

Guide Scrum pour Débutants

1 Qu'est-ce que Scrum ?

Scrum est une méthode de travail en équipe qui permet de réaliser des projets **étape par étape**, en livrant régulièrement des résultats concrets.

Pensez-y comme :

- Construire une maison pièce par pièce, plutôt que d'attendre la fin pour voir le résultat
- Travailler en petites étapes (appelées "sprints") de 1 à 4 semaines
- Ajuster le plan en fonction des retours réguliers

Pourquoi utiliser Scrum ?

- Livrer plus rapidement de la valeur aux clients
 - Réduire les risques d'échec
 - S'adapter facilement aux changements
-

2 La Méthode Agile : Fondation de Scrum

Les 12 Principes de la Méthode Agile

Scrum est basé sur la méthode Agile. Voici ses 12 principes fondamentaux :

- 1. Satisfaction du client** Livrer rapidement et régulièrement des fonctionnalités qui apportent de la valeur.
- 2. Accepter les changements** Même tard dans le projet, les changements sont les bienvenus pour améliorer le produit.
- 3. Livraisons fréquentes** Livrer un produit fonctionnel toutes les 2 à 4 semaines (plutôt que d'attendre des mois).
- 4. Collaboration quotidienne** L'équipe de développement et les représentants du client travaillent ensemble tous les jours.
- 5. Motivation de l'équipe** Faire confiance à l'équipe, lui donner les moyens nécessaires et la laisser s'organiser.
- 6. Communication en face à face** La meilleure façon de transmettre l'information est la conversation directe.
- 7. Produit fonctionnel** Le progrès se mesure par un produit qui fonctionne, pas par des documents ou des plans.

8. Rythme soutenable L'équipe doit pouvoir maintenir son rythme de travail indéfiniment (pas de surcharge).

9. Excellence technique Porter attention constante à la qualité technique et à la conception.

10. Simplicité Maximiser le travail non fait : faire simple et se concentrer sur l'essentiel.

11. Auto-organisation Les meilleures solutions émergent d'équipes qui s'organisent elles-mêmes.

12. Amélioration continue À intervalles réguliers, l'équipe réfléchit à comment devenir plus efficace.



Les 3 Types de Méthodes Agiles

Scrum fait partie d'une famille de méthodes. Voici les 3 principales :

1 Scrum (La plus populaire)

C'est quoi ? Méthode structurée avec des rôles, événements et artefacts précis.

Idéal pour : Projets complexes nécessitant de la structure et des livraisons régulières.

Caractéristiques :

- Sprints de 1 à 4 semaines
- 3 rôles définis (PO, SM, Équipe)
- 5 réunions rituelles
- Adapté aux équipes de 3 à 9 personnes

Exemple : Développement d'une application mobile, refonte d'un site web.

2 Kanban (La plus flexible)

C'est quoi ? Méthode visuelle basée sur un tableau avec des colonnes (À faire, En cours, Terminé).

Idéal pour : Projets avec un flux continu de tâches, support client, maintenance.

Caractéristiques :

- Pas de sprints fixes
- Limite le nombre de tâches en cours (WIP limit)
- Flux continu de travail
- Très visuel et simple

Exemple : Support technique, gestion des bugs, opérations quotidiennes.

Différence avec Scrum : Pas de sprints, pas de rôles définis, plus flexible.

3 Extreme Programming (XP) (La plus technique)

C'est quoi ? Méthode axée sur les pratiques d'ingénierie et la qualité du code.

Idéal pour : Projets nécessitant une excellence technique et des changements fréquents.

Caractéristiques :

- Programmation en binôme (2 développeurs sur 1 ordinateur)
- Tests automatisés systématiques
- Intégration continue du code
- Refactoring (amélioration) constant du code

Exemple : Logiciels critiques, projets nécessitant une qualité maximale.

Différence avec Scrum : Plus technique, focus sur les pratiques de développement.

