques et financiers) propose de nombreux renseignements sur le droit des marchés publics (<http://www.economie.gouv.fr/daj/textes-regissant-commande-publique>). Cette connaissance est nécessaire tant dans la mise en place du marché que dans son exécution. L'IGPDE propose des formations courtes pour apprendre les principes essentiels du droit (http://www11.bercy.gouv.fr/catalogue-igpde/co/FGP_AC_CommPub_Ini.html).

Il est ainsi nécessaire de veiller à ce que l'achat souhaité prenne en compte les spécificités du droit, notamment sur les aspects de propriété intellectuelle (cf. option A ou B du CCAG/TIC), des procédures et des conditions de commande et livraison des prestations réalisées.

2. La maîtrise des techniques d'achat informatique, la clef du succès

Les techniques d'achat permettent d'adapter la transcription des besoins en procédure de marché public. Le choix de bonnes tactiques d'achat est un vecteur essentiel de réussite des marchés, tant dans l'adéquation des offres fournisseurs avec le besoin, que dans la bonne exécution du marché.

Dans le domaine de l'achat lié aux systèmes d'information, de nombreuses pratiques permettent d'assurer et d'adapter les procédures aux méthodes et nature des besoins exprimés, notamment dans le segment des prestations intellectuelles informatiques.

Il est ainsi préconisé, dans le cadre de la mise en place des marchés pour les méthodes agiles, de maîtriser ces pratiques, telles que :

- La notion d'accord cadre, de marchés avec des procédures simplifiées ou non ;
- Les notions d'unités d'œuvre et leur pratique d'exécution ;
- Les différentes formes de procédures, afin de bien les adapter à la taille du marché ;
- Les pratiques relatives aux engagements spécifiques demandés aux prestataires ;
- Les pratiques de vérification des prestations pour les attestations de service fait ;
- Les dispositifs d'échéancier de versement d'acomptes, afin de bien les adapter aux spécificités du marché.

Pour l'ensemble des pratiques, une formation de perfectionnement à l'achat public est essentielle. Une formation spécialisée sur l'achat de prestations intellectuelles informatiques, telle que proposée par l'IGPDE (<http://www11.bercy.gouv.fr/catalogue-igpde/co/7752.html>) est recommandée.

La prescription technique d'un marché Agile est ainsi essentielle. Celle-ci assurera la bonne traduction des objectifs du projet dans le contexte de l'administration concernée. Le livrable associé sera le CCTP du marché.

Nous conseillons ainsi fortement de vous appuyer, dans le cadre de la mise en place d'un marché Agile, sur des acheteurs ou prescripteurs techniques ayant une bonne connaissance des techniques d'achat informatique sur le segment des prestations intellectuelles.

3

Bonnes pratiques pour l'achat
de prestations dans le cadre
de projets agiles SI

1. Les entrants nécessaires au choix d'une ingénierie d'achat

Une stratégie d'achat repose sur 3 piliers majeurs :



Une connaissance du besoin
(ce que l'on veut faire/acheter – produit final)



Une connaissance de l'existant
(ce que je suis capable de faire/ ou pas)



Une connaissance du marché fournisseur
(qui peut répondre à mon besoin)

En fonction de ces entrants, l'ingénierie d'achat devra s'adapter. Elle s'appuiera sur des éléments concrets et sera adaptée aux contraintes et besoins de l'organisation, mais également aux spécificités du marché fournisseur.

L'acheteur et le prescripteur devront préalablement au lancement du marché, avoir anticipé de nombreux points relatifs au projet de réalisation. Cette phase est essentielle pour élaborer un appel d'offre (consultation) et n'est pas spécialement liée aux méthodes agiles. Elle s'applique quelle que soit la méthode de mise en œuvre du projet.

Par exemple, les bonnes questions à se poser peuvent être :

- A-t-on besoin d'une externalisation totale ou partielle ?
- Est-on dans un cadre projet spécifique ou dans un cadre plus global (TMA, multi-projets,...) ?
- Est-ce qu'une méthode de gestion de projet (cycle V ou agile) existe ou est pré-déterminée ?
- Mon organisation actuelle peut-elle être adaptée à une méthode de gestion agile ?
- Quel est le planning ? Le cadrage a-t-il été effectué (jalons, macro-besoins, budget, moyens, ...) ?
- Etc.

PRECONISATIONS D'INSIC

Cadrer le projet agile avant de lancer un appel d'offre : même si les méthodes agiles sont pensées pour apporter le maximum de flexibilité, il est fondamental d'avoir anticipé la teneur du produit à délivrer.
Définir dans ce cadrage les tâches qui sont externalisées et celles qui ne le sont pas.

2. L'ingénierie d'achat

Une ingénierie d'achat permettra de décrire l'attendu de l'opérateur économique et se traduira par des règles de marché (CCAP) et une expression de besoin technique (CCTP). Il convient que ces deux documents soient étroitement liés et ne peuvent s'élaborer l'un sans l'autre.

L'aide à la définition des besoins et le sourcing

L'ingénierie d'achat est généralement réalisée suite à une première définition des besoins (macro-expression des besoins), que les acheteurs peuvent confronter à une réalité du marché fournisseur. Pour cela, une phase d'échanges préalables avec les opérateurs économiques est recommandée. Celle-ci vise à confronter le besoin par rapport aux pratiques des opérateurs économiques et vérifier qu'une concurrence peut s'établir sur le périmètre déterminé. Cette pratique doit être encadrée et respecter le droit, telle que le précise le décret « [ces pratiques] peuvent être utilisées par l'acheteur, à condition qu'ils n'aient pas pour effet de fausser la concurrence et n'entraînent pas une violation des principes de liberté d'accès à la commande publique, d'égalité de traitement des candidats et de transparence des procédures. »

En ce qui concerne les méthodes agiles, cette phase d'échange peut permettre :

- D'identifier des opérateurs en capacité de répondre au besoin (maîtrise de la méthode souhaitée, capacité de réalisation, expertise suffisante, respect des conditions de sécurité et de confidentialité,...) ;
- De « motiver » des opérateurs économiques au projet et ainsi permettre une concurrence sur votre besoin ;
- D'adapter les besoins aux contraintes ou avantages que peuvent proposer ces opérateurs.

Une fois la phase d'échange avec les opérateurs économiques terminée, l'élaboration de la stratégie d'achat peut être déterminée. Elle s'attachera à déterminer les conditions d'achat les plus adaptées au contexte du projet et de l'organisation.

Le choix de la forme de marché et les principes d'achat

Il conviendra de choisir la bonne forme (procédure) en fonction de l'objectif et des caractéristiques du projet.

- Les marchés à simples (forfait global) peuvent être utilisés dans certains contextes :
 - Si le cadrage projet et la description des attendus sont suffisamment détaillés pour permettre un engagement ferme sur la durée du marché. Il est toutefois conseillé de prévoir une capacité à commander des prestations complémentaires par l'intermédiaire d'unités d'œuvre et de bien déterminer les conditions de sortie du marché. Ceci concerne essentiellement les petits projets, très cadrés ;
 - Dans le cadre de prestations de développements de maintenance.
- Les marchés avec des tranches conditionnelles sont peu conseillés, notamment si celles-ci sont utilisées pour gérer des sprints projets. Ces principes sont moins souples que des unités d'œuvre. La conditionnalité ne facilite pas un bon positionnement des fournisseurs, qui peuvent ainsi provisionner des coûts de gestion du risque. Par contre, ils peuvent être utilisés pour permettre des extensions de marché ou la tierce maintenance applicative (TMA).

- Les marchés (accords-cadres) à bons de commande avec des unités d'œuvre (forfaitisation de nature de prestations) sont une solution assez compatible avec les méthodes agiles. Ces marchés offrent de la souplesse d'exécution, et peuvent s'accommoder avec des rythmes projet variés. Ils nécessitent cependant davantage de précautions dans l'exécution et la préparation du marché ;
- Les marchés mixtes (accords à bons de commande et marchés subséquents) sont une possibilité en méthode agile. Ils peuvent combiner une partie ferme permettant de s'engager sur une durée et sur un cadrage projet, et une partie à bons de commande d'unités d'œuvre permettant une flexibilité et un ajustement des besoins en fonction de l'avancement du projet.

Le mode agile nécessite de prévoir dans le CCAP et le RC (règlement de consultation) l'ensemble des clauses qui permettent de sécuriser la relation avec le titulaire.

Quelques principes nous paraissent importants pour l'écriture du RC :

- Limiter les critères de candidature qui pourraient avoir pour effet d'exclure certaines catégories d'entreprises, notamment les PME, compétentes dans le domaine de l'agile. Par exemple, les critères de chiffre d'affaire minimum, ou les exigences de processus qualité internes peuvent être déterminant ;
- Limiter le nombre de pages du cadre de réponse et ne pas s'offusquer d'un défaut de présentation : certaines PME, compétentes dans le domaine de l'agile, ne doivent pas être pénalisées par rapport à des sociétés ayant une capacité commerciale forte ;
- Travailler sur les critères de sélection (cf. chapitre dédié ci-après)

Quelques principes nous paraissent importants pour l'écriture du CCAP :

- Préciser les règles de commande: sur les accords-cadres à bons de commande, préciser les modalités de passation des commandes et d'établissement des services faits. L'ingénierie des commandes, adaptée à un rythme projet agile (par release par exemple), peut être indiquée ;
- Engager le fournisseur, même si l'achat est adaptatif, sur le résultat attendu. Les périodes et jalons doivent être déterminés et il peut être envisagé d'engager le titulaire sur un forfait TMA après le projet ;
- Intégrer des clauses sur la stabilité des équipes du titulaire et des exigences sur la gestion de changement de profils ;
- Prévoir des pénalités adéquates, applicables et raisonnables ;
- Prévoir un dispositif de versement des acomptes ou de règlements partiels définitifs incitatifs ;
- Les droits de propriété intellectuelle doivent être adaptés au contexte projet (cf. guides de l'APIE sur les spécificités du domaine informatique) : compatibilité des licences, mutualiser, réutiliser, périmètres des droits, etc.

