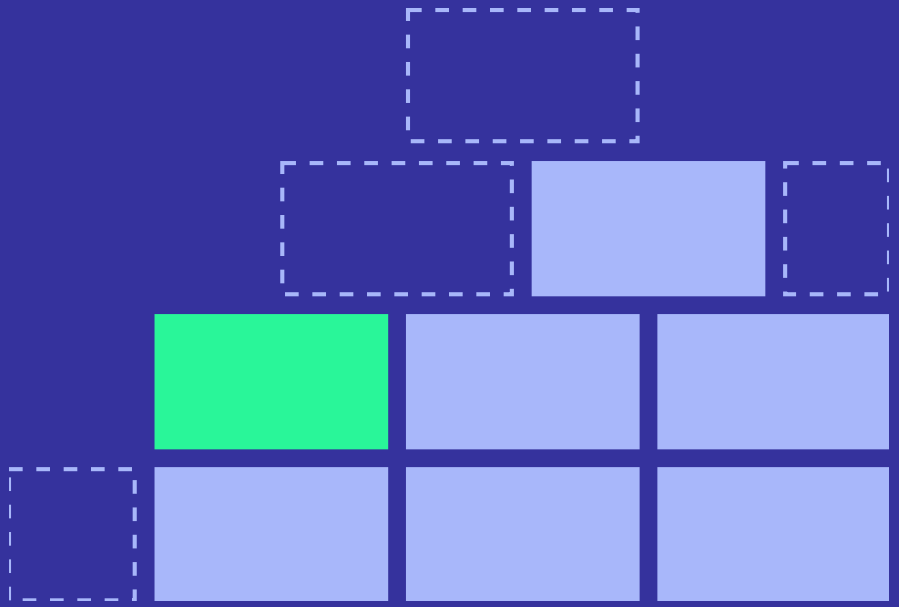
Bonx

LIVRE BLANC

De 2027 à 2030

ERP : développer ou acheter

Une entreprise industrielle doit-elle développer son ERP avec des agents de programmation IA, ou acheter une plateforme et l'adapter ?



ÉCRIT PAR

Rémi Beges

Cofondateur et CTO, Bonx



À propos de l’auteur

Rémi Beges

Cofondateur et CTO, Bonx

Rémi est docteur en génie électrique, spécialisé en semi-conducteurs. Il a ensuite travaillé pendant plusieurs années pour des industriels, en automatisant leurs processus grâce à des solutions numériques.

Il a développé des outils pour les opérateurs d’atelier. Quand ces entreprises ont eu besoin d’un ERP, rien ne convenait : les interfaces étaient lourdes, les modèles de données complexes, et aucun ne s’adaptait facilement. Il a donc développé lui-même des ERP sur mesure et des pipelines de données automatisés.

Il a ensuite rencontré Alex Barroux, arrivé à la même conclusion côté commercial, chez OSS Ventures, un studio de startups spécialisé dans l’Industrie 4.0. Ensemble, ils ont fondé Bonx.

Bonx est l’ERP industriel natif IA, conçu pour tous les types d’industriels, de l’agroalimentaire aux industries de process et de production par lots, et bien au-delà. L’équipe est basée à New York et à Paris et travaille avec plus de 80 industriels dans le monde.

Sommaire

Introduction 4

En bref 5

Notre conclusion : développer ou acheter son ERP 5

Synthèse des 11 critères 6

Analyse : les 11 critères 9

Conception technique 9

Règles métier 11

Sécurité 13

Performance 15

Fiabilité & protection des données 17

Intégrations 20

Conformité 22

Maintenance 25

Coût 26

Adaptabilité 29

Nouvelles fonctionnalités 30

Conclusions finales 33

Ce qui se passe avec le temps si vous développez en interne 33

Achetez la plateforme, développez la part qui vous est propre 35

Ce que cela implique pour vous 34

Hypothèse centrale, limites & transparence 36

Ce livre blanc examine sans détour les situations dans lesquelles développer son ERP se justifie et celles dans lesquelles il vaut mieux en acheter un.

