



Do Tank « Impacts de l'IA sur les activités d'ingénierie logicielle »

**GT 1/6**

**« Développeur augmenté  
& Passage à l'échelle »**

# Les sociétés ayant participé au GT



Pilote du GT :  
Yannick FERNBACH

amADEUS



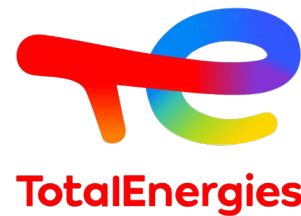
bpi**france**



hagergroup



NAVAL  
GROUP



- 1. Introduction**
- 2. Mise en place de la conduite du changement**
- 3. Pilotage et bonnes pratiques**
- 4. Impacts sur l'évolution des compétences**
- 5. Conclusions et perspectives**

# 01

# Introduction

# Introduction : contexte du Groupe de Travail

L'essor des outils d'IA générative utilisés dans le développement logiciel fait émerger une nouvelle figure : celle du **développeur augmenté**.

Les pratiques des développeurs évoluent, la manière d'écrire du code est redéfinie. Ces nouvelles façons de faire améliorent la productivité. Cependant, le véritable défi commence maintenant avec le **passage à l'échelle de l'entreprise** de **ces façons de procéder**.

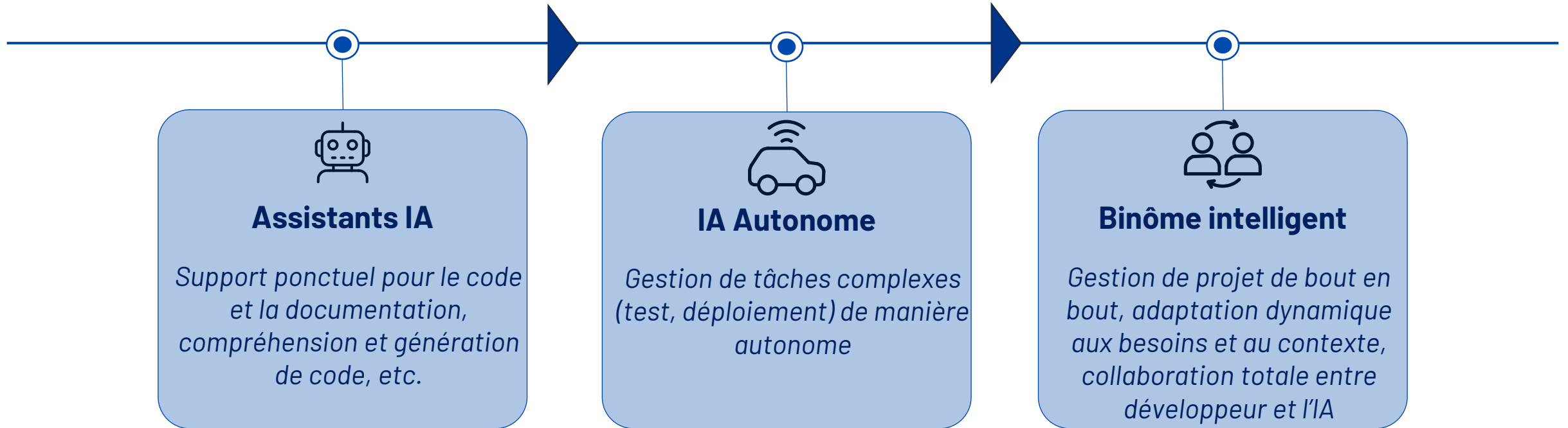
Car généraliser ce modèle ne relève pas d'un simple déploiement d'outils. Cette transition se heurte à des obstacles structurels et soulève des questions encore largement ouvertes :

- Quels **gains réels** peut-on en attendre au regard du coût, et surtout, comment les mesurer objectivement ?
- Quelles **évolutions des pratiques et des compétences** des équipes cela implique-t-il ?
- Comment **accompagner ce changement** pour garantir son adoption ?

Il ne s'agit plus seulement d'outiller des individus, mais de repenser le **modèle de production logicielle** à l'ère des IA génératives et agentiques.