Préparer la publication du marché

Les opérateurs économiques, généralement nommés ESN (Entreprise de Service Numérique) dans le domaine du SI, doivent avoir connaissance de votre besoin et du lancement de votre procédure d'achat. L'intérêt de l'échange préalable avec les opérateurs économiques, ou plus communément appelé le sourcing, sera vital pour cibler les ESN compétentes.

Il peut également être pertinent de sensibiliser les acteurs (TPE, PME et grandes entreprises) sur les possibilités offertes par l'achat public, avec par exemple l'utilisation de la bourse à la cotraitance (permet de fédérer des opérateurs économiques sur un besoin) et les marchés simplifiés.

Enfin, l'utilisation de la demande d'information (DI) ou « request for information » (RFI) sont des pratiques facilitant la visibilité du marché vers les fournisseurs.

PRECONISATIONS DINSIC

S'appuyer sur des pratiques de sourcing (échange préalable avec les opérateurs économiques) afin de mieux cibler les ESN compétentes et les plus adaptées au besoin.

Prévoir un marché qui offre une forte souplesse dans la commande telle que le permettent les accords-cadres à bons de commande avec des unités d'œuvre ou les accords-cadres mixtes (subséquent et bons de commande).

BROUILLON

3. La sélection des opérateurs économiques

La sélection des opérateurs économiques est cruciale pour mener à bien un projet en méthode agile. Il est essentiel de se donner les moyens de vérifier et de valider les compétences des candidats par rapport aux exigences techniques et à la méthode choisie.

Dans une démarche d'achat, la bonne anticipation des modalités de sélection des candidats se fait à différentes étapes :

- Dans le règlement de consultation (RC) du marché, les modalités d'attribution du marché doivent permettre de cibler les critères de compétences et de profils. Le critère TJM (Taux journalier Moyen) n'est pas à retenir dans une démarche de sélection : en effet, un développeur agile est plus cher qu'un développeur dit « classique », et sélectionner sur un critère de performance au profil n'est ainsi pas adapté. Le critère « prix » doit se focaliser sur les engagements par unités d'œuvre (dans le cadre d'un marché à UO). De plus, le critère prix ne doit pas être dominant sous peine de limiter fortement la sélection sur le critère technique. Un ratio de 65% à 70% (même 80%) minimum pour le critère technique est recommandé ;
- Dans le cadre de réponse (imposé de préférence), les informations de l'offre du fournisseur doivent être ciblées sur des attendus précis :
 - o Les expériences similaires peuvent permettre de déterminer si le candidat a déjà pratiqué dans un environnement proche de l'attendu. On s'appliquera à demander certains détails permettant de juger de la cohérence de l'exemple proposé : méthode utilisée, nombre de sprints, taille et organisation de l'équipe, durée projet,...
 - o Les exemples de profils demandés doivent être pré-formatés par des informations liées à l'agilité. Par exemple, des publications sur l'agilité, des formations et surtout des exemples de réalisations sont des moyens de vérifier la bonne compétence d'un profil proposé par le candidat ;
 - o La cohérence technique de la proposition doit être visible rapidement mais il ne faudra pas se focaliser sur la forme de la réponse pour ne pas pénaliser les candidats dont la structure ou la taille ne permettent pas un investissement commercial aussi important que de plus grandes organisations. Limiter le cadre de réponse à un nombre page et bien préciser les attendus permet d'éviter les réponses trop hétérogènes ou peu synthétiques.

Au-delà de la démarche, les profils proposés par le titulaire sont essentiels à la bonne réalisation d'un projet Agile. Non seulement l'équipe proposée devra offrir un ensemble homogène et cohérent de compétences, mais sa composition devra être compatible avec les principes des méthodes agiles, très contraignants sur les échéances et la rigueur méthodologique. Afin de se prémunir de profils non adaptés, des clauses sur les obligations du titulaire vis-à-vis de l'équipe et de garantie de continuité des prestations peuvent être insérées dans le CCAP du marché.

Certains critères types de sélection sont précisés en annexe : « référence type », « profil type », obligations sur les équipes, la continuité des prestations...

PRECONISATIONS DINSIC

Porter une attente particulière sur les critères de sélection des candidats, en ciblant des compétences et expériences dans la conduite de projet en méthodes agiles.

3. Les recommandations pour l'élaboration du CCTP

Au-delà des pratiques d'achats, l'élaboration du CCTP d'un marché agile nécessite une attention particulière. Nous proposons ainsi quelques bonnes pratiques à adopter ou à anticiper pour sa réalisation :

Clarifier la méthodologie

Afin de bien cibler les compétences attendues et les modalités de fonctionnement entre l'entité bénéficiaire et le titulaire, il est nécessaire de préciser la méthodologie concernée par le projet agile. Les éléments de méthodologie suivants devront ainsi être détaillés :

- Citer le nom de la méthodologie qui a été préconisée en tenant compte du contexte du projet (par exemple : scrum pour un projet d'innovation, kanban pour un projet de maintenance, SAFe pour un projet de grande taille, ...)
- Préciser les événements de l'agilité (« cérémonies ») constituant la dynamique du projet (ex : sprint planning, DEMO, ...)
- Préciser les modes d'interaction et les points d'intervention de l'administration, le cas échéant en architecture, définition du backlog, revue de code, contrôle de la dette technique, recette, gestion des bugs, audit sécurité...
- Définir clairement le « Done » pour les stories (point clé d'un contrat Agile), c'est à dire les critères d'acceptation des fonctionnalités livrées par le titulaire.

Définir les éléments de planification du projet

Les éléments de planification de la démarche Agile permettent de partager entre l'entité bénéficiaire et le(s) titulaire(s) les grands jalons du projet. Il est recommandé que les jalons majeurs du projet soient partagés dès la consultation et soient intégrés dans le dossier de consultation. Les opérateurs économiques auront ainsi connaissance des objectifs qui leur seront assignés. De la même manière que pour les fréquences d'itération et la cadence générale, la durée de la phase d'initialisation, le nombre de versions, ainsi que la durée de maintenance sont des jalons à positionner. Il est recommandé également de préciser les adhérences avec d'autres projets et les contraintes organisationnelles à prendre en compte.

Le Product Owner sera par ailleurs très vigilant sur les principes de base suivants : rigueur absolue sur la/les date(s) de livraison et la charge engagée (budget) de chaque release. La philosophie Agile incite en effet à arbitrer sur le périmètre des attendus par effet de priorisation relative.

Clarifier la répartition des responsabilités entre Product Owner (PO) et titulaire

Le Product Owner est le plus souvent un membre de l'entité bénéficiaire et a un rôle majeur dans le pilotage du projet. Il détermine les priorités et la valeur des fonctionnalités ainsi que leur adéquation avec les objectifs « métier » du projet. Nous recommandons ainsi que :

- L'évaluation de l'effort pour réaliser une fonctionnalité soit estimée par l'équipe de fabrication devant le PO et partagée (cf. méthode d'évaluation des charges ci-après) ;
- Toutes les parties prenantes se rencontrent régulièrement pour vérifier l'avancement du projet et ajuster l'approche si nécessaire ;
- Le PO soit disponible pour répondre aux questions fonctionnelles et aux demandes d'arbitrage de priorités ;

- Les livrables à fournir par le PO soient précisés (user stories avec critères d'acceptation, cas de test associés, exigences non fonctionnelles...).

Préciser une méthode d'évaluation des charges ou des points de fonction

La mesure du niveau de complexité d'un sprint ou d'une release, essentielle au chiffrage car correspondant pour partie à la quantité de ce qui va être commandé (le volume de travail), est une des difficultés principales à laquelle sont confrontés les commanditaires. Plusieurs méthodes peuvent être utilisées pour l'évaluer. Une première méthode dite des « points de fonctions » permet d'avoir des références stables dans la mesure de la complexité de la réalisation. Chaque unité d'œuvre commandée est ainsi graduée en fonction de sa complexité mesurée en points de fonction. Cette méthode est bien adaptée pour les besoins fonctionnels mais s'applique plus difficilement aux besoins techniques. D'autres méthodes basées sur des évaluations de charges (en j.h) existent et peuvent être adaptées à chaque contexte organisationnel et technique.

LE PLANNING POKER

Le planning Poker : une méthode de chiffrage permettant de faciliter le consensus par le biais de l'intelligence collective

Définir une ingénierie des unités d'œuvre

L'ingénierie des unités d'œuvre aura un rôle majeur pour la bonne exécution du marché. Il est recommandé de porter une attention forte à l'élaboration de celle-ci de manière à ce qu'il soit possible d'anticiper la majeure partie des cas d'intervention du titulaire.

Une unité d'œuvre doit préciser a minima :

- La description des prestations attendues ;
- Le(s) profil (s) et compétence(s) attendu(s) ;
- Le délai de réalisation/complexité estimée (en jours, en point de fonction, ...) ;
- Les livrables avec délais de livraison et vérification attendus.

