

Lexique d'adaptation de la méthode agile Scrum

à destination de l'enseignant pour encadrer les projets NSI

I) Contexte d'un projet

Projet

Un projet est un effort temporaire entrepris dans le but de créer un produit, un service ou un résultat unique.

Exemples :

- développement d'un logiciel ou d'un composant logiciel ;
- mise en place d'une nouvelle organisation dans une entreprise ;
- conception d'un nouveau véhicule électrique ;
- acquisition d'un nouveau logiciel de gestion ;
- construction d'un bâtiment ou d'une autoroute.

Contre exemples :

- production industrielle du véhicule électrique - travail à la chaîne ;
- saisie de la comptabilité (factures, bons de commandes) au fil de l'eau d'une entreprise ;
- saisie de l'appel au début de chaque heure de cours par l'enseignant ;
- maintenance d'un bâtiment, d'une autoroute.

Ouvrage

On appelle ouvrage le résultat physique ou intellectuel visé par le projet.

Œuvre

On appelle œuvre l'ensemble des activités réalisées au cours du projet.

Maîtrise d'Ouvrage ou Maître d'ouvrage (MOA)

Le MOA, dans le contexte informatique, est l'entité qui a besoin de réaliser un projet (intégration, développement logiciel, étude, migration, etc.) et qui ne dispose pas des compétences pour le mener à bien. En synthèse : le MOA est le client.

Maîtrise d'oeuvre ou Maître d'oeuvre (MOE)

Le MOE est l'entité à qui le MOA fait appel pour réaliser le projet. En synthèse : le MOE est le prestataire de service : Entreprises de Service du Numérique (ESN), anciennement nommées Sociétés de Services et d'Ingénierie en Informatique (SSII).

Assistance maîtrise d'ouvrage (AMOA ou AMO)

L'AMO est une entité tierce entre le MOA et le MOE apportant assistance au MOA dans le déroulement du projet. L'assistance peut être d'ordre technique, administrative, fonctionnelle, etc.

Cahier des charges

Le cahier des charges est le document décrivant le périmètre et le contenu du projet.

Il présente :

- les objectifs que doit atteindre la solution,
- les exigences que doit satisfaire la solution,
- les exigences en terme de gestion de projet.

En résumé : document décrivant les besoins du client, exprimé par le client.

II) Généralités et vocabulaire autour des exigences

Exigence (Source : IEEE)

Condition ou capacité que doit présenter la solution développée pour satisfaire un contrat, un standard, une spécification ou tout autre document formel imposé.

Les exigences sont construites à partir du cahier des charges.

Exigence fonctionnelle : Exigence décrivant une fonctionnalité du système. Les exigences fonctionnelles permettent de décrire ce que doit permettre de faire la solution : le quoi.

Récit utilisateur : (*user story* en anglais) exigence fonctionnelle racontée du point de vue l'utilisateur.

Format du type : en tant que ..., je veux ... , afin de ...

Exemple : *En tant que joueur du morpion
Je veux choisir mon pseudo
Afin de vivre une expérience utilisateur plus personnelle*

Remarque : Lors d'un projet professionnel, les *user stories* sont complétées par des spécifications détaillées. Au niveau d'un projet du lycée, on peut compléter les récits par des critères d'acceptation.

Exemple :

- *Le symbole du joueur apparaît sur tous les écrans de la partie.*
- *Le symbole est X ou O.*

Exigence extra-fonctionnelle : Exigence qui n'est pas fonctionnelle. Les exigences extra-fonctionnelles décrivent davantage comment doit être construite la solution

Exemples :

- exigences techniques (le logiciel doit être développé en Python version 3 ou supérieure) ;
- exigences qualité (le code doit être commenté, les noms de variables et fonctions doivent être explicite et avoir du sens, le code doit être testé de manière automatique, etc.) ;
- exigences de performance (toutes les pages du site doivent s'afficher en moins de 0,3 secondes...);
- contraintes juridiques (le logiciel doit se conformer au RGPD).

Exigence finissable : Exigence qui se termine définitivement après un effort réalisé par l'équipe. Les items de backlog sont nécessairement des exigences finissables.

Exemples :

- les récits utilisateurs décrivent des exigences fonctionnelles finissables ;
- une exigence de performance peut être une exigence finissable : la récupération des produits du catalogue doit être exécutée en moins de 0,01 seconde.

Exigence omniprésente : Exigence non finissable, qui doit être satisfaite de manière récurrente au cours du projet.

Exemples :

- la conformité à un niveau d'accessibilité : à chaque nouvelle exigence fonctionnelle développée, il faut vérifier que les interfaces sont accessibles aux personnes souffrant d'un handicap visuel ;
- les exigences de qualité du code sont des exigences omniprésentes : la couverture du code par les tests doit être satisfaite à chaque nouveau développement.