# Évolution vers le développeur augmenté : de l'assistant IA au binôme autonome

Trajectoire de maturité de l'IA générative :



L'enjeu n'est pas **seulement le code**, mais la capacité à **gérer l'ensemble du cycle de développement**



Ces réflexions trouvent un écho particulier dans le cadre du **passage à l'échelle** d'un projet logiciel tout au long de son cycle de vie

# 4 dimensions à cadrer pour garantir la maîtrise du passage à l'échelle de l'IA Générative dans le contexte du développement

## 1 Gains économiques

Quelles mesures mettre en place pour **mesurer et garantir le gain** de cette transformation ?

## 2 Maîtrise technologique

Comment mesurer précisément les **ressources IT consommées** (VM, GPU, stockage) ?

Comment sécuriser le **traitement des données sensibles** ?

Comment limiter les **risques de dépendance fournisseur** (*vendor lock-in*) ?

## 3 Organisation et pratiques

Comment garantir l'**accélération de la livraison** (*delivery*) ?

Comment se **réorganiser en interne** pour mieux déployer les nouvelles/bonnes pratiques ?

## 4 Dialogue et communication

Quel **dialogue social** instaurer ? Comment **embarquer l'ensemble des collaborateurs** ?

Quels leviers en interne pour **responsabiliser les équipes** sur leur consommation des ressources (tokens, infrastructure ...) ?

# 02

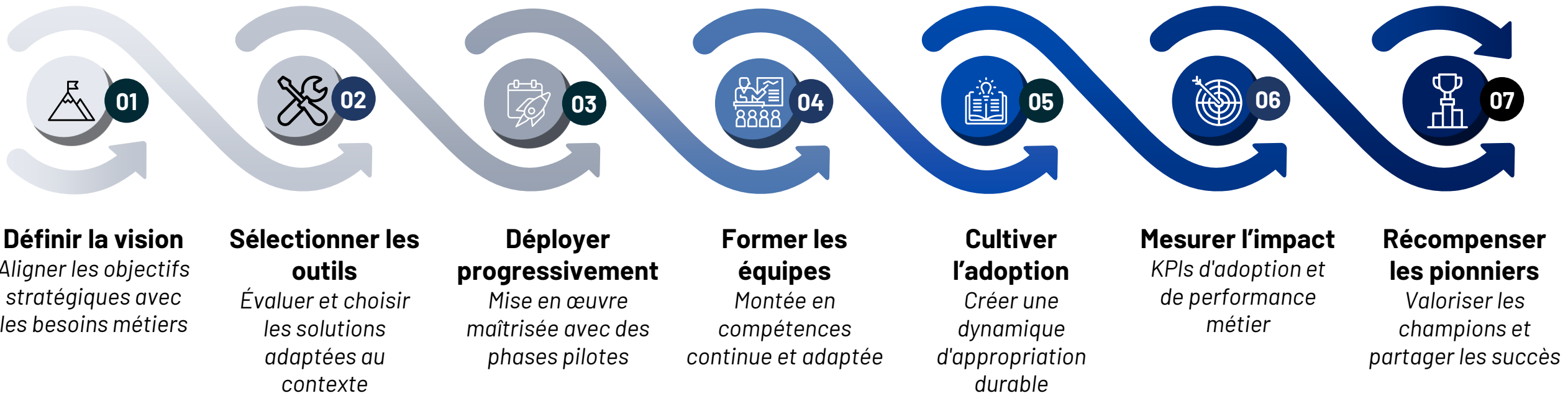
## Mettre en place la conduite du changement