Nous recommandons de s'appuyer sur des référentiels reconnus et maîtrisés par les ESN pour les niveaux de profil et les compétences attendues:

- Référentiel des compétences IT Européen : http://www.ecompetences.eu/wp-content/uploads/2014/02/European-e-Competence-Framework-3.0_FR.pdf
- Classification des ingénieurs et cadre SYNTEC : http://www.syntec.fr/fichiers/Annexes/20130719184036_Convention_Syntec_Annexe_02.pdf

Décrire les environnements et l'usine logicielle

Les environnements peuvent être multiples et variés suivant la taille et la typologie du projet. Dans le cadre d'une démarche Agile, une gestion optimale des environnements favorise la réactivité et la souplesse. Les principes d'externalisation de la gestion de l'hébergement des environnements doivent être anticipés en amont de toute contractualisation afin que les rôles et responsabilités soient bien partagés entre l'entité bénéficiaire et le(s) titulaire(s).

L'usine logicielle permettra l'intégration continue du code, l'automatisation de l'exécution des tests, la livraison automatique, le contrôle automatique de code. Si l'administration ne propose pas d'usine logicielle adaptée au projet, il conviendra de préciser dans le marché que le titulaire devra proposer l'usine associée au projet. Ce point étant très structurant, il faudra clarifier les responsabilités de chacun à ce sujet. Dans le cas où le titulaire en serait responsable, cela peut être un critère de sélection de ce dernier. La détention d'une bonne usine logicielle est par ailleurs souvent un gage de maîtrise et de pratique de l'agilité au d'un prestataire.

Exiger des métriques et un suivi de la qualité du travail effectué

La qualité peut s'aborder suivant deux perspectives :

- La qualité externe, validée via les tests ;
- La qualité interne et la maintenabilité, vérifiées via des revues de code et des contrôles automatiques du code (avec des outils comme SONAR par exemple). Ces éléments de contrôle doivent faire partie de la définition du « Done ». Le non-respect de la qualité interne du produit, que ce soit au niveau du code ou de l'architecture est source de dette technique et donc d'augmentation du temps d'intervention des développeurs.

Le CCTP doit prévoir un chapitre sur les métriques. Les candidats doivent préciser leurs définitions et les moyens mis en œuvre pour garantir le respect des métriques suivantes, à mesurer régulièrement :

- Productivité ;
- Qualité technique et fonctionnelle ;
- Satisfaction du client.

PRECONISATIONS DINSIC

Réaliser un cadrage projet qui détermine les entrants indispensables à l'écriture du marché, comportant :

- La méthode choisie (SCRUM par exemple) ;
- Les objectifs et l'attendu du produit final ;
- Le backlog initial (macro-fonctionnalités) ;
- Le PO (Product Owner) de l'entité bénéficiaire désigné (ne pas avoir de PO interne n'est pas envisagé) ;
- Les versions du produit et leurs échéances attendues, avec a minima une V0, par exemple un MVP (Minimum Viable Product) ;
- La localisation du plateau de développement (Etat ou chez titulaire) ;
- Les phases de tests fonctionnels alignées avec les échéances des sprints ;
- Le cadre technique préalable : usine logicielle interne ou client, où sont hébergés les développements/tests, open-source/propriétaire, sécurité du produit, ... ;
- La boucle de feedback (idéalement de production) avec un décideur fonctionnel dans l'équipe projet.

4. L'importance du suivi de l'exécution du marché

Un marché Agile basé sur les principes expliqués précédemment nécessite une attention particulière dans son exécution. En effet, bien que les méthodes agiles prônent un niveau minimum de formalisation écrit avant le développement, les fondamentaux du pilotage (rigueur et anticipation notamment) restent essentiels.

Afin de bien anticiper les exigences du pilotage d'un marché pour les méthodes agiles, il est notamment recommandé les points suivants :

- L'ingénierie des commandes, dans le cadre d'un marché avec des unités d'œuvre, doit être organisée et partagée avec le titulaire. La préparation des commandes est effectivement un élément important pour « calibrer » au mieux les commandes aux besoins ;
- Lors de chaque sprint et/ou en amont de chaque release, une « discussion » sera engagée avec le titulaire sur le contenu du sprint. Une méthode d'évaluation de la charge ou points de fonction devra être déterminée afin d'éviter les potentiels différents entre le titulaire et l'administration ;
- L'administration devra être en capacité de « challenger » le fournisseur sur les résultats et ses évaluations. Sa capacité à comprendre les évaluations de charge et les propositions techniques du fournisseur est incontournable ;
- Les cycles courts nécessiteront une forte réactivité également en interne, les travaux sous la responsabilité de l'administration devant suivre le rythme agile (par exemple tests, analyse fonctionnelle....). Si un écart de rythme entre l'Administration et le titulaire s'installe, le projet peut rapidement être déstabilisé ;
- La dérive des coûts devra être contrôlée par un suivi rigoureux et une bonne anticipation des prévisions de commande ;
- Les services administratifs doivent être en mesure de comprendre les particularités de pilotage du marché et s'adapter aux pratiques liées (commandes régulières, services faits partiels, ...) ;
- Il sera nécessaire d'être vigilant sur l'atteinte des objectifs et de définir les modalités de service faits adéquats (respect de la définition du « Done » du CCTP).

PRECONISATIONS DINSIC

Impliquer le chef de projet qui sera un acteur clé du pilotage fournisseur. Il devra s'impliquer dans l'ingénierie et le contenu des commandes fournisseurs. Sa compétence et sa capacité à piloter le projet agile devront être un élément déterminant de la validation du projet, avant son lancement.

A

Annexes

a. Bibliographie et documentation complémentaire

- Textes sur la commande publique en vigueur : <http://www.economie.gouv.fr/daj/textes-marches-publics>
- Formations sur les marchés publics : <http://www11.bercy.gouv.fr/catalogue-igpde/co/marches-publics-AC.html>
- Organisation des achats de l'Etat : <https://www.economie.gouv.fr/dae>
- Documents de références du SI de l'Etat : <https://references.modernisation.gouv.fr/>
- Clausier et exemples par la Direction des Achats de l'Etat (disponible sur demande auprès de la DAE)

b. Présentation des principales méthodes agiles SI

Scrum

Présentée pour la première fois en 1995 à la conférence OOPSLA par Ken Schwaber et Jeff Sutherland
Site de référence scrum.org

Définition de Scrum

Cadre de développement dans lequel des équipes plurifonctionnelles réalisent des produits de manière itérative et incrémentale. Scrum structure le développement en cycles de travail appelés Sprints. Ces itérations durent de deux à quatre semaines, et s'enchaînent l'une après l'autre sans interruption. Les Sprints sont d'une durée limitée.



Généralement les Equipes Scrum choisissent une durée de Sprint et la maintiennent durant le projet, jusqu'à ce qu'elles puissent encore augmenter leur productivité et utiliser alors un cycle plus court.

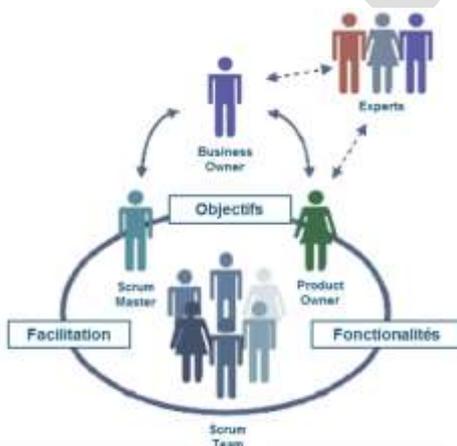
Au début de chaque Sprint, une Equipe plurifonctionnelle (d'environ sept personnes) sélectionne des éléments dans une liste priorisée appelée « backlog ». L'Equipe s'accorde collectivement sur une cible constituée de ce qu'elle pense pouvoir livrer à la fin du Sprint, de manière tangible et réellement « terminée ». Aucun nouvel élément n'est ajouté durant le Sprint;

Scrum accepte le changement pour le Sprint suivant, mais la durée fixe d'un Sprint en cours est faite pour se focaliser sur un objectif relativement stable, clair et limité. Chaque jour, l'Equipe se réunit brièvement afin de contrôler sa progression et ajuster les prochaines étapes nécessaires à la finalisation du travail restant.

A la fin de chaque Sprint, une revue est organisée avec les parties prenantes durant laquelle l'Equipe montre ce qu'elle a réalisé. Le feedback obtenu peut être pris en compte sur le Sprint suivant. Scrum insiste sur la nécessité de livrer un produit opérationnel à la fin de chaque Sprint, et réellement « terminé ». Dans le cas de logiciels, cela signifie un système intégré, entièrement testé, documenté pour ses utilisateurs et potentiellement déployable.