	Fonctionnelle	Extra-fonctionnelle
Finissable	<p>Exemple de récit utilisateur : <i>En tant que joueur du morpion</i> <i>Je veux choisir mon pseudo</i> <i>Afin de vivre une expérience utilisateur plus personnelle.</i> Une fois la fonctionnalité implantée, le travail est fini.</p>	<p>Exemple d'exigence technique : La requête de récupération des produits du catalogue doit s'effectuer en moins d'un centième de seconde</p> <p>Une fois la requête optimisée, le travail est fini.</p>
Omniprésente		<p>Exemple d'exigence de performance : Toutes les pages du site web doivent s'afficher en moins de 0,3 secondes lorsqu'on accède à la plateforme de production depuis les bureaux de l'entreprise.</p> <p>L'exigence est à satisfaire à chaque nouveau développement (le travail correspondant n'est donc jamais définitivement terminé tant qu'il reste des exigences fonctionnelles à satisfaire).</p>

Tableau 1 - Synthèse des types d'exigence

Le backlog ou backlog produit : liste des exigences finissables ordonnées par priorité.

NB : en Scrum le *backlog* ne contient que les exigences finissables qui peuvent être des récits utilisateurs, des exigences techniques, des études de faisabilité, etc. Dans le contexte de projet en lycée, nous préconisons de restreindre la liste aux récits utilisateurs.

Définition de fini :

Liste d'exigences non fonctionnelles, omniprésentes mais qui permettent de vérifier que le travail correspondant à la satisfaction d'un item de *backlog* est "vraiment" terminé pour chaque item de *backlog*. Pour chaque exigence dans le *backlog* , on dit qu'elle est terminée/finie lorsque la liste précédente est validée.

Exemple de définition de fini pour un projet de lycée :

- *les noms de variables et de fonctions sont explicites ;*
- *les critères d'acceptations sont respectés ;*
- *le code est commenté.*

Product owner :

Le responsable du *backlog*.

Il est responsable de la rédaction des récits utilisateurs et des critères d'acceptation.

En lycée : selon la nature du projet l'enseignant ou un élève dans un groupe peut occuper ce rôle.

III) Vocabulaire et étapes clés du développement

Cycle de développement :

Ensemble des phases et des activités menant à la réalisation du produit fini ou du service (résultat du projet).

Un cycle de développement agile est composé d'itérations (voir Figure 1).

Remarque : dans le cas d'un projet couvrant le développement de plusieurs versions majeures d'un logiciel, chaque version est obtenue à l'issue d'un cycle.

Itération

Période courte sur laquelle l'équipe se fixe des objectifs précis à atteindre à partir des exigences et tente de les réaliser. Dans le contexte des projets NSI, le travail est structuré par des séances en classe et prolongé hors du temps de classe. Plan de déroulement :

- En début d'itération : la planification en réunion d'équipe
 - Sélection des exigences fonctionnelles à implanter dans l'itération ;
 - "Découpage" de ces exigences en tâches à réaliser dans l'itération ;
- En cours d'itération :
 - Points étape à chaque séance de travail en classe ;
- En fin d'itération
 - Revue d'itération.

Dans Scrum, une itération est désignée par le terme "Sprint". Un sprint est de durée fixe variant de 1 à 4 semaines suivant le projet.

