

SCRUM : 10 RÉPONSES À VOS QUESTIONS LES PLUS FRÉQUENTES

Dans notre précédent webinar "Comprendre Scrum en 30 minutes", nous avons introduit la méthodologie Agile la plus populaire et la plus répandue au monde. De plus en plus d'entreprises songent à son adoption, non seulement pour le développement de logiciels mais également dans le cadre plus général d'une transition agile d'un point de vue organisationnel.

Scrum partage avec d'autres cadres de travail et méthodes (comme Agile Project Management) les principes de base des méthodes Agile, qui explorent le développement métier et la réalisation de produits et services à travers des cycles de travail courts, en s'adaptant rapidement et en temps opportun à des situations qui évoluent rapidement et à un temps de mise sur le marché de plus en plus comprimé.

Pour citer le Manifeste Agile de 2001, pour travailler de manière agile, il est plus important que vous sachiez comment réagir au changement (approche adaptative) plutôt que de simplement suivre un calendrier, qui risque de devenir bientôt obsolète (approche prédictive).

Scrum est basé sur la théorie du contrôle empirique des processus, à travers trois concepts clés (piliers):

- **Transparence:** vous devez partager un langage commun et une compréhension de ce qui se passe et des objectifs à poursuivre avec tous les acteurs du projet.
- **Inspection:** ce que vous produisez dans un cycle de travail Scrum doit être inspecté fréquemment, car c'est la meilleure garantie pour intercepter tous les problèmes dès que possible. ("Même les plus gros problèmes auraient pu être facilement résolus quand ils étaient encore petits " Lao Tzu).
- **Adaptation:** si, face à l'inspection, vous vous rendez compte que certains aspects du processus de production ne sont pas acceptables ou que le produit final risque de ne pas être accepté, vous devrez adapter les processus et les matériaux le plus rapidement possible afin de minimiser les écarts.

La structure de Scrum est légère et ... agile. Comme le disent ses créateurs, Ken Schwaber et Jeff Sutherland, il est "facile à comprendre mais difficile à maîtriser". Pour cette raison, que vous découvriez Scrum ou que vous l'ayez déjà utilisé, vous n'arrêterez jamais d'apprendre et de vous poser des questions quant à la meilleure façon de faire face à certaines situations. C'est pourquoi nous avons souhaité répondre aux 10 questions les plus fréquemment posées lors de nos formations et webinaires.

1) Comment la Revue de Sprint et la Rétrospective de Sprint sont-elles organisées?

La Revue de Sprint et la Rétrospective de Sprint ont lieu à la fin d'un sprint et avant la réunion de planification du prochain sprint. Ces deux événements sont l'exemple complet de la façon d'appliquer les trois piliers de la transparence, de l'inspection et de l'adaptation.

La revue de sprint a pour objet d'inspecter l'incrément réalisé et adapter le product backlog si nécessaire. Pendant la revue de sprint, l'équipe Scrum et les parties prenantes échangent sur ce qui a été fait durant le sprint. Qui participe :

- Les participants incluent l'équipe Scrum et les principales parties prenantes (client, utilisateurs) invitées par le Product Owner, lui même présent.
- Le Scrum Master, qui s'assure que l'événement ait lieu et que les participants en comprennent son but. Il facilite son déroulement et s'assure du respect des délais.
- L'Équipe de Développement, discute de ce qui s'est bien passé pendant le sprint, quels problèmes ont été rencontrés, et comment ces problèmes ont été résolus (transparence). Les participants collaborent et discutent des prochaines choses qui pourraient être faites pour optimiser la valeur (adaptation).

La rétrospective de sprint aura ensuite lieu, impliquant uniquement l'équipe Scrum (Product Owner + Scrum Master + Équipe de Développement) sans parties prenantes externes. La rétrospective de sprint (sprint retrospective) est une opportunité pour l'équipe Scrum de s'auto-inspecter et de créer un plan d'améliorations à adopter au cours du prochain sprint.

L'objectif est d'inspecter la manière dont le dernier sprint s'est déroulé en ce qui concerne les personnes, les relations, les processus et les outils, d'identifier et ordonner les principaux éléments qui ont bien fonctionné et des améliorations potentielles ; et, créer un plan pour mettre en œuvre des améliorations sur la façon dont l'équipe Scrum fait son travail.

Il est possible d'utiliser un concept bien connu de gestion de projet, les fameuses "lessons learned" (leçons apprises), qui sont collectées lors de la rétrospective mais à la différence des projets traditionnels, dans un projet Agile, elles sont partagées à partir du prochain sprint, sans avoir à attendre la fin du projet.

2) Combien de temps la revue de sprint et la rétrospective de sprint devraient-elles durer ? Doivent-elles être effectuées le même jour et à quel moment de la journée est-il préférable de les faire ?