Comparaison Rapide

Critère	Scrum	Kanban	XP
Structure	Élevée	Flexible	Moyenne
Sprints	Oui (1-4 sem.)	Non	Oui (1-2 sem.)
Rôles définis	3 rôles	Aucun	2-3 rôles
Focus	Gestion projet	Flux de travail	Qualité technique
Changements	Entre sprints	N'importe quand	Fréquents
Idéal pour	Projets complexes	Tâches continues	Dev logiciel

3 Les 3 Piliers de l'Empirisme (Fondements de Scrum)

Scrum repose sur l'**empirisme** : une approche qui se base sur l'expérience et l'observation pour prendre des décisions.

Pilier 1 : Transparence

C'est quoi ? Rendre visible tout ce qui affecte le projet.

Concrètement :

- Tout le monde voit l'état du projet (progrès, problèmes)
- Les informations sont accessibles à tous les membres de l'équipe

- Les décisions sont prises sur la base de faits visibles
- Utilisation de tableaux, graphiques, réunions pour partager l'info

Exemple pratique :

- Tableau Kanban visible par tous
- Burndown chart mis à jour quotidiennement
- Product Backlog accessible et compréhensible

Pourquoi c'est important ? Sans transparence, impossible d'inspecter et d'adapter correctement.

Pilier 2 : Inspection

C'est quoi ? Examiner régulièrement le travail et les progrès pour détecter les problèmes.

Concrètement :

- Vérifier fréquemment si on va dans la bonne direction
- Évaluer la qualité du travail produit
- Comparer les résultats avec les objectifs
- Identifier les écarts et obstacles

Moments d'inspection dans Scrum :

- **Daily Scrum** : inspection quotidienne de l'avancement
- **Sprint Review** : inspection du produit livré
- **Sprint Retrospective** : inspection du processus de travail

Exemple pratique : Lors de la Sprint Review, on présente la fonctionnalité de paiement. Le client remarque que le processus est trop long → problème détecté !

Pourquoi c'est important ? Détecter les problèmes tôt permet de les corriger rapidement et à moindre coût.

Pilier 3 : Adaptation

C'est quoi ? Ajuster le plan, le processus ou le produit selon les observations.

Concrètement :

- Si quelque chose ne fonctionne pas, on change d'approche

- Modifier les priorités selon les retours clients
- Améliorer les méthodes de travail
- Accepter et intégrer le changement

Moments d'adaptation dans Scrum :

- **Daily Scrum** : ajuster le plan quotidien si nécessaire
- **Sprint Planning** : adapter le contenu du sprint selon les retours
- **Sprint Retrospective** : améliorer les pratiques de l'équipe

Exemple pratique : Suite à la Sprint Review, on décide de :

- Simplifier le processus de paiement (adaptation du produit)
- Ajouter plus de tests avant la démo (adaptation du processus)
- Reprioriser les fonctionnalités suivantes (adaptation du plan)

Pourquoi c'est important ? L'adaptation permet de rester flexible et de livrer ce qui apporte vraiment de la valeur.

Comment les 3 Piliers fonctionnent ensemble ?

TRANSPARENCE



(Tout est visible)



INSPECTION



(On observe et analyse)



ADAPTATION



(On ajuste et améliore)



[CYCLE SE RÉPÈTE]

Exemple de cycle complet :

1. **Transparence** : Le tableau montre que 3 tâches sont bloquées depuis 2 jours
2. **Inspection** : Lors du Daily Scrum, on identifie que l'API externe est inaccessible

3. **Adaptation** : On décide de travailler sur d'autres tâches et de contacter le fournisseur de l'API

⚠ **Important** : Sans ces 3 piliers, Scrum ne fonctionne pas correctement :

- Sans **transparence** → impossible d'inspecter
 - Sans **inspection** → impossible de savoir quoi adapter
 - Sans **adaptation** → on répète les mêmes erreurs
-
-

4 Les 3 rôles dans Scrum

Product Owner (Propriétaire du produit)

Son rôle : Décider QUOI faire

- Représente les clients et utilisateurs
- Définit les besoins et la vision du produit
- Priorise les tâches les plus importantes

Exemple : Pour une application de livraison, il décide qu'il faut d'abord créer la fonction de commande avant celle des avis clients.