# Une transformation IA réussie repose avant tout sur l'humain (!)

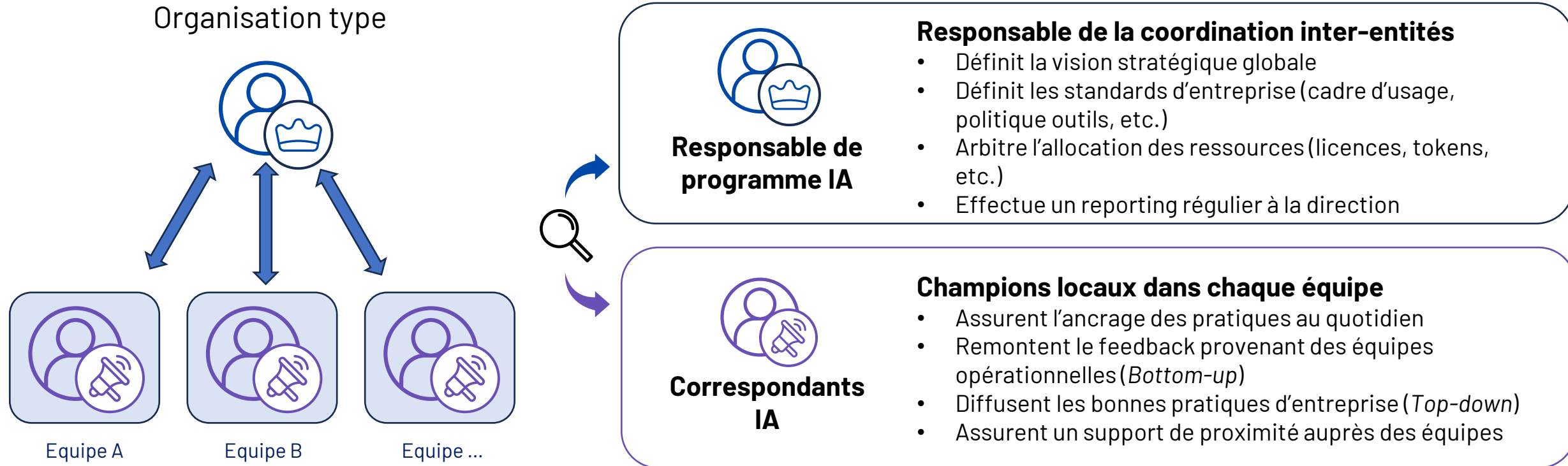
Le déploiement réussi de l'IA générative à l'échelle repose sur un cycle structuré qui équilibre dimensions technique et humaine.

Le succès dépend à **20% de la technologie** et à **80% de l'accompagnement des équipes**.

## 7 étapes génériques pour déployer l'IA à l'échelle



# Un pilotage qui doit garantir centralisation et diffusion



Les correspondants IA constituent la **clé du succès** : ils **incarnent la transformation** sur le terrain et créent le **lien entre stratégie et opérationnel**.

# Animer la transformation au quotidien

Une **formation initiale des équipes ne suffit pas** pour maintenir l'engagement des équipes. L'adoption de l'IA implique des **évolutions durables des pratiques**, qui **évoluent au fil du temps** et **s'alignent** en continu avec la **stratégie** : cette **conduite du changement** représente un **défi majeur pour nos organisations**.



## COMMUNAUTÉ DE PRATIQUE (COP)

Rituels d'échange entre pairs pour partager les cas d'usage, résoudre les blocages et diffuser les innovations. Sessions mensuelles cross-entités favorisant l'intelligence collective.



## PROGRAMMES DE FORMATION

Parcours structurés combinant expertise externe et formations internes. Du niveau débutant au niveau expert, avec certifications valorisantes.



## GAMIFICATION ET ÉVÉNEMENTS

**Hackathons IA** : Sprints créatifs pour explorer de nouveaux cas d'usages

**Café IA** : Sessions informelles pour désacraliser l'outil et partager les découvertes quotidiennes



**Ces rituels créent une dynamique d'apprentissage continu et transforment progressivement la culture de développement.**

# 03

## Pilotage et bonnes pratiques

---

# Pilotage de l'adoption et mesure des gains (1/2)



**Qu'est-ce que la "valeur"  
d'un développeur  
augmenté, et comment  
la mesurer ?**

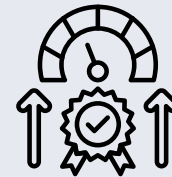


## Quelques principes clés



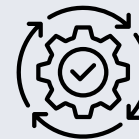
### **Productivité et rapidité : des objectifs majeurs**

L'objectif principal du passage à l'échelle est l'augmentation de la productivité, qui doit se traduire par une **réduction significative** de la **durée du cycle de vie complet** du développement logiciel.



### **Une qualité du code à ne pas négliger**

La qualité ne doit pas être sacrifiée au profit de la vitesse des développements : malgré l'accélération, les standards de **qualité doivent être maintenus** (voire renforcés).



### **Homogénéité des métriques**

Les KPIs doivent être homogènes entre les équipes d'une même entreprise ou administration publique, sous peine d'entraîner une charge de consolidation importante.