La durée de la revue de sprint et de la rétrospective de sprint dépendent de la durée du sprint. Plus le sprint est long en termes de semaines, plus il y aura de choses à discuter et donc les deux événements seront plus longs.

À titre indicatif, pour un sprint d'un mois (4 semaines), la revue de sprint ne devrait pas durer plus de 4 heures et la rétrospective de sprint ne devrait pas durer plus de 3 heures. Elles peuvent donc être organisées le même jour, le matin la revue de sprint (qui vient toujours en premier) et l'après-midi la rétrospective. La durée d'un sprint étant mesurée en semaines, la meilleure solution est d'organiser les deux événements le vendredi de la dernière semaine du sprint. De cette façon, le week-end devient la frontière entre deux sprints successifs et la prochaine planification de sprint peut avoir lieu le lundi matin.

3) Comment créer des user stories ? Doivent-elles nécessairement tracer les caractéristiques individuelles d'un logiciel ?

Les user stories représentent généralement toutes les fonctionnalités, fonctions, exigences, améliorations et corrections du produit à réaliser. La meilleure façon de les créer est d'utiliser des descriptions courtes et simples d'une caractéristique ou fonctionnalité du point de vue de l'utilisateur ou du client du système. Pour faciliter votre travail, vous pouvez utiliser le modèle suivant :

En tant que <type d'utilisateur>, **je veux** <but> **afin de pouvoir** <motif valable>

Par exemple: En tant qu'administrateur système, je souhaite pouvoir gérer les comptes afin de pouvoir ajouter de nouveaux utilisateurs au système.

En utilisant cette technique, vous êtes amenés à identifier pour chaque exigence :

- En tant que => le type d'utilisateur / de partie prenante à qui la fonctionnalité est destinée
- Je veux / Je souhaite => la fonctionnalité spécifique
- Afin de => la valeur / l'avantage que cette fonctionnalité apporte

Ainsi vous pourrez créer des user stories significatives, concises et compréhensibles par n'importe qui, même par ceux qui ne sont pas des techniciens ou ne font pas partie de l'équipe.

4) Une user story devrait-elle plus ou moins correspondre à une activité de combien de jours ou d'heures? À quelle granularité devons-nous nous arrêter dans la répartition des activités en user stories ?

Lorsque vous estimez des user stories, vous ne rapportez pas immédiatement en jours ou en heures, car souvent il n'y a pas suffisamment d'informations pour pouvoir exprimer avec précision cette mesure. N'oubliez pas que les cycles agiles sont nés précisément pour surmonter les limites d'une planification initiale trop détaillée, menée lorsqu'il n'y a pas encore suffisamment d'informations et qui est donc inévitablement destinée à être refusée.

La complexité des user stories est mesurée en "story points", qui sont une unité relative, pas directement ou pas nécessairement liée à une quantification temporelle. Relatif signifie que vous devez prendre comme référence une user story que vous savez bien mettre en œuvre, lui donnant la valeur conventionnelle de 1 story point. Ensuite évaluez les autres par rapport à cela. Une user story à 3 points est une user story qui, à votre avis, équivaut à 3 fois - en termes de complexité et d'effort à dépenser - celle de référence.

Le nombre de story points qui peuvent être mis en œuvre par l'Équipe de Développement lors d'un sprint est appelé "vélocité" et dépend à la fois de l'équipe et du contexte du projet. Au début, la vitesse est sensiblement inconnue et vous pouvez vous fier aux données historiques des projets précédents avec des équipes de même taille. Vous pouvez ensuite le mesurer au fur et à mesure des sprints puis, en fonction de celui-ci, revenir à un engagement en jours / personne, sachant qu'un sprint équivaut à un certain nombre de jours de travail.

Par exemple: si les sprints durent 4 semaines et que l'Équipe de Développement est composée de 5 personnes, chaque sprint aura un réservoir de 100 jours / personne d'effort potentiel. Si la vélocité détectée est, disons, de 50 story points, cela signifie que pour cette équipe et pour ce projet, 1 story point équivaut à 2 jours / personne. Ensuite, vous pouvez tracer l'évaluation des user stories jusqu'à une mesure temporelle absolue.

D'une manière générale, il n'y a pas de granularité optimale. Bien sûr, une user story ne doit pas être trop volumineuse et doit pouvoir être implémentée en un seul sprint. Si ce n'est pas le cas, vous devrez la décomposer en plusieurs petites user stories. Disons qu'en général l'horizon temporel d'une user story peut varier entre 10 et 20 jours / personne d'effort, sachant qu'alors, pendant le sprint, les user stories sont analysées et décomposées par l'Équipe de Développement en tâches de 1 à 2 jours / personne, c'est-à-dire dans des unités de travail qui peuvent garantir une mesure des progrès au niveau du daily Scrum.

5) Avant de commencer un projet, est-il nécessaire de le décomposer complètement en epics / stories et de définir tous les temps de développement pour quantifier le travail ?