Les acteurs principaux de Scrum

- ❑ La **Scrum Team** qui est constituée de 7 personnes environ, choisies pour que l'équipe soit pluridisciplinaire et autonome. L'équipe n'a pas de hiérarchie interne.
- ❑ Le **Scrum Master** qui est le facilitateur de l'équipe, garant du respect de la méthode, de l'animation des événements et de la fabrication, et qui élimine les obstacles pouvant empêcher l'équipe de réussir
- ❑ Le **Product Owner** qui est le porteur métier du produit, qui définit le périmètre et les priorités de façon à maximiser la valeur du produit avec le budget alloué. Il se doit d'être constamment disponible pour répondre aux questions fonctionnelles de l'équipe. Il peut s'appuyer sur des experts métier. Il rend compte aux décideurs de l'entreprise de l'adéquation du produit avec les enjeux stratégiques qui ont motivé sa fabrication



XP : Extreme Programming

Officiellement proposée par Kent Beck, en octobre 1999.
Site de référence XProgramming.com

Définition de XP

Ensemble de pratiques à dimension humaine facile à mettre en œuvre, au prix d'un changement de mentalité et d'une acceptation par l'ensemble des équipes impliquées. Le principe est d'appliquer "à l'extrême" les principes du développement agile, en se focalisant sur les quatre points suivants :

- Besoins du Client,
- Individualités,
- Développement Itératif
- Intégration Continue



Adapted from xprogramming.com

XP se base sur les 12 pratiques ci-dessous :

Pratique	Description	But
Mettre tôt en production un système simple	Cette première mise en production intervient aussi tôt que possible dans le projet : il faut trouver le contenu fonctionnel minimal qui commence à avoir un sens pour les utilisateurs	Disposer rapidement d'un pilote opérationnel et réduire les risques dès le début du projet.
Design simple	Se focaliser sur les besoins actuels du client, ni plus, ni moins. Laisser la réalité du métier structurer et orienter la généricité du code	Lisibilité, concision, modularité, cohérence, maintenabilité.
Produit testé et validé à tout moment, et piloté par les tests (Test-Driven Development)	Les tests sont écrits d'abord et les développeurs codent ensuite avec comme objectif majeur que le code exécute les tests avec succès Au niveau fonctionnel, il s'agit des tests d'acceptation des stories, qui vont donc structurer le développement par le sens métier. Au niveau unitaire, les tests, automatisés lorsque possible, sont écrits pour chaque classe, chaque méthode, et vérifiés souvent.	Qualité externe optimale Focaliser le développement sur l'essentiel : que ça marche Donner au développeur la meilleure illustration possible des exigences : les tests d'acceptation
Collaborer avec client pour faire le planning	Le client attribue une priorité aux scénarii de développement. Les développeurs discutent le contenu de ces scénarii, définissent les tâches techniques sous-jacentes, les estiment, puis les réalisent	Se focaliser sur les buts principaux et différer les autres.
Développement itératif/incrémental, logiciel livré souvent	Un rythme de livraisons fréquentes soutient la motivation des développeurs, l'intérêt des utilisateurs et la validation du produit. Le fonctionnel de l'application est décrit en brefs scénarii qu'on peut implémenter tout de suite et que le client peut juger très tôt.	Réduire mésententes de compréhension et pouvoir réadapter la trajectoire
Rythme soutenable	L'équipe maintient sa capacité à rester efficace en ne surchargeant pas le planning et en s'aménageant des horaires raisonnables	Diminution du nombre d'erreurs, donc augmentation de la qualité.
Le Client est représenté sur place	Pour une communication optimum, le représentant du client et les développeurs travaillent dans le même espace physique.	Assister les développeurs

Métaphores	Principes simples d'analogies utilisés pour faciliter la compréhension de l'application (Fonctionnel ou architecture). La métaphore ne doit pas être une simplification abusive du concept initial, mais doit permettre de simplifier l'expression d'une chose complexe.	Clarifier les fonctionnalités à atteindre d'une façon qui soit synthétique et parlante
Programmer à 2	Le code est produit par deux développeurs (le pilote et le copilote). Les binômes changent au cours du projet. Support moral pour aller de l'avant quand on est bloqué.	Code robuste et de qualité. Inspection permanente et en temps réel par le copilote
Le code est à tous les programmeurs	- Chaque développeur doit pouvoir modifier toutes les parties du code si le besoin s'en fait sentir - Responsabilité collective	S'il faut changer quelque chose, on peut le faire tout de suite => travail à plein régime
Intégration continue	Le système, dans son intégralité, est régulièrement et fréquemment, assemblé puis testé.	Système plus stable et opérationnel rapidement. Bugs détectés plus vite
Standards de codage	Le code doit respecter une convention de nommage et de présentation afin d'être lisible par tous les membres de l'équipe	Code de style unifié. Règles pour communication claire et efficace

Safe

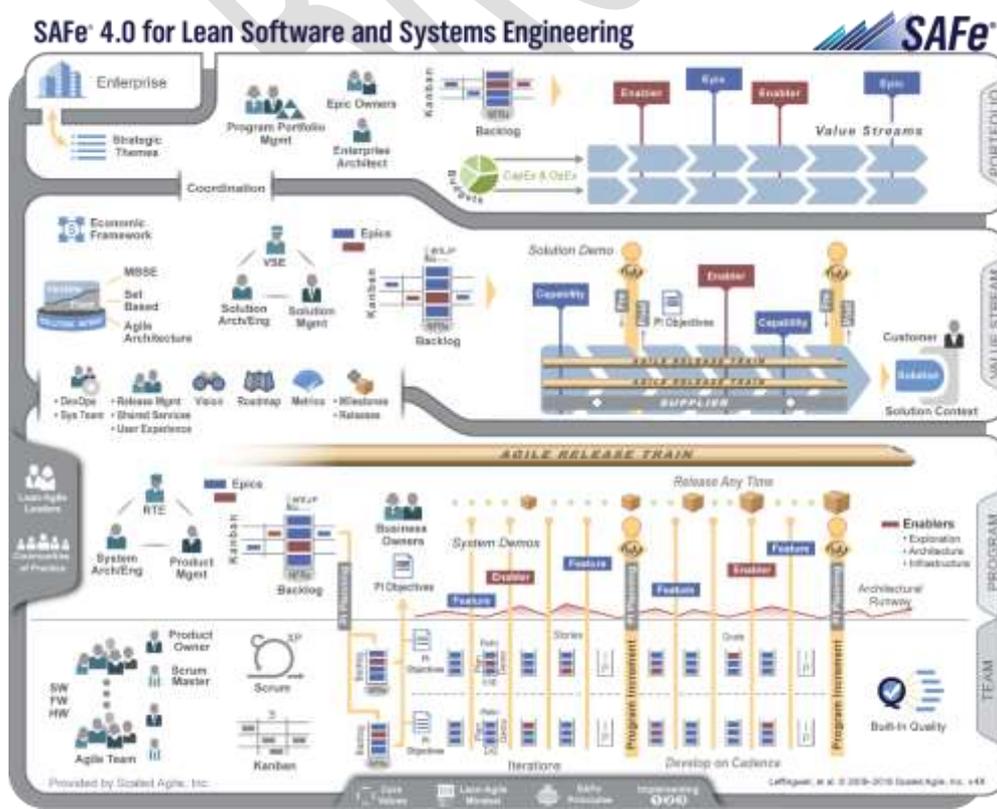
Créé par Dean Leffingwell et publiée en Version 1.0 en 2011

Site de référence : <http://scaledagileframework.com>.

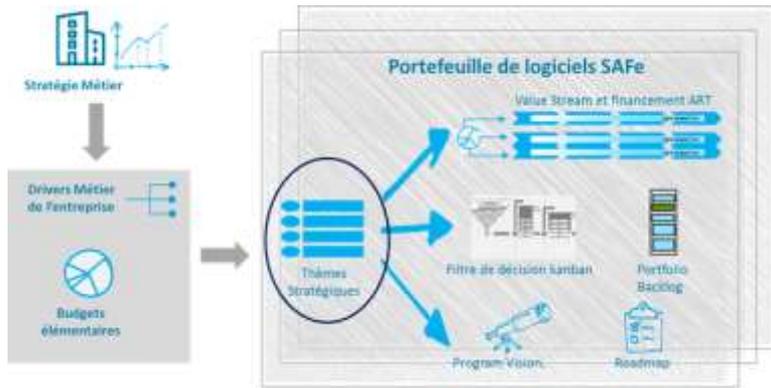
Cadre de développement logiciel Agile conçu par Scaled Agile, Inc. Il consiste en une base de connaissances de modèles intégrés destinés au développement **Lean-Agile à l'échelle de l'entreprise**. SAFe est évolutif et modulaire, ce qui permet à une organisation de l'appliquer de manière à répondre à ses besoins.

Le poster ci-dessous montre 3 strates principales. En bas la strate « équipe » qui représente des équipes de type « Scrum » qui travaillent ensemble de façon synchronisée pour produire la valeur globale attendue. Au-dessus, la strate « Programme » qui représente la fédération de ces équipes qui vont produire ensemble la valeur en intégrant leurs travaux élémentaires. La strate du haut représente la strate « Portfolio » qui remplace la gestion

classique de portefeuille de projets par un mode de gestion agile des initiatives métier et architecturales de l'entreprise, qui sont à l'origine des projets et des programmes.



Strate Portfolio



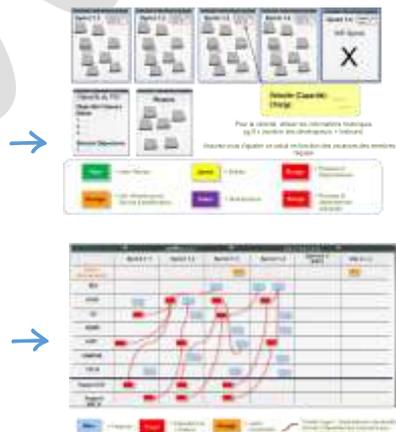
SAFe permet de réaliser un pilotage par la valeur au niveau Portefeuille en se rattachant à la stratégie de l'entreprise qui va piloter par la valeur toutes ses décisions opérationnelles et analyse leurs conséquences sur le long terme. La stratégie de l'entreprise est déclinée en une vision portefeuille qui identifie :

- Les chaînes de valeurs touchées par la stratégie
- Les thèmes d'investissement issus de la stratégie

A partir de cette vision on identifie les Initiatives métiers et techniques d'envergure (épics) dont on évalue la valeur de façon très régulière.

