

Dans le contexte actuel d'une entreprise de développement logiciel avec des projets variés et des équipes dédiées, l'efficacité opérationnelle est cruciale pour garantir la livraison de produits de haute qualité dans les délais impartis. Cependant, la nature fluctuante du flux de travail, caractérisée par des tâches en attente et des blocages fréquents, entrave l'efficacité et la productivité des équipes. C'est dans ce cadre que l'adoption de la méthode Kanban s'impose comme la solution la plus adaptée pour améliorer le processus et maximiser l'efficacité de l'entreprise de développement logiciel.

### **Pourquoi Kanban est la Solution Adaptée :**

1. **Visualisation du Flux de Travail :** La première étape cruciale dans l'amélioration de tout processus est la visualisation claire du flux de travail. Kanban offre un tableau visuel où chaque tâche est représentée par une carte, permettant à chaque membre de l'équipe de voir instantanément ce qui est en cours, ce qui est terminé et ce qui est à venir. Cette transparence accrue élimine les incertitudes et permet une gestion proactive des ressources.
2. **Limitation du Travail en Cours (WIP Limit) :** Kanban propose le concept de limiter le travail en cours. En établissant des limites sur le nombre de tâches autorisées à chaque étape du processus, Kanban prévient la surcharge de travail. Cela permet aux équipes de se concentrer sur un nombre limité de tâches, améliorant ainsi la qualité et la rapidité de leur achèvement.
3. **Réduction des Tâches en Attente et des Blocages :** Grâce à la visualisation en temps réel du flux de travail et à la limitation du travail en cours, les tâches en attente et les blocages sont immédiatement repérés. Les équipes peuvent réagir rapidement pour résoudre ces problèmes, évitant ainsi des retards inutiles et assurant un flux continu des tâches à travers le processus.
4. **Amélioration Continue grâce aux Rétrospectives :** Kanban encourage les équipes à organiser des réunions régulières de rétrospectives. Ces sessions permettent aux membres de l'équipe de réfléchir sur leur performance passée, d'identifier les obstacles rencontrés et de proposer des améliorations. Cette boucle de rétroaction constante favorise l'amélioration continue du processus.
5. **Adaptabilité aux Changements :** Dans le domaine du développement logiciel, les exigences peuvent changer rapidement. Kanban offre une grande flexibilité en permettant aux équipes d'ajuster leur flux de travail en fonction des nouvelles priorités. Cela assure que l'entreprise peut répondre rapidement aux demandes changeantes du marché sans compromettre la qualité.

6. **Focus sur la Livraison Rapide** : En réduisant les temps d'attente, les blocages et les retards, Kanban permet une livraison plus rapide des fonctionnalités et des produits. Cela peut améliorer la satisfaction du client et renforcer la réputation de l'entreprise sur le marché.

7. **Encouragement de l'Engagement de l'Équipe** : En impliquant activement les membres de l'équipe dans la gestion de leur propre flux de travail et en leur donnant la possibilité de contribuer aux améliorations, Kanban favorise un engagement accru. Les membres de l'équipe se sentent responsables du succès du processus, ce qui peut améliorer la motivation et la productivité.

### *Conclusion*

Dans le contexte de votre entreprise de développement logiciel, où la gestion des tâches en attente et des blocages entrave l'efficacité, l'adoption de la méthode Kanban offre une solution holistique. Grâce à sa visualisation claire du flux de travail, sa limitation du travail en cours, sa réduction des tâches en attente, son adaptabilité aux changements, son focus sur la livraison rapide, et son encouragement de l'engagement de l'équipe, Kanban est la clé pour améliorer le flux de travail, augmenter la productivité et garantir la satisfaction du client. Cette approche méthodique et orientée vers l'amélioration continue constitue la réponse idéale à vos défis opérationnels actuels, offrant ainsi à votre entreprise un avantage concurrentiel durable sur le marché du développement logiciel.

## **2. Principes et Concepts Clés de Kanban :**

### **LES PRINCIPES DU CADRE KANBAN :**

Le cadre repose sur quatre principes fondateurs que nous avons découverts pas à pas avec l'équipe de Didier et l'accompagnement de Johanna. Rappelons-les de façon plus synthétique.

#### ***Commencer là où on en est :***

Le premier principe propose de commencer là où on en est, qu'il s'agisse du processus ou de sa mise en œuvre. Le cadre ne se suffit pas à lui-même et s'applique au processus existant. Trop de changements rapides provoque du stress et de la résistance. Ne pas changer l'existant au démarrage d'une démarche d'amélioration est moins perturbant pour les membres de l'équipe. En contrepartie, la démarche peut être moins porteuse d'innovation immédiate.

