KAMANGA

GUIDE GRATUIT · 10 PAGES · FRAMEWORK KAMANGA

Réduire votre lead time de 50% en 90 jours.

Le framework en 4 phases que j'ai appliqué dans 12+ équipes engineering. Avec les métriques DORA à suivre, les cas réels chiffrés, et les 5 pièges qui font échouer 80% des transformations.

Kamanga · 25 ans en engineering leadership

kamanga.fr · Édition 2026

KAMANGA

Avant de commencer

Le lead time, vraiment, et pourquoi c'est l'indicateur clé

Le **lead time** est le temps entre le moment où une idée est validée et le moment où elle est en production, utilisée par de vrais utilisateurs.

Ce n'est ni le temps de développement estimé, ni la sprint velocity, ni le temps de build du CI. C'est la durée **totale** du flux, de l'idée au déploiement.

> *« Un lead time long n'est pas un problème de talent ou de vitesse des devs. C'est le symptôme de 3 ou 4 blocages précis qu'on peut nommer, mesurer, et éliminer. »*

Pourquoi c'est l'indicateur qui change tout

- Feedback utilisateur rapide, donc décisions business meilleures
- Moins de risque par release, donc plus de releases, donc moins de stress
- Équipe qui voit l'impact de son travail, donc motivation qui tient
- Capacité à répondre aux opportunités marché en jours, pas en trimestres

Le framework en 4 phases, sur 13 semaines

Discipline phases 1 et 2, aucune migration technique **Outillage** phases 3 et 4, automatisation

PHASE 1
Mesurer
Calculer le lead time réel et baseliner les métriques DORA.

›

PHASE 2
Débloquer
Réduire les lots, fluidifier la review, limiter le WIP.

›

PHASE 3
Stabiliser
Pipeline CI rapide, tests utiles, Definition of Done.

›

PHASE 4
Accélérer
Trunk-based, feature flags, déploiement en confiance.

La largeur de chaque bloc reflète sa durée réelle : la phase 3 (stabiliser) est la plus longue, c'est là que les gains s'ancrent.

À GARDER EN TÊTE

Le framework est **séquentiel**. Sauter une phase pour aller plus vite est le piège n°2 (voir page 8). Les phases 1 et 2 ne demandent aucune migration technique, juste de la discipline.

Phase 1, Mesurer

Semaines 1 et 2, calculer le réel avant de toucher au système

PHASE 01 · DIAGNOSTIC

On ne peut pas améliorer ce qu'on ne mesure pas.

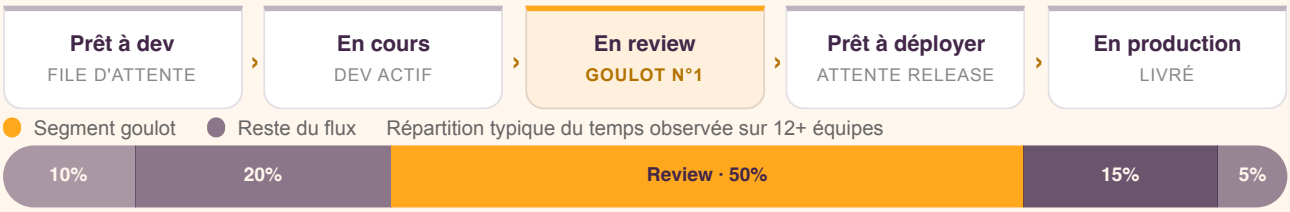
1.1 Calculer votre lead time réel

Prenez les 20 dernières features livrées en production. Pour chacune, calculez la durée entre *prête à développer* et *en production*. Faites la **médiane**, pas la moyenne (les outliers faussent tout).

Ce que vous allez découvrir : votre lead time est probablement **30 à 50% plus long** que ce que vous estimiez. Et 2 ou 3 goulots représentent 70% du temps.

1.2 Décomposer le lead time

Découpez le flux en segments mesurables. Chaque flèche est un point de passage où le temps peut s'accumuler :



Dans 4 cas sur 5, **la review est le segment le plus long** (40 à 60% du total). Et *prêt à déployer* révèle des problèmes de release management qu'on ne soupçonnait pas.

1.3 Les 4 métriques DORA à baseline

MÉTRIQUE	VOTRE BASELINE	OBJECTIF 90 JOURS
Lead time for changes	*à mesurer*	**−50%**
Deployment frequency	*à mesurer*	**×3 À ×5**
Change failure rate	*à mesurer*	**< 15%**
Time to restore service	*à mesurer*	**< 1 JOUR**

LE PIÈGE CLASSIQUE DE LA PHASE 1

Vouloir un outil de mesure parfait avant de commencer. Un tableur partagé suffit pour 4 semaines. L'outillage viendra en Phase 3.

PHASE 02 · FRICTION

Supprimer les frictions majeures dans le flux.

2.1 Réduire la taille des lots

C'est souvent **la seule action qui divise le lead time par 2 en 3 semaines**. Règle pratique : une PR ne dépasse pas 400 lignes de code (hors tests générés).

