

1 Définition par le Scrum Guide

Les Developers sont les membres de la Scrum Team qui s'engagent à traiter tout ou partie utile d'un Increment à chaque Sprint. Les compétences spécifiques requises pour les Developers sont souvent larges et varient selon le domaine d'activité. Toutefois, les Developers sont toujours redevables de :

- Créer un plan de Sprint, un Sprint Backlog ;
- Inculquer la notion de qualité en adhérant à une *Definition of Done* ;
- Adapter leur plan chaque jour par rapport au *Sprint Goal*
- Se tenir mutuellement responsables en tant que professionnels.

2 Développeurs ou codeurs

Affirmation 1. *Dans la version 2020 du Scrum Guide on ne parle plus d'équipe de développement, mais de développeurs.*

Équipe de développement : ensemble de codeurs qui exécutent (des exécutants) des tâches fournies.

Développeurs : est une personne qui travaille pour réaliser un *Incrément* : codeur, testeur, analyste, ...

2.1 Quelle influence ?

Affirmation 2. *Scrum ne reconnaît aucune sous-équipe. Tous le monde travaille dans une seule et même équipe.*

Certaines personnes considéraient *l'équipe de développement* comme une sous-équipe. La nouvelle formulation renforce le fait qu'il existe une seule équipe Scrum qui contient des personnes capables de remplir les responsabilités (Scrum Master, Product Owner et Développeurs).

Avec Le terme de *développeur* on englobe toutes les personnes de l'équipe Scrum qui s'engagent à créer un Incrément utilisable à chaque Sprint. Cela inclut les développeurs, les testeurs et tous les autres spécialistes nécessaires pour s'assurer de la bonne réalisation de l'Incrément.

2.2 Travailler avec des gens

Néanmoins les personnalités de chacun doivent être prise en compte pour la bonne réalisation du projet. Malgré le fait que le Scrum Guide nous dit de *travailler avec des personnes motivées* ceci peut être compliqué à appliquer en entreprise. En effet, au sein d'un groupe de travail des personnes seront *passives* et d'autres *actives*.

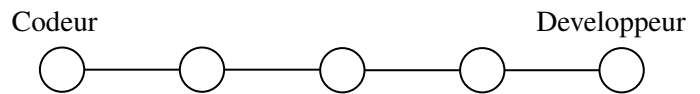
Note : que la personne doit passive ou active il n'y a aucun jugement de valeur à porter.

2.3 Le danger

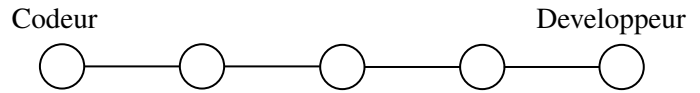
Affirmation 3. *Attendre des autres ce qu'ils ne sont pas prêts à donner.*

En tant que chef de projet ou facilitateur Agile (e.g. Scrum Master) il est important de savoir comment les personnes se sentent et veulent se sentir au sein du projet.

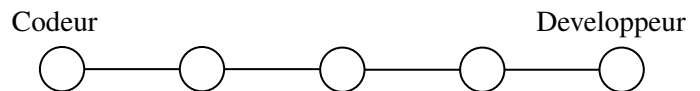
Comment pensez vous être considéré par votre entreprise en ce moment ?



Comment vous considérez-vous en ce moment ?



Comment souhaitez-vous être considéré ?



Ces questions (et d'autres) permettront de comprendre le "mood" de chacun et ce qu'il souhaite ou non *être* pour l'entreprise.

1. Chaque personne explique son "mood" intellectuel
2. Regarder l'écart entre les personnes
⇒ avis différents et confronter les points de vue
3. Tendance globale de l'équipe
⇒ adapter vers la vision où on souhaite aller
4. Plan d'action

⇒ On rend le cadre clair, être au courant de qui est qui.

2.4 Règles de vie (working agreement)

Cela permettra également de définir les règles de l'équipe. Comme la partie précédente, ceci se fait avec la Scrum Team au complet.

Affirmation 4. *Il peut être désagréable que certaines personnes choisissent et d'autres subissent.*

- À quelle heure le daily se fait ?
- Qui va dans quel bureau ?
- Qui anime les réunions ?
- Quelles sont les disponibilités de chacun ?
- ...

Définition. *Privilégier les discussions et des compromis à un vote à la majorité.*

L'objectif est de définir un cadre clair et explicite où tout le monde est en accord avec les règles. À noter que les règles peuvent évoluer (Sprint review).

2.5 Aménagement des ACTIFS

Les personnes actives sont celles qui sont le plus impliquées dans la vie du logiciel. Il est important de leur accorder du temps pour des tâches annexes pour la survie du logiciel :

- Veille technologique
- Discussion inter-équipe dans l'entreprise
- Évènements et forums (e.g. Devvxx)

2.6 Dette de motivation

Definition. *Developer Experience (DX) décrit les interactions et les sentiments qu'éprouve un développeur lorsqu'il travaille avec un corps de code afin d'atteindre un objectif spécifique.*

- Un ordinateur adéquat
- Une gestion de projet bien organisée
- Code legacy
- ...

Quand l'environnement n'est pas à la hauteur, on accumule une *dette de motivation*. Au quotidien les développeurs prennent sur eux afin de faire évoluer l'application dans le bon sens.

1. Identifier la dette et mettre sur la table les problèmes
2. Trouver avec eux (choix technologique) pour réduire la dette

⇒ Amélioration de la confiance entre les managers et l'équipe

<https://www.redhat.com/architect/developer-experience>