Précisons d'emblée notre position : Bonx est l'ERP industriel natif IA. Nous commercialisons une plateforme ERP, ce qui influence forcément notre point de vue. Mais nous développons aussi chaque jour nos propres logiciels avec des agents de programmation IA, et nous nous posons la même question pour notre produit. Si des agents peuvent créer des logiciels, pourquoi un industriel achèterait-il le nôtre ?

C'est la question centrale de ce livre blanc : une entreprise industrielle doit-elle développer son ERP avec des agents de programmation IA, ou acheter une plateforme ERP et l'adapter ?

Pour trancher, nous recommandons d'examiner les 11 critères du tableau ci-dessous, avec les principaux points à prendre en compte pour chacun. La section suivante résume les résultats de la recherche et les situations qui justifient l'achat ou le développement. L'analyse détaillée examine ensuite chaque critère à partir de données publiées. Enfin, la dernière section montre comment les conséquences d'un développement mal maîtrisé évoluent dans le temps.

Critère	Points à examiner
Conception technique	Qui définit le modèle de données, les transactions et les dépendances entre modules
Règles métier	Montants, dates, valorisation des stocks, traçabilité et opérations répétées : des règles explicites et testées
Sécurité	Règles d'accès côté serveur, secrets, dépendances et surveillance
Performance	Temps de réponse avec plusieurs années de données et aux pics d'utilisation, traitements en arrière-plan
Fiabilité & protection des données	Sauvegardes testées et hors de portée des agents, droits limités, procédure de mise en production et astreinte
Intégrations	Comportement de chaque connexion lorsqu'un message est perdu, reçu plusieurs fois ou en retard
Conformité	Facturation électronique, export comptable pour les contrôles fiscaux, piste d'audit, conservation et traçabilité
Maintenance	Qui corrige et fait évoluer le système, les tests et la documentation
Coût	Coût des modèles par tâche, outils et temps des équipes
Adaptabilité	Prise en charge des processus spécifiques et sécurité de ces adaptations
Nouvelles fonctionnalités	Ce que les modèles permettent de créer au-delà d'un ERP standard, et qui peut le faire

Notre conclusion : développer ou acheter son ERP

Après avoir examiné l'ensemble des recherches, voici notre conclusion. Le reste de ce livre blanc vous donne les éléments nécessaires pour établir la vôtre.

Les tâches simples deviennent plus faciles. Les difficultés de fond demeurent.

Si les modèles de langage continuent de progresser*, écrire du code devient peu coûteux. Les recherches présentées ci-dessous montrent de réels progrès dans la création de nouvelles applications, l'optimisation, la sécurité et le respect de spécifications lorsqu'il existe une suite de tests officielle.

En revanche, elles ne montrent pas de progrès comparables pour maintenir la cohérence d'une architecture au fil des années, appliquer des règles métier non écrites, assurer des mises en production stables et l'exploitation au quotidien, gérer les intégrations ou respecter des obligations sans outil officiel de vérification. Même constat pour le coût d'utilisation des meilleurs modèles.

Les équipes expérimentées tirent davantage parti de l'IA.

Les éditeurs d'ERP, Bonx compris, n'ont pas accès à une IA secrète et plus performante. Ils utilisent les mêmes modèles que les industriels. L'IA seule ne donne donc pas à un industriel un avantage sur un éditeur : les deux disposent des mêmes outils.

Cela dit, la plus grande étude disponible à ce jour constate que les développeurs expérimentés réalisent la quasi-totalité des gains de productivité liés à l'IA. Les progrès des modèles tendent ainsi à creuser l'écart entre une équipe qui développe et exploite des ERP depuis des années et une entreprise qui crée son premier ERP.

Autrement dit, à mesure que l'IA progresse, les éditeurs disposant de développeurs expérimentés devraient prendre davantage d'avance sur les entreprises qui créent leur premier ERP. C'est l'expérience qui transforme l'IA en meilleur produit, pas le simple accès aux modèles.

L'essentiel

Développer son ERP se justifie lorsque la part des besoins propres à l'entreprise approche 80 %. Dans les autres cas, les données plaident surtout pour l'achat d'une plateforme *adaptable*, sur laquelle développer la partie spécifique avec l'IA.