D'après mon expérience, nous finissons par ne pas pouvoir respecter les temps qui sont assez longs. Si au début du projet vous essayez de le décomposer complètement en epics / stories, de définir à l'avance tous les temps de développement et de quantifier pleinement le travail, vous ne pouvez pas être surpris si les temps s'allongent et vous ne pouvez pas respecter les délais. En fait, vous appliquez la planification prédictive traditionnelle. Le cycle de travail Agile, en revanche, fonctionne différemment précisément pour surmonter les limites de la gestion traditionnelle. Plutôt que de fixer à l'avance la portée, c'est-à-dire l'ensemble complet des fonctionnalités du produit final, puis de déterminer les délais et les coûts en conséquence, ceux-ci sont définis et la portée est laissée libre de varier dans certaines limites. En d'autres termes, l'ensemble des fonctionnalités essentielles (MVP, produit minimum viable) sera décidé au début, qui sera un sous-ensemble des fonctionnalités finales, mais pendant les sprints, les fonctionnalités qui seront réellement ajoutées aux fonctionnalités minimales pour terminer le produit seront déterminées en fonction du budget et du temps disponible (ex: 3 mois), en recherchant sprint après sprint pour implémenter les fonctionnalités qui apportent la valeur métier maximale au produit final.

6) Il n'est pas clair pour moi si le Scrum Master fait partie de l'Équipe de Développement ou s'il peut également être une figure avec des compétences techniques, mais qui s'occupe de la partie la plus organisationnelle de l'équipe.

Le Scrum Master peut éventuellement également faire partie de l'Équipe de Développement avec un rôle technique. En tant que Scrum Master, vous êtes responsable de faciliter le travail de l'équipe et de vous assurer qu'elle suit correctement les processus Scrum. Vous pouvez soit jouer uniquement ce rôle, qui demande beaucoup d'efforts, soit être également membre de l'Équipe de Développement. Dans ce cas, votre nomination en tant que Scrum Master pourrait être prise directement par l'équipe.

7) Est-il possible d'annuler un sprint et si oui, qui peut le faire ?

Vous pouvez annuler un sprint uniquement si vous êtes le Product Owner, généralement en consultant le Scrum Master et l'équipe de toute façon. Vous aurez rarement recours à cette possibilité, généralement lorsque l'objectif du sprint devient obsolète en raison de changements stratégiques affectant l'organisation, le marché ou la technologie.

8) Qui devrait mener la mêlée quotidienne. L'équipe ou le Scrum Master ?

Le daily Scrum autrement appelé la mêlée Scrum quotidienne est un événement quotidien de 15 minutes qui sert à l'Équipe de Développement pour synchroniser les activités et créer un plan pour les prochaines 24 heures, vérifier le travail effectué depuis la dernière réunion et prédire le travail qui aura lieu jusqu'à la prochaine réunion, le lendemain.

Une équipe Agile, quelle que soit la méthode spécifique adoptée, est interfonctionnelle et surtout auto-organisée. L'Équipe de Développement est donc responsable de la conduite du daily Scrum, favorisant un processus décisionnel partagé. La tâche du Scrum Master est plutôt de veiller à ce que seule l'Équipe de Développement se réunisse pour cette réunion, facilitant son développement et son enseignement afin de respecter la durée maximale de 15 minutes.

9) Le Scrum Master correspond-il au chef de projet d'un projet traditionnel ?

Non. L'organisation et la dynamique relationnelle entre les acteurs d'un projet Agile sont différentes de celles traditionnelles. La structure hiérarchique de la gestion traditionnelle est remplacée par le principe de délégation et d'autonomie technique de l'équipe, donc une organisation sensiblement plate. Les responsabilités du chef de projet d'un projet traditionnel dans le projet Agile sont réparties entre les 3 principaux rôles : le Product Owner, le Scrum Master et l'Équipe de Développement. Le premier est responsable du produit à fabriquer et de la conformité de ce qui a été réalisé avec les besoins réels du client. Le second est un facilitateur, un coach pour l'équipe, en un mot un servant leader qui s'assure que Scrum est compris et approuvé par tous et qui supprime les obstacles nuisant au travail de l'Équipe de Développement qui, à son tour, a la responsabilité de décider comment décomposer les user stories en tâches élémentaires, les répartir entre les membres et assurer la croissance des produits à la fin du sprint.

10) Qui est responsable du product backlog ?

Le Product Owner est celui qui a les compétences et la responsabilité pour tout ce qui concerne le produit à créer. Il représente la «voix du client» et garantit que la valeur métier maximale soit libérée pour chaque sprint. Pour ces raisons, en tant que Product Owner, vous êtes le propriétaire et le responsable du product backlog, qui décrit toutes les fonctionnalités, fonctions, exigences, améliorations et corrections pour le produit à créer.