Scrum Master (Facilitateur)

Son rôle : Aider l'équipe à bien travailler

- S'assure que les règles Scrum sont respectées
- Élimine les obstacles (problèmes techniques, manque de ressources)
- Facilite la communication entre tout le monde

Exemple : Si l'équipe attend une autorisation depuis 3 jours, le Scrum Master intervient pour débloquer la situation.

Équipe de développement

Son rôle : Faire le travail

- Auto-organisée : décide COMMENT faire les tâches
- Polyvalente : développeurs, designers, testeurs...
- Livre un produit fonctionnel à chaque sprint

Taille idéale : 3 à 9 personnes

5 Les 3 documents de travail (Artefacts)

Product Backlog (Liste de souhaits du produit)

C'est quoi ? La liste complète de tout ce qu'on veut faire sur le produit, classée par priorité.

Exemple : Pour un site e-commerce

1. Système de paiement (priorité haute)
2. Suivi de commande
3. Programme de fidélité
4. Chatbot d'assistance (priorité basse)

Sprint Backlog (Liste des tâches du sprint)

C'est quoi ? Les tâches sélectionnées pour le sprint en cours (1 à 4 semaines).

Exemple : Pour le sprint 1, on choisit uniquement "Système de paiement" et "Suivi de commande".

Increment (Résultat livrable)

C'est quoi ? Le produit fonctionnel obtenu à la fin du sprint, prêt à être utilisé.

Exemple : À la fin du sprint 1, les clients peuvent payer et suivre leurs commandes.

6 Les 5 réunions Scrum

Réunion	Objectif	Durée	Fréquence
Sprint Planning	Planifier les tâches du sprint	2-4h	Début de sprint
Daily Scrum	Point quotidien de 15 min	15 min	Chaque jour
Sprint Review	Présenter le travail fini	1-2h	Fin de sprint
Sprint Retrospective	Améliorer la méthode de travail	1-1.5h	Fin de sprint
Sprint	Période de travail	1-4 semaines	Continu

Sprint Planning (Planification)

Objectif : Décider ce qu'on va faire pendant le sprint.

Questions clés :

- Quelles tâches du Product Backlog choisir ?

- Comment va-t-on les réaliser ?

Daily Scrum (Réunion quotidienne)

Objectif : Se synchroniser rapidement chaque jour.

3 questions pour chaque membre :

1. Qu'ai-je fait hier ?
2. Que vais-je faire aujourd'hui ?
3. Ai-je des obstacles ?

Important : Debout, 15 minutes maximum !

Sprint Review (Revue)

Objectif : Montrer le travail accompli aux parties prenantes (clients, managers...).

On fait quoi ?

- Démonstration du produit fonctionnel
- Recueillir les retours
- Ajuster les priorités si nécessaire

Sprint Retrospective (Rétrospective)

Objectif : Améliorer la façon de travailler.

3 questions :

1. Qu'est-ce qui a bien fonctionné ?
2. Qu'est-ce qui a mal fonctionné ?
3. Comment s'améliorer pour le prochain sprint ?

Le cycle Scrum en 7 étapes

1. Préparer le Product Backlog



2. Sprint Planning (Planifier)



3. Sprint (Travailler pendant 1-4 semaines)



4. Daily Scrum (Point quotidien)

- ↓
5. Sprint Review (Présenter)
- ↓
6. Sprint Retrospective (S'améliorer)
- ↓
7. Nouveau sprint → Recommencer le cycle

Ce cycle se répète jusqu'à la fin du projet.

7 Outils pratiques

Logiciels de gestion

- **Jira** : très complet, idéal pour les équipes tech
- **Trello** : simple et visuel, parfait pour débiter
- **Monday** : flexible, s'adapte à tous types de projets
- **Asana** : bon compromis simplicité/fonctionnalités

Tableaux et graphiques

Kanban Board (Tableau Kanban) :

À faire			
En cours		Terminé	
Tâche A	Tâche C	Tâche D	
Tâche B		Tâche E	

Burndown Chart (Graphique d'avancement) : Un graphique qui montre le travail restant jour après jour dans le sprint.