***Hypothèse centrale, limites & transparence**

L'analyse suppose que les grands modèles de langage (LLM) continuent de progresser. Leurs progrès ont été rapides au cours des 24 à 36 derniers mois. Nous supposons qu'ils se poursuivront pendant les 24 à 48 prochains mois.

> **12 % → 85 %**
> Tâches pour lesquelles le meilleur modèle égale ou dépasse un professionnel, de mai 2024 à avril 2026 (GDPval).

Ces chiffres sont à lire avec prudence : les deux derniers sont publiés par OpenAI sur ses propres modèles et son propre benchmark.

Note complète, page 36

Synthèse des 11 critères

Ce tableau résume la suite du livre blanc. Pour chaque critère, il présente les résultats de la recherche, les perspectives et les situations qui justifient l'achat ou le développement. L'analyse complète suit le tableau.

Critère	Résultats de la recherche	Perspectives 2027-2030	Acheter si...	Développer si...
Conception technique	Les agents IA font évoluer correctement une application existante environ une fois sur trois. Dans 39 % des échecs, ils provoquent une régression.	Le prototypage est maîtrisé. Maintenir la cohérence de l'architecture au fil des évolutions reste un point très faible.	Personne dans l'entreprise n'a les compétences pour être responsable du modèle de données et de la structure.	Un ingénieur expérimenté prend les décisions structurelles et vérifie le travail des agents.
Règles métier	Sur de longues séries d'écritures comptables, les meilleurs agents produisent un bilan exact dans au plus 46 % des cas.	Les règles théoriques sont connues. Les longues séquences et les règles non écrites restent problématiques.	Vous avez des règles spécifiques, mais avez aussi besoin de bases solides sur les fonctions ERP courantes, comme les stocks et la facturation.	Environ 80 % de vos règles et workflows sont propres à votre activité.
Sécurité	Seuls 37 % du code fonctionnel du meilleur agent (septembre 2026) sont également sécurisés.	Les progrès sont rapides, mais les attaquants utilisent les mêmes outils.	Personne ne peut surveiller le système, appliquer les correctifs et répondre aux incidents de sécurité.	Une personne dans l'entreprise possède les compétences techniques nécessaires pour gérer la sécurité.

En bref

Critère	Résultats de la recherche	Perspectives 2027-2030	Acheter si...	Développer si...
Performance	Sur des tâches réelles d'optimisation, le meilleur agent atteint moins d'un quart du gain de vitesse d'un expert.	Les progrès sont importants lorsque l'objectif est explicite.	Personne ne sait prendre les décisions d'architecture dont dépend la performance, ni assurer son suivi et sa maintenance.	Les volumes restent modestes (tables de moins de 50 000 lignes), aucune croissance n'est prévue, une personne ayant des bases techniques suit les performances et vous acceptez un système plus lent qu'un produit conçu par des professionnels. Les seules tables de traçabilité peuvent dépasser 1 million de lignes en quelques mois.
Fiabilité & protection des données	Les équipes ayant adopté l'IA ont effectué des mises en production moins stables en 2024 et 2025.	Aucune amélioration mesurée.	Personne ne sait gérer les sauvegardes et vérifier leur fonctionnement, ou une perte de données est inacceptable.	Une personne de toute confiance garantit la protection des données, ou vous acceptez de les perdre du jour au lendemain si un agent efface tout.
Intégrations	Les développeurs consacrent 39 % de leur temps à créer de nouvelles intégrations.	Les connecteurs s'écrivent plus vite, mais les systèmes à connecter sont plus nombreux.	Les connexions nécessaires existent déjà sur la plateforme.	Les connexions sont réellement propres à votre activité, par exemple avec des logiciels de niche ou des machines spécifiques.
Conformité	Les agents approchent 100 % lorsqu'une suite de tests officielle vérifie la règle, contre 44 à 76 % lorsque celle-ci n'existe que sous forme de texte.	Des progrès là où existent des contrôles automatiques, mais les règles changent chaque année.	Les obligations sont courantes : facturation électronique, export comptable et traçabilité.	Les obligations sont propres à votre secteur et un expert les traduit en contrôles.