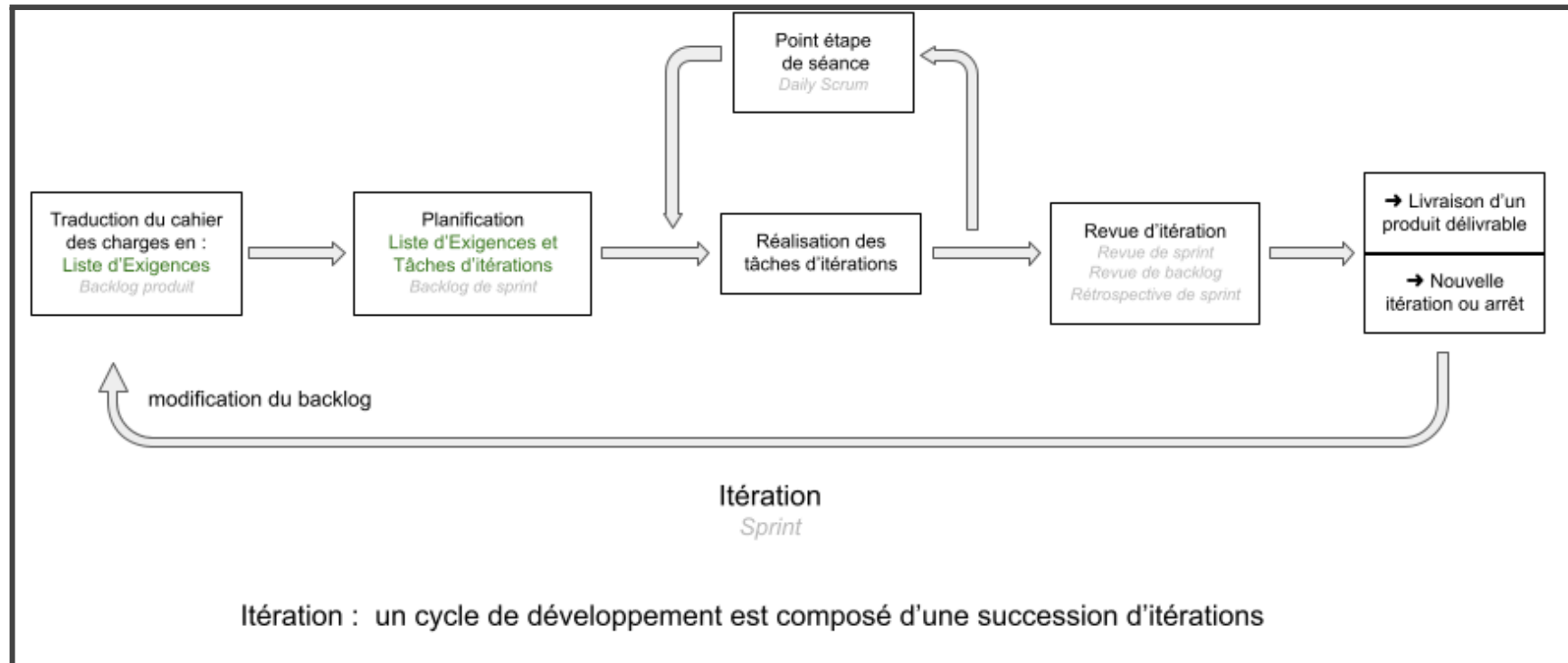


Figure 1 : Schéma d'une itération.
Le vocabulaire Scrum correspondant est en gris.

Exigences d'itération :

En résumé : Les exigences sélectionnées par l'équipe pour être implantées durant l'itération en cours.

Dans Scrum, les exigences d'itération composent le backlog de sprint.

En détail : durant l'étape de sélection des exigences d'itération, parmi les exigences les plus prioritaires, l'équipe sélectionne celles qu'elle pense être en capacité de traiter dans l'itération puis planifie les tâches qui permettront de les traiter. Elles doivent conduire à un produit potentiellement livrable (testable) à l'issue de cette itération.

Point étape de séance : Concertation rapide (5 min) à chaque début de séance de travail de groupe. Trois questions guident les échanges :

- Qu'ai-je fait depuis la dernière fois pour aider mon groupe à réaliser les tâches du cycle ?
- Que vais-je faire aujourd'hui pour aider mon groupe à accomplir les tâches du cycle ?
- Ai-je identifié des obstacles qui empêchent le groupe de réaliser les tâches du cycle ? Liste des obstacles à traiter.

Revue d'itération : à la fin d'une itération, deux éléments doivent être revus :

- **Revue du produit :**
Objectif pour l'équipe : améliorer le produit en collectant du *feedback* des parties prenantes.
Ce qui a été réalisé est-il fonctionnel ?
Y-a-t-il des adaptations à faire aux exigences ? Si oui, le *backlog* produit sera modifié en conséquence durant la revue de *backlog*.
- **Revue du backlog :**
Objectif pour l'équipe : ajuster le *backlog* suite aux retours émis lors de la revue du produit.
- **Revue de la méthode :**
Objectif pour l'équipe : améliorer son processus de développement en adoptant une posture réflexive.
Dans la manière de travailler, qu'est-ce qui a bien fonctionné (environ deux remarques) et qu'il faudrait consolider ?
Qu'est ce qui n'a pas fonctionné et qui mériterait d'être amélioré (environ deux remarques)?

Dans Scrum, la revue du produit s'appelle Revue de sprint, et la revue de la méthode s'appelle Rétrospective de sprint. La revue de Backlog, dans Scrum, peut se dérouler dans des réunions dédiées en cours de sprint pour préparer la réunion de planification du sprint suivant.

Produit délivrable : à l'issue d'un certain nombre d'itérations, le produit peut être mis à disposition de tous ses utilisateurs cibles. Le nombre d'itérations n'est pas défini à l'avance.

IV) Suggestions pour la mise en oeuvre de la méthode :

Pour permettre aux projets d'aboutir dans un délai raisonnable, nous suggérons :

- de réaliser les projets en un seul cycle de développement. On évitera ainsi la livraison de plusieurs versions successives d'un livrable ;
- de prévoir deux itérations par exemple pour un projet sur 4 séances.

Phase de planification :

- utiliser un tableau de type Kanban pour l'organisation tout au long du projet ; des outils tels que Trello peuvent être mobilisés pour réaliser un tableau de ce type ;
- l'enseignant pourra participer à la réunion de planification afin d'une part d'inciter les élèves à réfléchir en termes d'effort et/ou de temps à consacrer à chaque exigence, d'autre part valider la sélection des exigences de l'itération ainsi que le découpage de chaque exigence en tâches.

Pour les points d'étapes de début de séance :

- pour gagner du temps sur des itérations très courtes, utiliser un document partagé à compléter par les élèves avant chaque séance pour préparer l'échange du groupe avec éventuellement l'enseignant ;
- pour des itérations plus longues, commencer par les groupes ayant des points de blocage.

Pour la revue d'itération :

- Revue produit : prévoir un scénario de présentation du point de vue fonctionnel, du point de vue technique (partie de code détaillée) ;
- Revue méthode : exploiter la grille de notation finale à chaque étape (évaluation formative).