#### ***Respecter le processus actuel, les rôles et les responsabilités :***

Le deuxième principe permet de construire le changement dans le respect de l'organisation, notamment des rôles et responsabilités, et des personnes en tant qu'individus. S'ils ne changent pas initialement, ils vont naturellement évoluer après la mise en place d'un système kanban.

### ***S'engager à changer de manière incrémentale et évolutive :***

Le point de vue du kanban est une évolution par petits pas plutôt qu'un changement radical, même si les deux peuvent être compatibles. Cette évolution va porter sur le système de manière incrémentale. À chaque étape de son évolution, le système kanban sera opérationnel : les équipes pourront transformer des demandes en solutions utilisables par des utilisateurs. L'objectif est de développer une culture de l'amélioration continue et d'expérimentation, avec le droit à l'erreur. Kanban fait évoluer un existant : on n'évolue pas vers du kanban mais avec le kanban.

### ***3.2.4 Avec des actes de leadership à tous les niveaux :***

Cette évolution du système va être menée par des actes de leadership ! Pour engager l'organisation, le management et les équipes dans cette transformation, il faut encourager des actes de leadership à tous les niveaux, au travers de contributions individuelles ou collectives. Ces contributions peuvent aller du simple mode de visualisation du travail défini par l'équipe, à une nouvelle règle de priorisation stratégique définie par un décideur ; ou encore, de la part des managers, aller voir sur le terrain et engager des discussions avec l'équipe autour de leur management visuel. Cette notion de leadership est développée plus tard dans ce livre.

### **Les concepts :**

#### **a. Limitation du Travail en Cours (WIP Limit) :**

- **Explication** : La limitation du Travail en Cours détermine le nombre maximum de tâches autorisées à chaque étape du processus. Cela assure que l'équipe ne soit pas submergée, permettant une concentration optimale et un travail de qualité.
- **Exemple** : Si le WIP Limit est de cinq et qu'il y a déjà cinq tâches en cours, aucune nouvelle tâche ne peut être commencée avant qu'une des tâches en cours ne soit terminée.
- **Bénéfice** : En empêchant la surcharge de travail, les membres de l'équipe peuvent se concentrer pleinement sur chaque tâche, améliorant ainsi la qualité du travail et la rapidité de l'achèvement.

#### **b. Tableau Kanban :**

- **Explication** : Le tableau Kanban est un outil visuel divisé en colonnes représentant différentes étapes du flux de travail. Les cartes de tâches se déplacent à travers ces colonnes, indiquant leur progression dans le processus.
- **Exemple** : Une carte de tâche se déplace de la colonne "À Faire" à "En Cours" lorsque le travail commence sur cette tâche, offrant une visualisation en temps réel du travail en cours.
- **Bénéfice** : La visualisation claire du flux de travail élimine les ambiguïtés, facilite la communication et permet à l'équipe de suivre facilement le progrès des tâches.

### c. Colonnes du Tableau Kanban :

- **À Faire** : Tâches à venir, en attente d'être traitées.
- **En Cours** : Tâches actuellement en traitement par l'équipe.
- **En Attente** : Tâches en pause en raison de dépendances externes ou d'attente de validation.
- **Terminé** : Tâches terminées et prêtes à être livrées.
- **Exemple** : Une tâche est déplacée de "En Cours" à "Terminé" une fois que le travail est terminé et qu'elle est prête à être livrée au client ou à la phase suivante du projet.
- **Bénéfice** : La structuration des colonnes assure une progression ordonnée des tâches, évitant les confusions et les retards, et permettant un suivi précis du statut de chaque tâche.

### 3. Indicateurs Comme "Lead Time" :

#### a. Lead Time :

- **Explication** : Le Lead Time mesure la durée totale nécessaire pour terminer une tâche du début à la fin. Cela inclut toutes les étapes du processus, de la création à la livraison.
- **Exemple** : Le Lead Time pour développer une fonctionnalité particulière est de 10 jours. Cela signifie qu'il faut en moyenne 10 jours, du moment où la tâche est créée jusqu'à ce qu'elle soit complètement terminée et livrée.
- **Bénéfice** : Le suivi du Lead Time permet de comprendre les délais réels de chaque tâche, identifiant ainsi les domaines à améliorer pour accélérer le processus global, assurant ainsi une satisfaction client plus rapide.