8 Conseils pour réussir

À faire :

- Garder des sprints courts (2 semaines idéalement)
- Communiquer ouvertement sur les problèmes
- Livrer quelque chose de fonctionnel à chaque sprint

- Faire des rétrospectives honnêtes pour progresser
- Limiter les interruptions pendant le sprint

❌ À éviter :

- Changer les objectifs en cours de sprint
 - Sauter les réunions Daily Scrum
 - Négliger les rétrospectives
 - Travailler en silos (sans communiquer)
-

9 Avantages de Scrum

✨ **Flexibilité** : on peut changer de direction facilement 🚀 **Rapidité** : on livre rapidement de la valeur 📈
Amélioration continue : on s'améliore à chaque sprint 💪 **Motivation** : l'équipe est autonome et responsabilisée 🎯 **Moins de risques** : on teste régulièrement avec les utilisateurs

10 Limites et défis

⚠️ Attention à :

- Nécessite de la discipline et de la rigueur
 - Difficile pour les très grandes équipes (+ de 9 personnes)
 - Le Product Owner et Scrum Master doivent être compétents
 - Ne fonctionne que si les parties prenantes sont impliquées
-

1 1 Exemples concrets

📱 Exemple 1 : Application mobile de restaurant

Product Backlog :

1. Inscription/Connexion
2. Visualiser le menu
3. Commander en ligne
4. Paiement

5. Suivi de commande

6. Avis clients

Sprint 1 (2 semaines) : Inscription + Menu **Sprint 2 (2 semaines) :** Commander + Paiement **Sprint 3 (2 semaines) :** Suivi + Avis

Exemple 2 : Campagne marketing

Product Backlog :

1. Campagne email
2. Posts réseaux sociaux
3. Articles de blog
4. Analyse des résultats

Sprint 1 (1 semaine) : Campagne email + 1 article de blog **Sprint 2 (1 semaine) :** 5 posts réseaux sociaux

Sprint 3 (1 semaine) : Analyse des données + ajustements

Comment mesurer le progrès ?

Velocity (Vélocité)

C'est quoi ? Le nombre de tâches ou de points terminés par sprint.

Exemple :

- Sprint 1 : 20 points
- Sprint 2 : 25 points
- Sprint 3 : 23 points → Vitesse moyenne : 22-23 points par sprint

Burndown Chart

C'est quoi ? Un graphique qui montre le travail restant jour après jour.

Si la courbe descend bien, le sprint est sur la bonne voie !

Lead Time / Cycle Time


C'est quoi ? Le temps nécessaire pour finir une tâche du début à la fin.

Exemple : Une tâche prend en moyenne 3 jours → on peut prévoir nos sprints en conséquence.

- **Sprint** : Période de travail de 1 à 4 semaines
 - **User Story** : Description d'une fonctionnalité du point de vue de l'utilisateur
 - **Backlog** : Liste de tâches ou fonctionnalités à réaliser
 - **Increment** : Produit fonctionnel livré à la fin d'un sprint
 - **Velocity** : Vitesse de travail de l'équipe
 - **Burndown** : Graphique montrant le travail restant
-

Checklist pour démarrer avec Scrum

- ☐ Former une équipe de 3 à 9 personnes
 - ☐ Désigner un Product Owner
 - ☐ Désigner un Scrum Master
 - ☐ Créer le Product Backlog initial
 - ☐ Choisir un outil de gestion (Trello, Jira...)
 - ☐ Planifier le premier sprint (2 semaines recommandé)
 - ☐ Organiser le Daily Scrum quotidien à heure fixe
 - ☐ Prévoir la Sprint Review et Retrospective
-

 **Conseil final** : Scrum est simple dans ses principes mais demande de la pratique. Ne vous découragez pas lors des premiers sprints, l'équipe s'améliore à chaque itération !

Guide créé pour une impression facile - Format A4