Strate Programme

Tous les deux mois, l'ensemble des équipes du programme sortent un incrément validé et potentiellement livrable, constitué d'une dizaine de fonctionnalités majeures. Pour être aligné sur les enjeux, les objectifs métier, les priorités et la cadence de développement, l'ensemble des équipes du programme se réunissent au début de cette période de deux mois pour s'aligner sur une vision métier commune, autour d'un événement de deux jours appelé « PI Planning » rassemblant l'ensemble des membres du programme : équipes de développement, architectes, acteurs métier, sponsors...au cours duquel les équipes définissent leurs contributions respectives aux objectifs, leurs interdépendances et la façon de livrer la valeur globale de façon optimale et sécurisée.



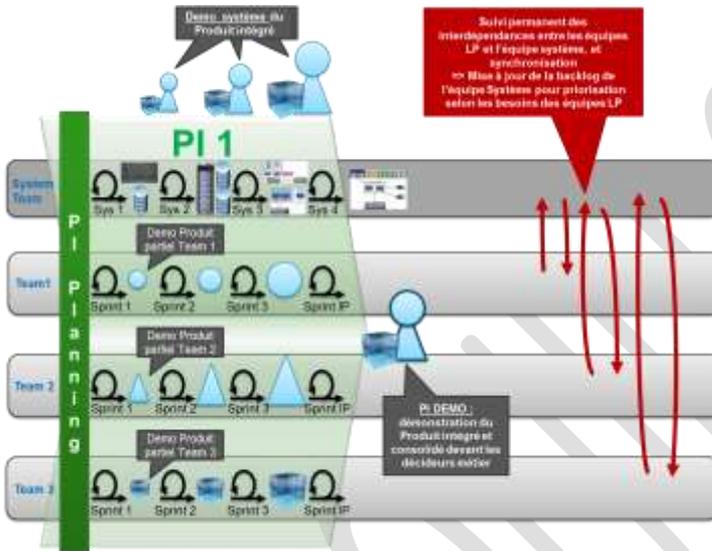
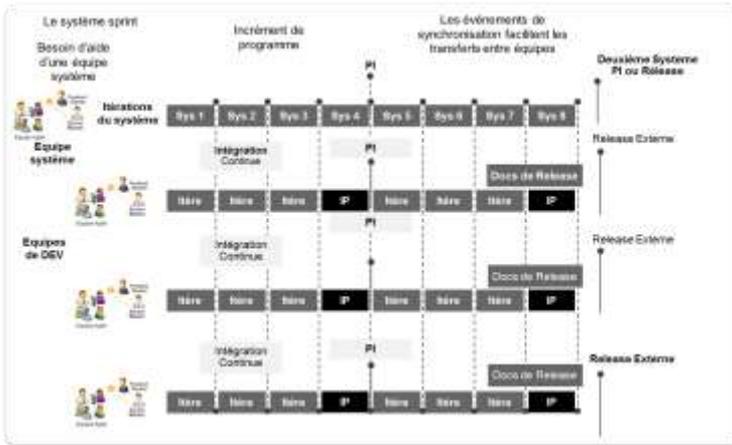
Un livrable essentiel du PI Planning : les stories de chaque équipe réparties sur chaque sprint, dépendances comprises. Les objectifs de l'équipe pour l'incrément de programme sont aussi écrits, puis consolidés

Un livrable essentiel du PI Planning : le « Program Board », qui montre les grandes fonctionnalités délivrées par les équipes sur les sprints, avec les dépendances inter-équipes

Après le PI Planning, les équipes de fabrication et une équipe technique transverse (« System Team ») lancent leurs 4 sprints tels que cadrés lors du PI Planning, en s'intégrant de façon continue, et en produisant une version opérationnelle intégrée du Produit pour chaque fin de sprint. Une démonstration du produit global est réalisée à la fin de chaque sprint.

Au bout des 4 sprints, une grande démonstration devant les acteurs métier majeurs du programme (sponsors, investisseurs, parties prenantes...) permet de valider l'incrément du produit réalisé (PI ou Program Increment) et de faire une inspection globale du programme pour adaptation via un plan d'action décliné de cette inspection.

Tout au long de la réalisation, les équipes se synchronisent à

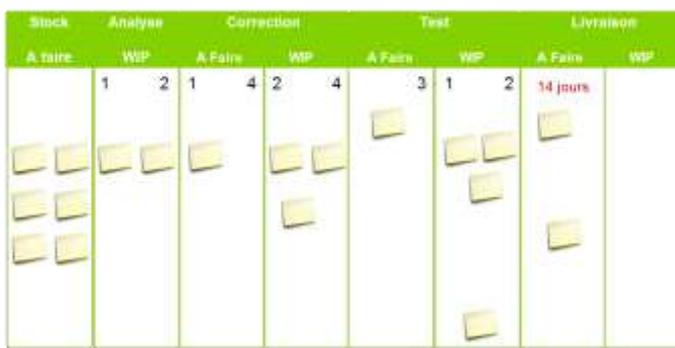


Kanban

La méthode Kanban énoncée par David J. Anderson est une méthode agile initiée en 2004 et publiée en 2009
Ouvrage de référence KANBAN de David J. Anderson



Kanban (« étiquette » en japonais) est à la base une méthode simple, visuelle et facilement compréhensible par tous. Créée à la fin des années 50 au Japon pour l'industrie automobile pour optimiser la gestion de flux. Elle est basée sur le principe du « juste à temps ». Ce sont les commandes émises par un poste aval en fonction de ses besoins, qui régulent la production d'un poste en amont. La méthode **kanban** est une méthode de gestion dite à « flux tiré » qui permet de réduire les délais, d'avoir moins de stocks, donc plus de trésorerie, moins de produits en fin de vie et plus de réactivité.



WIP : Work in Progress (travail en cours)

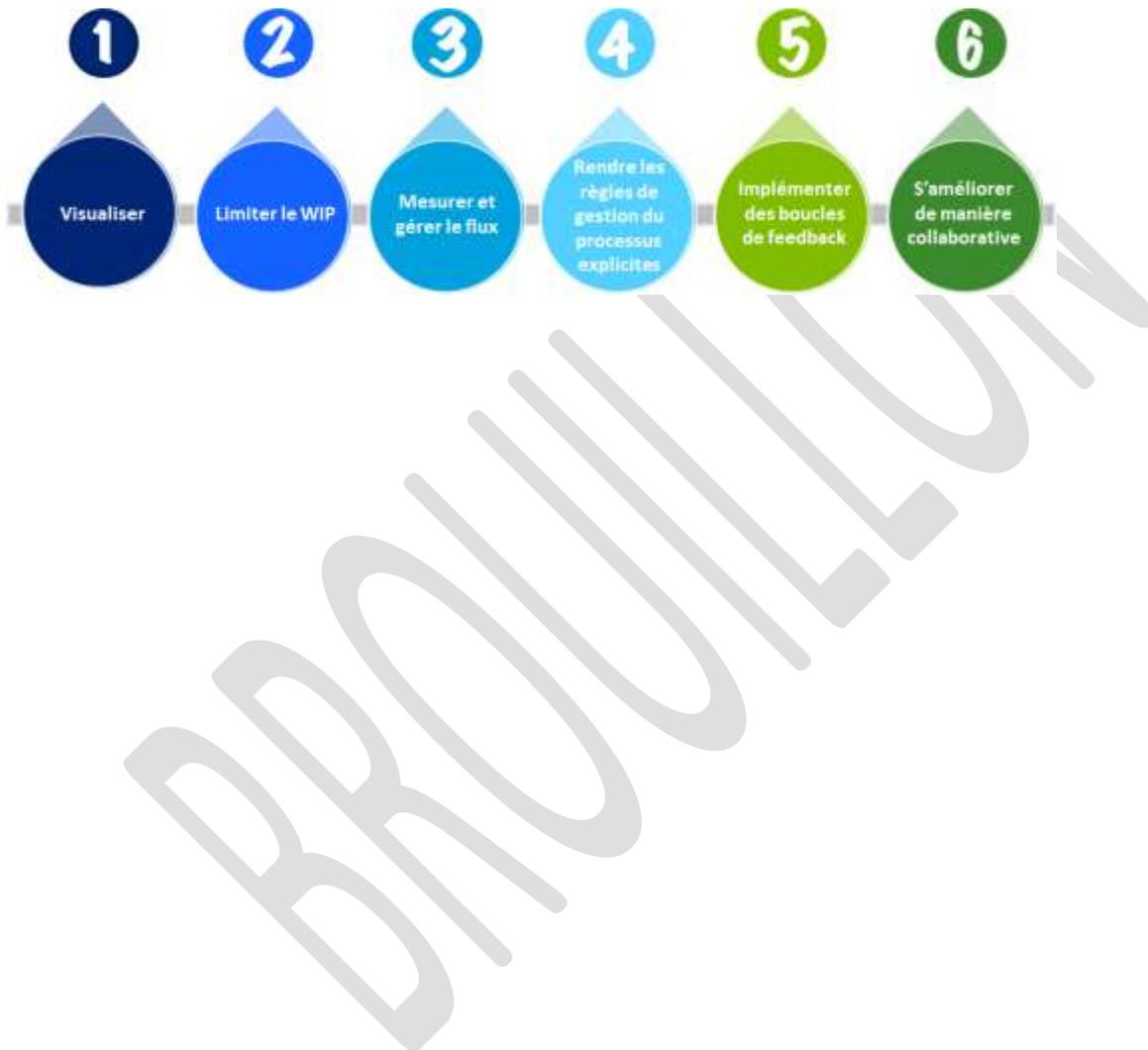
Kanban est ensuite devenue une pratique agile utilisée pour le pilotage du projet, avec les modifications suivantes sur le processus agile classique :