- Définir une "PR size rule" en équipe (400 LOC max, 1 concern max)
- Feature slicing : découper les features en tranches verticales livrables
- Walking skeleton : livrer la structure avant le contenu

2.2 Fluidifier la code review

La review est le goulot principal dans 4 équipes sur 5. Deux niveaux d'action :

TACTIQUE

Immédiat, en équipe

SLA review 4h max sur la 1ère review. PR template (3 lignes + checklist). Pair review pour les PRs complexes (30 min ensemble > 2h en solo).

STRUCTUREL

Sous 3 semaines

Definition of Ready avant soumission. Rotation des reviewers. Review buddy junior + senior pour faire monter l'équipe.

2.3 Limiter le WIP

Règle simple : **maximum 2 tâches en cours simultanément par développeur**. Finir > commencer. Toujours.

> **LEVIER LE PLUS IMPACTANT DE LA PHASE 2**
>
> La PR size rule, semaine 3. Simple à décider, rapide à mettre en place, résultats mesurables en 2 semaines. Si vous ne devez faire qu'une chose, faites celle-là.

PHASE 03 · AUTOMATISATION

Construire les fondations pour que les gains durent.

3.1 Pipeline CI sous les 10 minutes

Un pipeline lent décourage les commits fréquents, et tue les gains de la Phase 2. Objectif : moins de 10 minutes end-to-end.

- Paralléliser les jobs de test
- Utiliser le cache agressivement (dépendances, images Docker)
- Identifier et isoler les tests lents (> 2s, candidats à l'optimisation)
- Splitter les suites de tests par module

3.2 Couverture de tests suffisante

Pas d'objectif absolu : les zones à risque doivent être couvertes, le reste peut attendre.

- Tests d'intégration sur les chemins critiques (> tests unitaires partout)
- Contract tests pour les APIs et les intégrations
- Tests de smoke en production post-déploiement

3.3 Definition of Done enrichie

VOTRE DOD DOIT INCLURE

- ☐ Tests écrits et passants
- ☐ Pipeline CI vert
- ☐ Code reviewé par au moins 1 personne
- ☐ Documentation à jour (si applicable)
- ☐ Feature flag en place (si feature risquée)
- ☐ Rollback possible en moins de 5 minutes

> *« En Phase 3, on cesse d'être héroïque. Si le système ne tient pas sans qu'une personne précise soit là un vendredi soir, c'est qu'il n'est pas encore stabilisé. »*

Phase 4, Accélérer

Semaines 11 à 13, aller vraiment vite une fois les bases solides

PHASE 04 · VÉLOCITÉ

Une fois les fondations solides, on peut vraiment voler.

4.1 Trunk-Based Development

Au lieu de branches feature longues, des commits fréquents sur main avec feature flags. Effet observé : **réduction du lead time de 40 à 70%** dans les équipes qui l'adoptent.

> **TRANSITION DOUCE SUR 3 SEMAINES**
> Semaine 1, branches max 2 jours. Semaine 2, branches max 1 jour. Semaine 3, trunk-based complet.

4.2 Feature Flags

Déployer sans livrer. Cela découple le déploiement technique de la mise en production business.

- **Dark launch** : déployer sans activer pour les utilisateurs
- **A/B testing** : activer pour X% des utilisateurs
- **Kill switch** : désactiver instantanément si problème en prod

Outils recommandés : LaunchDarkly, Unleash (self-hosted), ou simple table en base.

4.3 Déploiement en confiance

Objectif : **chaque développeur peut déployer en production à tout moment, seul**.

PIPELINE

Zéro action manuelle

Du merge au prod, tout est automatisé. Monitoring et alertes configurés en amont.

ROLLBACK

Documenté et testé

Runbook de rollback < 5 minutes. Testé en exercice, pas seulement écrit.

> **LE SIGNAL QUE LA PHASE 4 A RÉUSSI**
> Vos développeurs déploient le vendredi après-midi sans angoisse. Pas de freeze, pas de "on attend lundi".

KAMANGA

3 cas réels, avant et après

Ce que ça donne quand le framework est appliqué jusqu'au bout

Services financiers, DSI grande banque

Équipe engineering de 80+ développeurs, déploiement sous coordination multi-équipes.

AVANT	Lead time moyen **35+ jours**, déploiement mensuel sous coordination multi-équipes.
APRÈS 90J	Lead time **réduit de moitié**, fréquence de déploiement multipliée par 4.
LEVIER CLÉ	Suppression des dépendances de livraison inter-équipes + Definition of Done partagée.

Groupe média & broadcast, produit digital

Équipe de 40+ développeurs, taux d'incident post-release élevé.

AVANT	Lead time **30+ jours**, taux d'incident post-release élevé, rollbacks fréquents.
APRÈS 90J	Lead time **divisé par 3**, taux d'incident réduit significativement.
LEVIER CLÉ	Trunk-based development + feature flags + pipeline CI/CD automatisé.