***Aujourd'hui, les gains constatés par nos organisations sont de l'ordre de 0 à 20% ; la promesse de « Développeur 10x » n'est pas d'actualité.***

# Pilotage de l'adoption et mesure des gains (2/2)

La mesure de la performance s'articule autour de **deux horizons temporels complémentaires** qui permettent d'ajuster la stratégie en continu. Ces indicateurs doivent faire l'objet d'une mesure avant l'introduction des outils IA, pour servir de point de comparaison

## Temps 1 : Lancement

*Focus Adoption*



### Adoption

- **Taux d'utilisation** par équipe
- **Fréquence d'usage** des outils
- Nombre d'**utilisateurs actifs**
- **Engagement** de la communauté

## Temps 2 : Maturité

*Focus Rentabilité & Performance*



### Vitesse de développement

- **Lead Time** : Temps entre la formulation du besoin et sa livraison finale
- **Fréquence de déploiement** : Fréquence à laquelle le code est déployé en production



### Qualité des développements

- **Change Failure Rate** : % de mises en production résultant sur des incidents et/ou services dégradés
- **Mean Time To Recover** : Temps moyen pour remédier à une défaillance produit ou système



### Vélocité des développements

- Nombre de **points de complexité** totaux réalisés par **sprint**
- **Gains de productivité** mesurés sur l'ensemble du SDLC (*Software Development Life Cycle*)

# Les bonnes pratiques de maîtrise technologique



## Bonnes pratiques



### Maîtrise et gouvernance des coûts (FinOps)

Un déploiement à l'échelle nécessite une maîtrise des coûts en termes de tokens et de traitement, mais aussi des **coûts « cachés »** : consommation ressources on-premise (VM, GPU ...).

Un modèle de **refacturation interne** peut permettre de responsabiliser les équipes



### Sécurité et gestion des données sensibles

Un déploiement à l'échelle exige des pratiques éprouvées de **classification et d'anonymisation des données**

L'usage de **LLM de confiance** est préconisé pour les **données** classifiées comme **sensibles**. Lors de leur traitement les données sont forcément en clair.

Les **juristes** sont des parties prenantes essentielles à inclure dans ces réflexions, afin délimiter les contraintes à mettre, notamment vis-à-vis de l'AI Act et des besoins de Cloud de confiance. Il faut **les inclure dès le départ** dans le cadre d'un passage à l'échelle, les réflexions associées étant très chronophages



### Gestion des fournisseurs et contractualisation

Afin de prévenir le risque du *vendor lock-in*, il est recommandé d'adopter une **stratégie multicloud / multi-LLM**, d'adopter des engagements courts (12-24 mois), d'assurer la réversibilité technique des solutions, et de mener des benchmarks réguliers à l'échelle de l'organisation sur les solutions déployées.

Un équilibre doit être trouvé entre le **besoin de stabilité des outils** (qui évoluent très vite), et la **prévention du risque de dépendance éditeur**. Dans les faits, **seules quelques solutions doivent être retenues** pour un passage à l'échelle

# Les bonnes pratiques de dialogue et de communication



## Bonnes pratiques



### Le dialogue social, pour une IA éthique et responsable

Il est obligatoire d'instaurer un **dialogue avec les instances sociales** dans le cadre d'un passage à l'échelle

Préciser **clairement les attendus de l'IA** est nécessaire vis-à-vis des salariés et cela peut se faire par la rédaction d'une **charte IA**, à visée interne et/ou externe

L'**AI Act** apporte un certain nombre de bons principes de gouvernance pour la mise en place responsable de l'IA à l'échelle (usages autorisés/interdits, données d'entraînement, etc.) → Cf. [Guide de mise en œuvre de l'AI Act Cigref-Numeum](#)

**L'état d'esprit des collaborateurs** doit faire l'objet d'un suivi régulier, pour prévenir l'anxiété due au risque d'être « remplacé », détecter les lassitudes, empêcher des dérives d'usage, etc.