En bref

Critère	Résultats de la recherche	Perspectives 2027-2030	Acheter si...	Développer si...
Maintenance	La maintenance représente 40 à 80 % du coût d'un logiciel sur sa durée de vie. Cette estimation classique est antérieure aux agents de programmation.	L'IA produit davantage de code à maintenir, dont davantage de copies.	Personne ne connaîtra suffisamment le code pour le corriger et le faire évoluer pendant toute la durée d'utilisation de l'ERP.	L'entreprise conserve des personnes qui connaissent le code et savent le corriger et le modifier aussi longtemps que l'ERP est utilisé.
Coût	Un système développé en interne coûte 1,7 à 5 fois son coût de développement sur sa durée de vie. Un développeur coûte environ 200 000 $ par an aux États-Unis.	Le développement devient moins cher, la maintenance moins nettement, et le meilleur modèle coûte davantage par tâche.	Le nombre d'utilisateurs est faible par rapport à l'équipe nécessaire pour développer et maintenir le système, et il vaut mieux réserver la trésorerie à l'activité.	Vous avez de nombreux utilisateurs, une équipe de développement déjà en place et préférez détenir un actif tout en assumant des charges salariales fixes.
Adaptabilité	Parmi les entreprises, 38 % ont modifié leurs processus pour s'adapter à leur ERP.	Adapter une plateforme devient de moins en moins coûteux.	La plateforme peut être adaptée à vos données sans modifier son cœur.	La plateforme ne s'adapte qu'en modifiant le code de son cœur, ou près de 80 % de vos besoins sont spécifiques plutôt que 20 %.
Nouvelles fonctionnalités	Sur 160 097 développeurs, les plus expérimentés bénéficient de la quasi-totalité des gains de productivité liés à l'IA et des progrès qu'elle permet dans l'exploration de nouveaux domaines.	Tous disposent des mêmes modèles, mais les équipes expérimentées en tirent davantage parti.	Vos efforts serviraient à recréer des fonctions standard.	Votre équipe d'ingénierie et de produit sait créer ce qu'aucune plateforme ne propose.

Conception technique

Dans le meilleur des cas : une architecture cohérente maintient la rapidité et la stabilité du système à mesure qu'il se développe, tout en limitant le coût des évolutions.

Dans le pire des cas : l'architecture se définit par défaut, une demande après l'autre, sans cohérence d'ensemble. Les correctifs s'accumulent sans résoudre les défauts de structure, et chaque évolution devient plus lente et plus risquée.

Verdict

Aujourd'hui

les agents identifient les composants d'un système, mais se trompent sur leurs relations. Ils parviennent presque toujours à créer une nouvelle application. En revanche, lorsqu'ils doivent faire évoluer une application existante, ils échouent deux fois sur trois, souvent en provoquant des régressions.

De 2027 à 2030

les agents continueront de progresser dans l'écriture de nouveau code. Les données ne montrent pas encore qu'ils savent maintenir la cohérence d'une architecture au fil des années d'évolution, ce qu'exige un ERP. Une entreprise sans équipe technique qui crée son ERP à partir de zéro avec des agents obtiendra ses premiers écrans en quelques semaines. Chaque demande ultérieure sera ensuite traitée isolément : une nouvelle table ici, une règle copiée ailleurs. Après un an ou deux, les modifications commenceront à provoquer des régressions, sans que personne dans l'entreprise puisse en expliquer la cause ni corriger la structure. À ce stade, il faudra probablement abandonner le logiciel, et perdre l'investissement engagé, qui pourra alors atteindre plusieurs millions de dollars, même avec un recours massif à l'IA et une équipe de développement interne réduite.

Éléments de preuve

Il s'agit ici de comprendre ce qui se passe lorsqu'un grand modèle de langage (LLM) prend les décisions structurelles à la place d'un ingénieur : modèle de données, limites des transactions, dépendances entre modules, etc.