- itérations, estimations et vélocité sont abolies
- la mesure du temps de cycle se substitue à la vélocité
- tableau de tâches remplacé par un "tableau kanban" ayant les caractéristiques suivantes :
 - colonnes correspondant aux états par lesquels transite une "unité de valeur" (ex : user story)
 - limites de stock: si une activité se voit attribuer une limite de stock égale à 2, il sera "interdit" de démarrer l'activité concernant une troisième user story, si 2 y sont déjà
 - pas "réinitialisé" en début de chaque itération, c'est un tableau perpétuel dont le but est d'établir un flux permanent à optimiser

Les 4 piliers de la méthode Kanban



Les 6 pratiques de la méthode Kanban



c. Glossaire

Glossaire des concepts et événements de l'agilité

Terme	Définition	Source
Acceptance criteria	Les Acceptance Criteria d'une story définissent la façon dont l'équipe pourra savoir si l'intention d'une story a été atteinte ou non. Cela couvre plusieurs aspects, tels que : - le comportement du Système - les exigences non fonctionnelles	SAFe Academy
Acceptance Test Driven Development	Technique de développement qui entremêle la programmation, l'écriture de tests et l'activité de spécification des exigences. Les tests sont écrits avant les développements et les structurent, notamment par le biais des critères d'acceptation des User Stories. C'est donc "développer ce qu'on veut tester plutôt que tester ce qu'on a développé".	SAFe Academy
Backlog	Liste ordonnée de toutes les actions à entreprendre, souvent dans le cadre d'un projet informatique, mais pas seulement. Pour le développement d'un produit, c'est souvent la liste priorisée et qualifiée des stories, on parle de « Product Backlog »	SAFe Academy
Burn-down Chart	Représentation graphique du reste à faire dans une période (souvent un sprint), actualisé très souvent et permettant de montrer la tendance de l'avancement.	SAFe Academy
Burn-up Chart	Graphique montrant l'évolution d'une augmentation dans une mesure par rapport au temps. Les Burn-up charts sont une implémentation facultative au sein de scrum pour rendre les progrès transparents.	Scrum.org
Daily Scrum	Brève réunion d'Equipe qui se tient chaque jour, durant laquelle les membres inspectent leurs travaux, synchronisent leurs tâches et leurs avancements, et font état des obstacles au ScrumMaster afin de les supprimer. A la suite des Daily Scrum, des réunions peuvent avoir lieu pour adapter le travail à venir dans le but d'optimiser le Sprint.	ScrumPrimer
Definition of Done	Compréhension partagée des attentes auxquelles les logiciels doivent répondre pour pouvoir être livrés en production. Géré par l'équipe de development.	Scrum.org
Development Team	Rôle responsable de la gestion, de l'organisation et de l'exécution de tous les travaux de développement nécessaires pour créer un incrément de produit sur chaque Sprint.	Scrum.org
Élément du Product Backlog	Exigences fonctionnelles, exigences non fonctionnelles, et problèmes, priorisés par ordre d'importance pour le métier, avec leurs dépendances et leurs estimations. La précision de l'estimation dépend de la priorité et de la granularité de l'élément du Product Backlog, les éléments de plus forte priorité sélectionnables pour le prochain Sprint étant les plus fins et les plus précis.	ScrumPrimer
Feature team	Equipe autonome et transverse, aussi bien fonctionnellement que sur le plan des composants informatiques, focalisée sur la construction d'un produit informatique d'un bout à l'autre de la chaîne technique. Cette équipe regroupe toutes les compétences nécessaires, de l'idée à la mise en production, pour pouvoir réaliser le produit. Dans ce modèle, plus de silo : tous travaillent ensemble et sont co-localisés pour une communication optimale.	SAFe Academy
Incrément	Fonctionnalité du produit qui est développée par l'Equipe durant chaque Sprint et qui est potentiellement livrable ou utilisable par les parties prenantes.	ScrumPrimer
Itération	Synonyme de sprint	Scrum.org

Terme	Définition	Source
NFR : Non Functional Requirements	Exigences Non fonctionnelles décrivant des attributs du SI comme : sécurité, fiabilité, maintenabilité, adaptabilité ou utilisabilité. Peuvent aussi être des contraintes/restrictions sur la conception du SI. Les NFRs sont aussi critiques que les Features fonctionnelles ou les User Stories, car assurent l'utilisabilité et l'efficacité du SI entier. Les NFRs sont des qualités & contraintes persistantes et, à la différence des exigences fonctionnelles, elles sont revisitées à chaque itération	SAFe Academy
PI (Program Increment)	Unité de temps la plus vaste pour un développement de Programme SAFe, qui utilise la cadence et la synchronisation pour faciliter la planification, limiter le WIP (Work In Progress), agréger la valeur liée aux nouvelles fonctionnalités et assurer des rétrospectives cohérentes du niveau de programme.	SAFe Academy
PI Planning	Événement de planification fondamental, en face-à-face, basé sur la cadence, qui sert de battement de cœur du programme agile SAFe.	SAFe Academy
Product Backlog	Liste priorisée d'exigences avec estimation du temps pour les transformer en fonctionnalités finies du produit. Plus un élément possède une priorité élevée dans le Product Backlog, plus son estimation est précise. La liste émerge et évolue en fonction des changements du métier ou des technologies.	ScrumPrimer
Product Backlog refinement	Activité dans un Sprint par lequel le product owner et l'équipe de développement affinent la granularité du Product Backlog.	Scrum.org
Product Management	Responsable de l'identification des besoins du client, porteur de la vision et de la feuille de route d'un programme SAFe. Responsable des coûts, budgets du ROI et de la Backlog du programme. Ils pilotent les objectifs d'un incrément de programme et publient le contenu via les fonctionnalités prioritaires et les critères d'acceptation.	SAFe Academy
Product Owner	Personne en charge de gérer le Product Backlog, afin de maximiser la valeur. Le Product Owner est le représentant de toute personne ayant un intérêt dans le projet et dans son produit final.	ScrumPrimer
Reste estimé à faire	Estimation faite par l'Equipe du nombre d'heures restant à travailler sur une tâche donnée. Ces estimations sont revues chaque jour pour chaque tâche du Sprint Backlog. L'estimation correspond au total des heures restantes, indépendamment du nombre de personnes qui exécutent le travail.	ScrumPrimer
Rétrospective Sprint	Réunion animée par le ScrumMaster durant laquelle l'Equipe discute du Sprint qui vient de s'achever et détermine ce qui peut être modifié et qui rendrait le Sprint à venir plus agréable et plus productif.	ScrumPrimer
Réunion de Planification de Sprint	Réunion d'une durée limitée à quatre heures (pour un Sprint de deux semaines) qui marque le début de chaque Sprint. La réunion est divisée en deux parties de deux heures, également d'une durée fixe. Durant la première partie le Product Owner présente à l'Equipe les éléments du Product Backlog les plus prioritaires. L'Equipe et le Product Owner collaborent afin que l'Equipe puisse déterminer la quantité d'éléments qu'elle peut transformer en fonctionnalité durant le Sprint. Durant la seconde partie, l'Equipe détermine comment elle va implémenter ces éléments, en concevant et en détaillant son travail pour atteindre le But du Sprint.	ScrumPrimer
Réunion de Revue de Sprint	Réunion limitée à deux heures (pour un Sprint de deux semaines) se tenant à la fin de chaque Sprint. L'Equipe collabore avec le Product Owner et les parties prenantes et inspectent le résultat du Sprint. Cela commence généralement par une revue des éléments terminés, une discussion sur les opportunités, les contraintes et les découvertes, et une discussion sur ce qu'il est préférable de	ScrumPrimer