**L'eNPS** (*employee Net Promoter Score* – indicateur de satisfaction salarié) peut être un moyen de repérer des **signaux faibles associés à l'usage de l'IA**

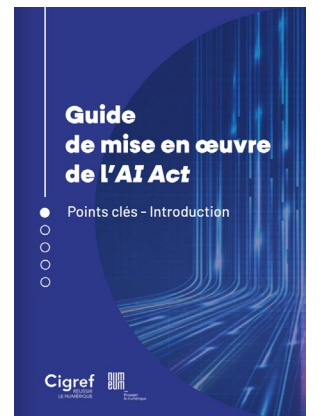
Ce suivi doit être complété d'une **communication régulière** auprès des salariés



### L'importance de la santé et du bien-être des collaborateurs



Exemple de la charte IA du Crédit Mutuel



Guide de mise en œuvre de l'AI Act du Cigref

# 04

## Impacts sur l'évolution des compétences

# Introduction sur les évolutions RH

## Nouvelles compétences introduites par l'IA Générative

### Prompting

Formuler des requêtes structurées et optimisées pour guider l'IA dans la génération de code, de tests, de docs ou d'analyses pertinentes.

### Gestion des tests assistés IA

Générer ou suggérer des cas de tests, automatiser la validation, détecter les zones non couvertes, et analyser les résultats à l'aide d'IA.

### Analyse du code IA

Utiliser l'IA pour relire, expliquer et détecter bugs, incohérences ou mauvaises pratiques dans une base de code.

### Sécurité et risques IA

Identifier les vulnérabilités induites par l'usage d'IA : fuite de données, hallucinations, dépendance excessive, biais, attaques prompt injection, conformité.

### Assistance IA


S'appuyer sur des outils d'IA pendant le développement pour accélérer la production et améliorer la qualité du code.

### Veille IA

Assurer une veille continue sur les nouveaux modèles, outils et pratiques IA, pour maintenir la stack technique à jour.

### Documentation assistée IA

Produire, maintenir et mettre à jour la documentation technique grâce à l'IA : résumés, commentaires, changelogs, exemples de code.



**Ces compétences impactent à des degrés variés les développeurs, qu'ils soient juniors ou seniors**

# Évolution des compétences du développeur

Profil développeur junior

Vision à  
janvier 2026

## COMPÉTENCES & MISSIONS

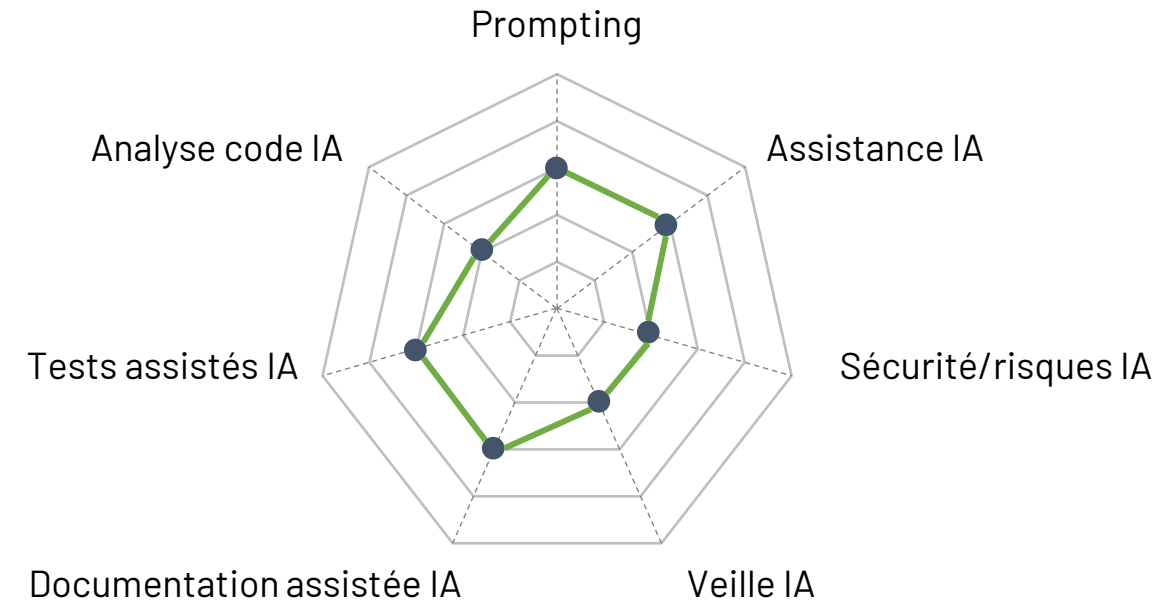
### Compétences attendues en IA

- Maîtriser les outils d'aide au développement intégrant l'IA.
- Savoir concevoir des prompts structurés pour créer ou améliorer du code (notions de prompt engineering).
- Utiliser l'IA pour :
  - la rédaction de documentation, la création d'exemples et la détection d'erreurs simples.
  - la génération de tests unitaires.
  - l'analyse rapide de code legacy ou de dépôts existants (résumés, explications).
- Être capable d'évaluer la pertinence et la cohérence du code produit par l'IA (examen critique de base).
- Comprendre les risques associés à l'utilisation de l'IA (confidentialité du code, hallucinations, sécurité).