- Fred Brooks écrivait que « **l'intégrité conceptuelle** est le critère le plus important dans la conception d'un système » (The Mythical Man-Month, chap. 4). David Parnas a montré que **des modifications réalisées sans comprendre la conception initiale dégradent la structure jusqu'à ce que plus personne ne comprenne le produit** (Software Aging, ICSE 1994).
- Dans sept projets Apache, les défauts d'architecture absorbaient 51 à 85 % de l'effort de maintenance. Les cinq défauts les plus coûteux concernaient 8 à 25 % des fichiers sujets aux erreurs, mais mobilisaient 20 à 61 % de l'effort. Leur coût augmentait avec le temps (Xiao et al., ICSE 2016).
- Les agents choisissent les frameworks et l'infrastructure, mais presque personne ne traite ces choix comme des décisions d'architecture à vérifier. La seule formulation d'un prompt a produit des systèmes aux structures différentes pour une même tâche (Konrad et al., avril 2026).
- Lorsqu'on leur demandait de transformer des exigences en architecture pour 68 projets, les modèles identifiaient la plupart des composants, mais obtenaient au mieux 0,18 sur 1 pour les relations entre ces composants. Environ un tiers des choix de conception ne reposaient sur aucune exigence. Les relations inventées étaient l'erreur la plus fréquente (R2ABench, 2026).

- Sur 3 275 tâches de normalisation de bases de données, les modèles échouaient régulièrement à déduire les dépendances entre données et à reconstituer les contraintes entre tables (DNBench, septembre 2026).
- Lorsqu'on leur demandait de recréer 200 programmes existants, allant d'outils en ligne de commande à SQLite, aucun modèle ne parvenait à en reproduire un seul entièrement dans l'étude initiale. Au classement de septembre 2026, le meilleur modèle en résout entièrement 4,5 % et presque entièrement 37 % (ProgramBench).
- Créer une **nouvelle** application web **(c'est bien le mot « nouvelle » qui compte ici)** à partir d'une spécification est un problème presque résolu : le meilleur modèle atteint 92 % sur la version actuelle de Vibe Code Bench, contre 41 % en février 2026, en partie grâce à une révision du benchmark (Vals.ai). **Faire évoluer une application existante au fil de 10 demandes successives reste en revanche difficile :** le meilleur score est de 34 %, et 39 % des échecs provoquent une régression (VCB 1-100).
- Les architectes interrogés en 2026 utilisent l'IA pour des tâches mécaniques et rarement pour des décisions qui dépendent du contexte du projet (Felder et al., FSE 2026).
- Dans un benchmark où les agents font évoluer la même base de code sur de nombreuses étapes, la structure se dégradait dans 80 % des exécutions. Le code des agents était 2,2 fois plus verbeux que le code humain. De meilleurs prompts amélioraient la qualité initiale, sans ralentir la dégradation (SlopCodeBench, mars 2026).
- Après l'adoption de l'éditeur IA Cursor, le développement de 806 projets open source s'est accéléré pendant un à deux mois, puis a ralenti à mesure que leur code devenait plus complexe et plus difficile à modifier (Carnegie Mellon, MSR 2026).
- À l'inverse, une étude de 151 projets Java n'a constaté aucune évolution des problèmes d'architecture après l'adoption d'agents, alors que le volume de code augmentait de 12,8 % (Larsen et Moghaddam, juin 2026).

La plupart des données récentes proviennent de prépublications de 2026, et une étude ne relève aucun effet. Une tendance se dégage néanmoins : les modèles traitent bien les éléments pris isolément, mais perdent en cohérence dans les relations entre ces éléments et au fil des modifications. Or un ERP évolue pendant des années, précisément dans les conditions où les agents obtiennent leurs plus faibles scores. Ce constat rejoint 50 ans de recherche en génie logiciel : la qualité d'un système dépend de ses premières décisions structurelles (Roychoudhury et Zeller, 2025).

Règles métier

Dans le meilleur des cas : les règles communes à toutes les entreprises sont appliquées correctement partout : montants, dates, valorisation des stocks, traçabilité, etc.

Dans le pire des cas : de petites erreurs silencieuses affectent la comptabilité, le bilan ou la délimitation d'un rappel de produits.

