



GUIDE PRATIQUE

Comment **former** son équipe dev sans casser le sprint

24 fiches pratiques pour monter en compétences sans sacrifier la velocity

00 Introduction

Pourquoi la formation est un investissement, pas un coût. Le piège du "pas le temps".

01 Diagnostiquer les besoins de l'équipe

Carte de compétences, signaux faibles, entretiens individuels

02 Formats courts : le micro-learning dans le sprint

Lightning talks, katas de code

03 La revue de code comme outil de formation

Revue pédagogique, rotation des reviewers

04 Le pair programming stratégique

Sessions ciblées, binômes senior/junior, mob programming

05 Les rituels d'équipe formateurs

Post-mortem, tech radar, rétro orientée compétences

06 Former sans sortir du backlog

Tickets stretch, refactoring guide, spike techniques

07 Onboarder sans ralentir l'équipe

Parcours structuré, premier ticket, buddy system

08 Mesurer la progression sans bureaucratie

Indicateurs et grille de progression, feedback continu

09 Convaincre le management et le PO

Argumentaire chiffré, ROI de la formation, négociation du temps

10 Construire une culture d'apprentissage durable

Ancrer les habitudes, éviter les rechutes, responsabiliser l'équipe

-- Conclusion et annexes

Tableau récapitulatif, modèle de plan de formation, checklist du lead

Le piège du "pas le temps de former"

Une équipe qui ne monte pas en compétences finit par ralentir. La formation n'est pas un luxe, c'est de la maintenance préventive.

Le sprint est plein. Le backlog déborde. Le PO attend ses features. Dans ce contexte, proposer de "prendre du temps pour se former" ressemble à une provocation. Pourtant, les équipes les plus performantes investissent systématiquement dans la montée en compétences. Pas en bloquant des semaines entières. En intégrant la formation dans le flux de travail quotidien.

Ce livret regroupe **24 fiches pratiques** pour former votre équipe sans interrompre la production. Chaque fiche décrit une situation, une méthode et des conseils de mise en œuvre. L'objectif : que la formation devienne invisible, intégrée dans le sprint au lieu de le concurrencer.

Pourquoi les équipes stagnent

- **Dette technique silencieuse** : les développeurs contournent ce qu'ils ne maîtrisent pas au lieu de l'apprendre. Le code se dégrade
- **Bus factor** : un seul expert porte un pan entier du système. Son absence paralyse l'équipe
- **Turnover** : les développeurs qui ne progressent plus partent. Le recrutement coûte plus cher que la formation
- **Velocity en trompe-l'oeil** : livrer vite du code fragile génère du retravail. La velocity réelle chute, pas celle affichée

94%

des employés resteraient plus longtemps si leur entreprise investissait dans leur développement

LinkedIn Workplace Learning Report

2x

le salaire annuel, c'est ce que coûte le remplacement d'un collaborateur

Gallup, 2025

22%

seulement des devs ont du temps dédié à l'upskilling chez leur employeur

HackerRank Developer Skills Report, 2025

PRINCIPE DIRECTEUR

La formation la plus efficace est celle que l'équipe ne perçoit pas comme de la formation. Elle se confond avec le travail quotidien : revue de code, pair programming, tickets bien choisis.