### Exemples de missions à ajouter à la fiche de poste

- Exploiter l'IA pour optimiser la rédaction de code, la documentation, les tests et les recherches techniques.
- Collaborer à l'élaboration de playbooks IA.
- Renforcer la qualité du code grâce à l'IA.
- Participer à la veille technologique en IA (outils, plugins, extensions).

## RADAR DE COMPÉTENCES



# Évolution des compétences du développeur

Profil développeur senior

Vision à  
janvier 2026

## COMPÉTENCES & MISSIONS

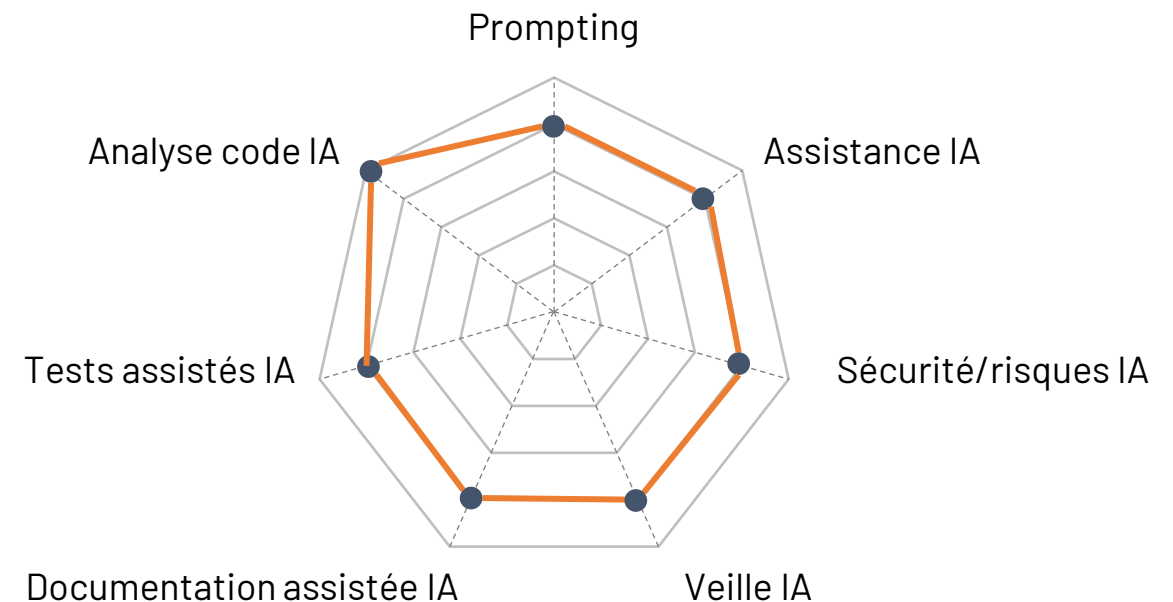
### Compétences attendues en IA

- Savoir gérer un processus de développement assisté par IA, incluant conception, codage, tests et documentation.
- Expertise dans les méthodes avancées de prompting.
- Amélioration de la productivité grâce à l'utilisation d'outils IA pour l'automatisation, les scripts et le refactoring à grande échelle.
- Aptitude à analyser, configurer et choisir les outils IA adaptés aux besoins de l'équipe.
- Contrôle approfondi du code produit par IA, notamment en termes de modèles, sécurité, architecture et performance.
- Participation à l'évolution continue des pratiques IA au sein de l'équipe, en établissant des bonnes pratiques et des directives internes.
- Capacité à accompagner et former les développeurs juniors dans l'utilisation optimale de l'IA.