Terme	Définition	Source
	démarrer (ce qui peut donner lieu à des changements dans le Product Backlog). Seules les fonctionnalités terminées peuvent être présentées.	
Scrum Values	Ensemble de valeurs et de qualités fondamentales qui sous-tendent le cadre Scrum; L'engagement, l'attention, l'ouverture, le respect et le courage.	Scrum.org
ScrumMaster	Personne responsable du processus Scrum, de la conformité de son implémentation, et de l'optimisation de ses bénéfices.	ScrumPrimer
Self-organization	Principe de management selon lequel les équipes organisent leur travail de façon autonome. Les équipes choisissent la meilleure façon d'accomplir leur travail, plutôt que d'être dirigées par d'autres personnes en dehors de l'équipe.	Scrum.org
Spike	Type de story utilisé pour les activités comme la recherche, la conception, l'exploration et le prototypage. Le but d'un spike est de gagner la connaissance nécessaire pour réduire le risque d'une approche technique, mieux comprendre une exigence, ou augmenter la fiabilité de l'estimation d'une story. Peuvent être fonctionnelles ou techniques comme les autres stories. Chacune est estimée, prise en charge par un membre de l'équipe et démontrées en fin de sprint.	SAFe Academy
Sprint	Courte période avec Butée temporelle pendant laquelle les équipes livrent de la valeur de façon incrémentale, sous forme de logiciel s'exécutant et ayant été testé. Chaque sprint a un pattern standard : planification, engagement sur un objectif, exécution (implémentation & test) démonstration aux stakeholders ; et, finalement, Rétrospective, durant laquelle l'équipe analyse et fait les changements nécessaires à l'amélioration de la performance.	SAFe Academy
Sprint Backlog	Liste définissant le travail d'une Equipe durant un Sprint. Elle se décompose souvent en un lot de tâches plus détaillées. La liste est créée durant le Sprint Planning Meeting et peut être mise à jour durant le Sprint par l'Equipe, avec la suppression ou l'ajout de tâches en fonction des besoins. Chaque tâche du Sprint Backlog sera suivie durant le Sprint et montrera l'estimation du reste à faire.	ScrumPrimer
Sprint Goal (l'objectif du Sprint)	Courte expression du but d'un Sprint, souvent un problème métier abordé. La fonctionnalité pourrait être ajustée pendant le Sprint afin d'atteindre l'objectif du Sprint.	Scrum.org
Sprint Planning	Événement temporellement limité de 8 heures, ou moins, pour commencer un Sprint. Il sert à l'équipe Scrum d'inspecter le travail à partir du Backlog, de commencer avec ce qui a le plus de valeur, et à ensuite concevoir ce travail dans le Sprint Backlog.	Scrum.org
Sprint Retrospective	Événement temporellement limité de 3 heures, ou moins, pour finir un Sprint. Il sert à l'équipe Scrum d'inspecter le Sprint passé et de planifier des améliorations à adopter lors du prochain Sprint.	Scrum.org
Sprint Review	Événement temporellement limité de 4 heures, ou moins, pour conclure le travail de développement d'un Sprint. Il sert à l'équipe Scrum et aux parties prenantes pour inspecter l'incrément du produit résultant de Sprint, évaluer l'impact du travail effectué sur les progrès globaux et mettre à jour la backlog afin de maximiser la valeur de la prochaine période.	Scrum.org
Sprint	Itération, ou cycle répétitif de travaux similaires, produisant des incréments du produit ou du système. Ne dure pas plus d'un mois et généralement plus d'une semaine. La durée est fixée pour la totalité du projet et toutes les équipes travaillant sur le même produit ou système utilisent la même durée.	ScrumPrimer

Terme	Définition	Source
Stakeholder Partie prenante	Personne ayant un intérêt dans le résultat d'un projet, parce qu'il le finance, qu'il l'utilise, ou qu'il est concerné par celui-ci.	ScrumPrimer
Story	Les Stories (user, technique, infrastructure...) représentent le remplacement fait par l'agilité pour les formes traditionnelles d'exigences. Les Stories sont petites, de comportement indépendant et produisent de la valeur. Les Stories sont redécoupées au besoin pour tenir en une seule itération. L'expression première du comportement du système est la Story	SAFe Academy
Tâche du Sprint Backlog	Tâche que l'Equipe ou l'un de ses membres identifie comme nécessaire à la transformation d'un élément du Product Backlog en fonctionnalité du système et sur laquelle l'Equipe s'est engagée.	ScrumPrimer
Terminé	Finalisé conformément à ce qui est mutuellement convenu par toutes les parties et qui se conforme aux standards, aux conventions et aux directives. Quand quelque chose est identifié comme « terminé » lors de la Revue de Sprint, cela doit être conforme à cette définition.	ScrumPrimer
User Experience	Tandis que des Équipes Agiles ont la pleine responsabilité d'implémenter le code, y compris les éléments d'interface utilisateur, les travaux de UX designer sont de Niveau de Programme pour fournir des conseils & guides de conception transverses au programme afin de fournir une expérience d'utilisateur cohérente à travers les composants et les systèmes de la solution	SAFe Academy
Vision	C'est ce qui communique l'intention stratégique de la solution à développer. Pour les développeurs, elle décrit la façon de voir des stakeholders de la solution, selon les besoins de ces stakeholders et les fonctionnalités. La vision rend compte de l'essence de la solution envisagée sous forme de fonctionnalités de haut niveau, de contraintes non fonctionnelles et de contraintes de conception. En outre, la vision fournit une vue globale de la solution	SAFe Academy
WIP (Work in Progress)	Les limites de travail en cours (WIP) déterminent la quantité minimale et maximale de travail qui vit dans chaque état d'un flux de travail. Limiter la quantité de WIP améliore le débit et réduit la quantité de travail « presque terminée » en obligeant l'équipe à se concentrer sur un plus petit ensemble de tâches. Les limites de WIP mettent également en évidence les goulets d'étranglement d'une équipe avant qu'une situation ne devienne grave. Ces avantages garantissent des augmentations de valeur au client plus tôt, ce qui rend WIP limite un outil précieux dans le développement agile.	Atlassian

Glossaire des pratiques



Pratique	Définition
Agile Games	Jeux grâce auxquels l'état d'esprit ludique permet de faire sauter des verrous et fait manipuler concrètement des concepts, que ce soit sous forme de simulation ou analogie => aide à l'assimilation de théories complexes
Agile Modeling	Méthodologie consistant à modéliser et documenter des systèmes logiciels basés sur les meilleures pratiques. C'est une collection de valeurs et de principes, qui peuvent être appliqués sur un projet de développement (agile) de logiciel. Cette méthodologie est plus flexible que les méthodes de modélisation traditionnelles, ce qui correspond mieux à un environnement en rapide évolution
Acceptance Test Driven Development	Technique de développement qui entremêle la programmation, l'écriture de tests et l'activité de spécification des exigences. Les tests sont écrits avant les développements et les structurent, notamment par le biais des critères d'acceptation des User Stories. C'est donc "développer ce qu'on veut tester plutôt que tester ce qu'on a développé".
Atelier collaboratif	Méthode de communication interactive qui met en présence un personnel expérimenté et investi d'un pouvoir de décision durant une session de travail. L'atelier est un processus utile lorsqu'il est nécessaire de prendre des décisions, d'explorer des idées et d'échanger des connaissances en vue de résoudre un problème. Il doit être piloté par un « facilitateur » indépendant. Permet aux membres d'un groupe d'individus clés d'échanger des informations et de prendre collégialement des décisions qui seront donc acceptées de façon globale.
BDD (Behaviour-Driven Development)	Behaviour-Driven Development : réunir dans un même document des exigences (User Stories) exprimés selon le formalisme rôle-fonction-bénéfice et des scénarios ou exemples exprimés selon le canevas given-when-then,

Pratique	Définition
Built-in Quality	Pratiques qui garantissent que chaque élément de solution, à chaque incrément, satisfait aux normes de qualité appropriées tout au long du développement
Burn Down Chart	Représentation graphique de l'évolution de quantité de travail restante par rapport au temps sur une période donnée. Le travail restant se situe sur l'axe vertical, et le temps sur l'axe horizontal.
Collective Code Ownership	Responsabilité collective du code : S'il faut changer quelque chose, on peut le faire tout de suite => travail à plein régime
Communautés de pratiques	Groupe de personnes partageant une préoccupation, une passion, ou un ensemble de problèmes, autour d'un sujet, approfondissant leur expertise dans ce domaine en interagissant en continu. Sont enracinées dans l'auto-organisation et n'apparaissent pas sur l'organigramme. La participation est volontaire (passion d'apprendre ou de contribuer).
Continuous integration ou Intégration continue	Ensemble de pratiques utilisées en génie logiciel consistant à vérifier à chaque modification de code source que le résultat des modifications ne produit pas de régression dans l'application développée. Bien que le concept existât auparavant, l'intégration continue se réfère généralement à la pratique de « extreme programming ». Pour appliquer cette technique, il faut d'abord que : <ul style="list-style-type: none"> - le code source soit partagé (en utilisant des logiciels de gestion de versions) - les développeurs intègrent (commit) quotidiennement (au moins) leurs modifications des tests d'intégration soient développés pour valider l'application Les principaux avantages d'une telle technique de développement sont : <ul style="list-style-type: none"> - le test immédiat des modifications - la notification rapide en cas de code incompatible ou manquant - les problèmes d'intégration sont détectés & réparés continuellement, évitant les problèmes urgents - une version est toujours disponible pour un test, une démonstration ou une distribution
Continuous deployment	L'équipe s'appuie sur une infrastructure qui automatise l'ensemble des étapes de déploiement, de sorte qu'après chaque intégration dont les tests passent avec succès, l'application en production est mise à jour dans la foulée. But : réduire le plus possible le temps de cycle, écoulé entre l'écriture d'une nouvelle ligne de code et l'utilisation réelle de ce code par les utilisateurs finaux.
Daily Scrum	Événement quotidien de l'agilité (aussi appelée Daily Standup). L'équipe se réunit une fois par jour, à heure fixe, pour mettre en commun les apports de chacun au produit, et signaler les obstacles rencontrés. C'est aussi le moment pour donner de la visibilité sur ce qui va être fait dans la journée à venir et pour créer de l'entraide en cas de problème.
DevOps	Mouvement visant à réduire la friction organisationnelle entre les « DEV » (les développeurs, chargés de faire évoluer le système d'information) et les « OPS » (les exploitants, chargés d'exploiter les applications existantes, donc qui veulent plutôt sa stabilité et tendent à préférer peu de changements). Ce que l'on pourrait résumer en « travailler ensemble pour produire de la valeur pour l'entreprise ». Dans la majorité des entreprises, la valeur sera économique mais pour d'autres, elle sera sociale ou morale.
Hack days	Événements au cours desquels des membres d'une équipe collaborent intensivement sur des sujets qui les concernent, et essayent des idées nouvelles et échangent entre eux sur leurs connaissances et outils. Permet d'être régulièrement remis à jour et up-to-date avec les nouveaux outils, les nouvelles techniques et les importantes innovations.
Hackatons	Le terme est un mot-valise constitué de hack et marathon. C'est à l'origine un rassemblement de développeurs organisés par équipe autour de porteurs de projet avec l'objectif de produire un prototype d'application en quelques heures. Sous forme de concours chronométré, l'équipe gagnante est généralement désignée par un jury à l'issue du