Organisme de protection sociale, core platform

Équipe de 25 développeurs, processus de validation lourd.

AVANT	Lead time **20+ jours**, 1 release/mois sous processus de validation lourd.
APRÈS 90J	Cycle raccourci à **moins de 8 jours**, processus de validation allégé et outillé.
LEVIER CLÉ	Réduction de la taille des lots + automatisation des tests de non-régression.

« Dans les trois cas, le vrai goulot n'était pas là où l'équipe pensait. C'est ce que révèle systématiquement la Phase 1. La transformation commence toujours par cette surprise. »

KAMANGA

Les 5 pièges qui font échouer 80% des transformations

Si vous évitez ceux-là, vous êtes déjà devant la plupart

PIÈGE 1

Attaquer l'outil avant le flux

Adopter un nouvel outil (Jira, Copilot, nouveau CI) sans d'abord comprendre et optimiser le flux humain. Résultat : le même problème avec un outil plus cher.

ANTIDOTE Phases 1 et 2 avant tout outillage.

PIÈGE 2

Changer tout en même temps

Trunk-based + refonte CI + nouvelles pratiques de test + formation IA, le tout en un sprint. L'équipe est déstabilisée, rien ne tient.

ANTIDOTE Le framework est séquentiel. Respectez l'ordre des phases.

PIÈGE 3

Mesurer la vélocité plutôt que le flux

Les story points mesurent l'effort, pas la valeur livrée. Une équipe qui finit 40 points par sprint avec un lead time de 6 semaines ne va pas bien.

ANTIDOTE Basculer sur les métriques DORA comme indicateurs principaux.

PIÈGE 4

Ignorer la dette technique

On veut aller vite, alors on saute le refactoring. La dette s'accumule. Le lead time remonte. On revient à la case départ 6 mois plus tard.

ANTIDOTE Allouer 20% du temps d'équipe au refactoring, non négociable.

PIÈGE 5

Le management non aligné

L'équipe tech progresse, mais le PO arrive avec des stories de 50 points, le management veut des estimations en semaines, et les releases restent gérées à la main.

ANTIDOTE Aligner le management dès la Phase 1. Présenter les métriques DORA en comité de direction.

KAMANGA

Template, Lead Time Review

Rétro mensuelle de 45 minutes, équipe tech + PO

L'instrument qui garde la transformation vivante au-delà des 90 jours. Une fois par mois, même rituel, même structure. Toujours un seul chantier d'amélioration à la sortie, jamais une liste.

ÉTAPE 1
10 minutes

Métriques du mois

- Lead time médian : ___ j (vs mois précédent : ___ j)
- Deployment frequency : ___
- Change failure rate : ___ %
- Temps de review médian : ___ h

ÉTAPE 2
15 minutes

Les 3 plus longues features du mois

Pour chacune : où s'est passé le temps ? Quel était le goulot ? Pas de blâme, juste un diagnostic de système.

ÉTAPE 3
5 minutes

Ce qui a marché

Une chose que l'équipe a bien faite et qu'on garde explicitement.

ÉTAPE 4
10 minutes

Une chose à améliorer

Pas une liste, une seule. Avec un owner nommé et une date d'échéance.

ÉTAPE 5
5 minutes

L'objectif du mois prochain

Un chiffre, un seul. Exemple : « on vise un lead time médian de 12 jours ».

> **RÈGLE D'OR DE LA REVIEW**
>
> Ne jamais transformer la review en planning. Si une décision prend plus de 5 minutes, on la sort et on la traite ailleurs. La review mesure, ne planifie pas.

90 jours.
C'est le temps qu'il faut.

Le guide donne les grandes lignes. La mise en œuvre dépend de votre contexte, de votre dette technique, de votre stack et de votre équipe. Si vous voulez construire un plan d'action sur mesure, voici les portes d'entrée.

QUATRE FAÇONS D'AVANCER AVEC KAMANGA

Engineering Maturity Assessment →

Auto-évaluation gratuite de la maturité engineering de votre équipe. 12 axes, restitution écrite, 20 minutes.

app.kamanga.fr/forms/ema

AI-Ready Engineering Checklist →

25 questions, 5 dimensions (Tests, CI/CD, Gouvernance IA, Pratiques d'équipe, Dette technique). Score sur 25 et plan d'action 30/60/90 jours. Gratuit, 7 minutes.

app.kamanga.fr/forms/ai-checklist

Programme Craft & Velocity →

Accompagnement équipe sur 90 jours. Le framework de ce guide, déployé dans votre contexte, avec mesure et coaching.

kamanga.fr/craft-velocity

Session découverte (30 min) →

Échange gratuit sans engagement. On regarde votre situation, on identifie le levier prioritaire.

app.kamanga.fr/forms/discovery-call

> *« Un lead time de 6 semaines n'est pas un problème d'engineering, c'est un problème de rentabilité. »*

POUR RESTER DANS LA BOUCLE

kamanga.fr LinkedIn @kamangacode herve@kamanga.fr