### Exemples de missions à ajouter à la fiche de poste

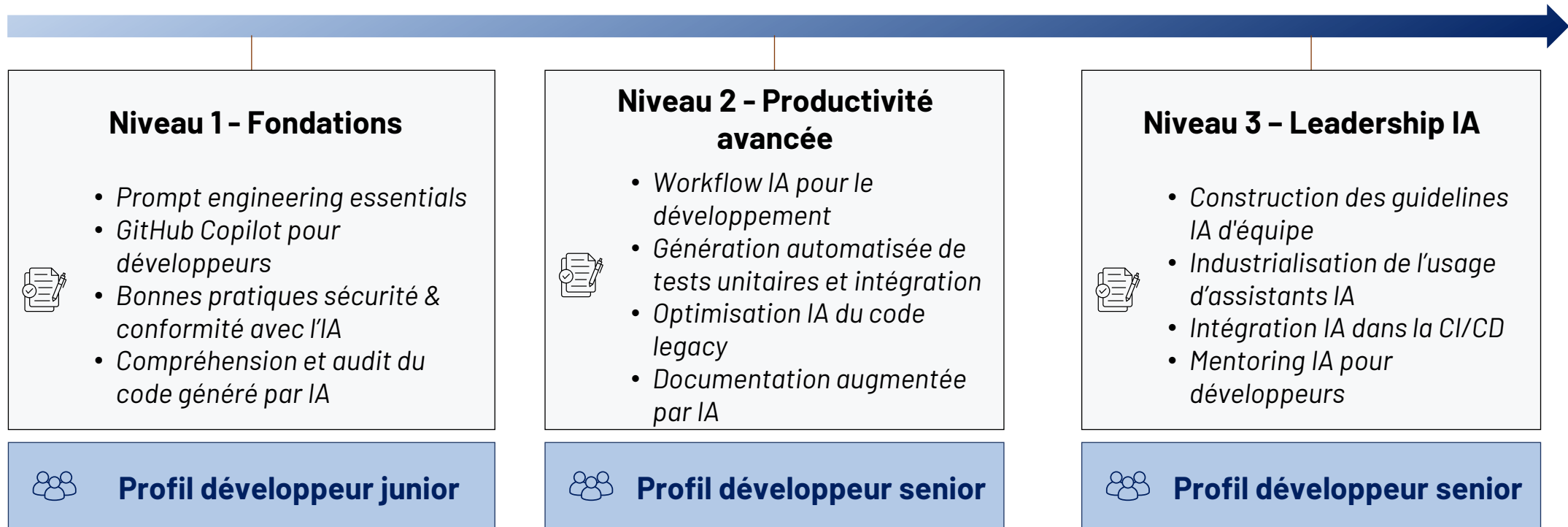
- Définir et maintenir les guidelines de l'IA dans les projets.
- Superviser les processus de travail liés à l'IA au sein de l'équipe.
- Piloter l'intégration de plugins/assistants IA dans l'environnement de développement.
- Garantir la conformité et la sécurité des usages de l'IA

## RADAR DE COMPÉTENCES



# Évolution des compétences du développeur

Formations types pertinentes



**Le format des formations** doit être ajusté en tenant compte de deux aspects principaux :

- Chaque DSI dispose déjà d'une **communauté de développeurs établie**.
- Les **formations classiques** sont **onéreuses, longues à organiser** et souvent **peu motivantes**.



Dans cette optique, **les formats tels que les hackathons** représentent une opportunité intéressante pour **renforcer l'engagement** des équipes et favoriser une **montée en compétences régulière au sein de la communauté**.

# 05

## Conclusion et perspectives

# Conclusion

L'IA générative n'est pas une fin en soi, c'est un **levier stratégique** pour la performance opérationnelle des développeurs et plus globalement de l'ensemble du SDLC.

**La vision : une plateforme cohérente où l'IA s'intègre naturellement dans tout le cycle de vie du développement (SDLC).**

**Technologie** : capacité d'industrialisation complète, de l'idée à la production



**Culture** : incitation à l'innovation, et une adoption durable et mesurable



**Gouvernance** : pilotage centralisé et diffusé, une animation continue des équipes



## Des questions qui restent ouvertes

Impact sur les organisations et évolution des pratiques actuelles à l'échelle ?

Implications du passage à l'échelle de l'ensemble du SDLC, depuis l'idée jusqu'à la production ?

Impact sur le recrutement, en particulier de juniors ?

Capacité à déployer à l'échelle des solutions dont le cycle de vie est très court ?

Intégration de la prochaine génération d'outils (IA Agentique) ?