Pratique	Définition
	temps imparti. La référence au Marathon se justifie par le travail sans interruption des développeurs pendant deux jours, généralement lors d'un week-end ¹ .
3C (Card, Conversation, Confirmation)	Card : story courte, 1 ou 2 phrases et peut être écrite sur une carte 8x13 cm, c'est mieux Conversation : Les détails d'une story sont discutés par les équipes avec métier, ergonomes ...pour son amélioration Confirmation : La story est confirmée par des tests d'acceptation rédigés au même moment
INVEST	Utilisation de six critères pour la définition des Stories : Independent, Negotiable, Valuable, Estimable, Simple, Testable
Planning Poker	Méthodologie de chiffrage en mode « Atelier collaboratif animé » dans lequel les participants utilisent un jeu de cartes avec des nombres pour annoncer leurs chiffrages sur des fonctionnalités décrites par un référent métier. Le but du jeu de cartes étant de permettre à chacun de présenter son chiffrage sans être influencé par les autres, puis de le défendre ensuite. Les discussions qui s'ensuivent permettent que le chiffrage soit réalisé par la confrontation d'une multiplicité de perspectives et par un « brain storming » à la fois engageant, remportant l'adhésion finale, et le plus exhaustif possible.
Pair Programming	Deux programmeurs partagent un poste de travail, se répartissant les rôles entre un "conducteur" (aux commandes) et un "copilote" (surveillant l'écran et intervenant en cas de besoin), intervertissant les rôles fréquemment.
Personas	Persona : définit un utilisateur-type, une représentation fictive des utilisateurs cibles. Objectif : Ne pas perdre de vue la notion « utilisateur » d'un système
Refactoring	L'équipe se réunit périodiquement, pour réfléchir explicitement sur les événements saillants depuis la précédente réunion, et décider ensemble des actions d'amélioration ou de remédiation suggérées
Test Driven Development	Technique de développement qui entremêle programmation, écriture de tests et remaniement. Les tests sont écrits avant les développements et les structurent => "développer ce qu'on veut tester plutôt que tester ce qu'on a développé".
Time boxing	Principe consistant à fixer des butées temporelles sur les travaux à réaliser. Une fois la date de fin des travaux fixée ; elle ne bouge plus, et si on estime que les objectifs ne peuvent être atteints, on les révisé à la baisse et on livre moins que prévu, ce qui oblige à se focaliser sur le plus important.
Visual Management	Démarche de management qui trouve son origine dans l'industrie automobile japonaise comme pilier du Toyota Production System (TPS). Il privilégie le management de proximité, l'écoute et l'interaction avec les collaborateurs au moyen d'outils simples, essentiellement un tableau. Il est le point de rencontre entre le manager et son équipe. Le concept se fonde sur une réflexion basée sur les sens : on intègre beaucoup mieux ce qu'on voit que ce qu'on entend, et beaucoup mieux ce qu'on écrit que ce qu'on lit...

d. Exemples de modalités de sélections des opérateurs économiques

Critères de dépouillement :

Critères	Critère prix	Critère technique
Pourcentage	30%	70%
Recommandations	Préciser un scénario de commande (ensemble d'unités d'œuvre pouvant être cohérent avec le projet et juger les candidats sur ce scénario).	Sous-critères proposés (à adapter au contexte) : <ul style="list-style-type: none"> • Profils des intervenants pressentis (demander des informations ciblées ; cf. ci-joint) • Compétences et expériences similaires (demander au minima 5 expériences similaires) • Compréhension du contexte et du besoin (une manière assez efficace pour juger de la compréhension du candidat sur l'attendu) • Méthodologie et organisation proposée (demander à minima un planning avec l'enchaînement des sprints, l'équipe type et les rôles de chaque membre de l'équipe, un résumé de l'attendu des unités d'œuvre)

Un cadre de réponse limité en nombre de pages est recommandé.

Exemple de fiche profil adapté aux exigences méthodes Agiles

Séniorité du profil : en année d'expérience

Grade dans l'entreprise (cf. convention SYNTEC par exemple) :

Diplôme : Ingénieur, Ecole de commerce, Universitaire, DUT,...

Durée de présence dans l'entreprise : en mois et années

Compétences :

- **Agilité :** méthodes connues et utilisées
- **Développement :** langages,...
- **Test :**
- **Gestion de projet :**
- **Autres :** ergonomie, design, sécurité, BD,....

Références :

- **Par référence, préciser :** Entreprise, durée de présence mission, description des activités réalisées par le profil, exemples de livrables réalisés/collaboré, rôle dans l'équipe.

Certifications ou formations liées à l'agilité :

Publications sur l'agilité :

Exemple de modèles d'expérience client

Nom / type entreprise cliente :	Taille entreprise cliente:
Année :	Durée de la mission :
Descriptif du projet / cadre de la mission :	
Rôle de votre entreprise sur la mission :	
Exemple de livrables produits :	
Equipe mobilisée: <i>Nombre de profils, type de profils</i>	
Méthode de projet agile :	Nombre de sprints :
Taille du backlog initial estimé :	Nombre de releases :
Autres précisions sur la mission (réussites, éléments distinguant, point de contact client,...) :	

Exemples d'obligations sur les équipes du titulaire (non exhaustif)

EQUIPE DU TITULAIRE

Chaque titulaire s'engage, à compter de la signature du présent marché, à affecter une équipe d'intervenants opérationnels pendant toute la durée d'exécution des prestations. Par ailleurs, chaque titulaire s'engage à faire bénéficier à l'entité bénéficiaire, de tout son savoir-faire, de ses méthodes et de son expérience, concrétisé par l'intervention de son personnel, professionnel et compétent dans le domaine de chaque prestation objet de la présente opération.

En conséquence, chaque titulaire s'engage notamment à :

- constituer des équipes de personnels compétents, c'est-à-dire formés en conformité avec les exigences de l'entité bénéficiaire telles que stipulées dans le CCTP ;
- veiller et contrôler le maintien constant des compétences ;
- maintenir une forte réactivité, notamment en adaptant très rapidement ses équipes le cas échéant en termes de nombre.

Afin d'assurer le succès des prestations, le titulaire s'engage à prendre toutes les mesures nécessaires et possibles pour maintenir tout au long de l'exécution des prestations, des personnels homogènes par prestation, disponibles et réactifs.

La bonne exécution des prestations dépend essentiellement des personnes qui s'y trouvent nommément désignées pour en assurer la conduite. Si ces personnes ne sont plus en mesure de remplir leur mission, le titulaire doit en aviser immédiatement l'organisme contractant concerné et prendre toutes les dispositions nécessaires pour que la bonne exécution des prestations ne s'en trouve pas compromise.

Le porteur de lot et les membres de l'équipe du titulaire sont considérés comme acceptés si l'entité bénéficiaire ne les récuse pas dans le délai d'un mois à compter de la date de leur mise à disposition. Si l'entité bénéficiaire récuse un ou plusieurs de ces profils, le titulaire dispose de 15 jours pour désigner un remplaçant et en informer l'entité bénéficiaire.

En cas d'absence répétée et injustifiée du porteur de lot ou de l'un des membres de l'équipe, ou en cas de non remplacement après récusation dans le délai de 15 jours visé ci-dessus, l'entité bénéficiaire se réserve la possibilité de recourir aux pénalités prévues à l'article X du présent accord-cadre.

Enfin, chaque titulaire sera seul responsable de son personnel intervenant dans le cadre du marché, qui demeure sous son contrôle et sa seule autorité hiérarchique. Chaque titulaire est tenu au respect des normes d'hygiène et de sécurité ainsi qu'à l'information complète de son personnel en ce qui concerne lesdites normes et lesdites obligations visées dans le règlement intérieur applicable dans les locaux de l'entité bénéficiaire, et/ou autres lieux d'intervention.

GARANTIE DE CONTINUITÉ DES PRESTATIONS

En cas d'indisponibilité pour raison de force majeure, maladie, ou congés de l'un des membres du personnel du titulaire affecté à l'exécution des prestations dans le cadre du marché, chaque titulaire prendra les moyens nécessaires pour assurer la continuité de ses prestations et remplacera le personnel défaillant par du personnel de compétence et de qualification équivalentes. Il s'engage à assurer la continuité des prestations dans les mêmes conditions de qualité, de délais et de coûts.

En cas de démission d'un membre du personnel du titulaire, et si le besoin est confirmé par l'entité bénéficiaire par courrier électronique, chaque titulaire prendra les moyens nécessaires pour assurer la mise à niveau et le transfert de connaissance du nouvel entrant, et ce sans engager de surcoût pour l'entité bénéficiaire et dans la limite de dix jours ouvrés.

BROUILLON

BROUILLON



Direction interministérielle du numérique et du système d'information et de communication de l'Etat (DINSIC)

Contact : dinsic-sec.sgmap@modernisation.gouv.fr