Verdict

Aujourd'hui

les agents connaissent les règles théoriques, mais appliquent souvent mal les règles métier lorsqu'aucun contrôle ne les vérifie : des bilans exacts dans au plus 46 % des cas sur de longues séries d'écritures, des commandes acceptées malgré des quantités négatives et, lorsqu'une règle manque, du code écrit sans demande de précision.

De 2027 à 2030

les progrès des modèles ne fourniront pas les règles que personne n'a écrites. Les erreurs ne sont détectées que si quelqu'un formule précisément les règles et les vérifie automatiquement. Une entreprise sans équipe technique qui développe son ERP à partir de zéro obtiendra des écrans apparemment corrects et des chiffres légèrement faux : un écart d'arrondi sur chaque facture, deux méthodes de valorisation du même stock, un utilisateur qui voit des marges auxquelles il ne devrait pas avoir accès. Ces erreurs resteront invisibles jusqu'à ce qu'une clôture annuelle, un contrôle fiscal ou un rappel les révèle.

Éléments de preuve

Certaines règles sont universelles. Les montants se calculent avec des nombres décimaux exacts, avec des arrondis à des étapes définies. Une opération relancée après une erreur réseau ne doit pas s'appliquer deux fois. Les dates sont enregistrées avec leur fuseau horaire. Deux mises à jour simultanées d'un même stock ne doivent pas s'écraser mutuellement. Chaque écriture comptable doit être équilibrée. Une seule méthode cohérente doit servir à valoriser le stock.

- Sur 392 tâches de développement côté serveur, environ la moitié des programmes fonctionnellement corrects étaient exploitables par un attaquant. Selon la tâche, des défauts de contrôle d'accès apparaissaient dans 22 à 51 % des résultats corrects. Une spécification précise de l'interface améliorait la justesse de 7 à 9 points, alors qu'un rappel générique de sécurité ne changeait presque rien (BaxBench, ICML 2025).
- Lorsque cinq agents ont chacun créé trois applications, quatre acceptaient des commandes aux quantités négatives et trois des prix négatifs, parce que la demande ne l'interdisait pas (SecurityWeek, janvier 2026).
- Une étude a examiné directement les montants, les dates et les opérations répétées. Dans du code côté serveur écrit par cinq modèles de pointe pour 50 tâches sans spécification, elle a recensé 66 défauts sur les montants (calcul en virgule flottante, arrondis), 32 sur les opérations répétées, 31 sur les dates et 19 sur les accès. Une spécification de 267 mots les réduisait de 84 %. C'est la seule étude de ce type à ce jour. Elle a été rédigée par un auteur unique, qui a aussi créé les outils de vérification, et n'a pas été évaluée par les pairs (Dhuri, septembre 2026).

- Les modèles de pointe maîtrisent les exercices classiques de comptabilité en partie double : Gemini 3 réussissait 16 contrôles sur 16 et Claude Opus 4.5, 15/16 (Modern Treasury, janvier 2026, quatre scénarios, test réalisé par un fournisseur).
- Sur des séquences plus longues, six modèles produisaient un bilan exact dans au plus 46 % des cas, avec un écart de 26 à 41 points entre leur bilan et celui recalculé à partir de leurs propres écritures (FinBalance, juin 2026). Sur 160 tâches comptables de niveau expert, le meilleur modèle atteignait 56,4 %, et aucun ne dépassait 2,6 % lorsqu'il fallait réussir la même tâche huit fois de suite (APEX-Accounting, juillet 2026).
- Les tests unitaires ne peuvent pas explorer tous les ordres d'exécution des opérations simultanées. Ils valident donc du code concurrent pourtant défectueux (CONCUR, 2026).
- Lorsqu'une exigence manque, les modèles écrivent du code au lieu de poser des questions dans plus de 60 % des cas (HumanEvalComm, ACM TOSEM 2025). Les agents récents ne demandent presque jamais de précisions, sauf instruction explicite (Ambig-SWE, ICLR 2026). Les modèles devinent correctement une exigence non écrite dans 41 % des cas (Carnegie Mellon).

Les modèles de pointe connaissent les règles théoriques. Ils échouent sur les longues séquences, les opérations simultanées et les règles non écrites. Des spécifications précises et vérifiables les aident ; des rappels génériques, non. Dans tous les cas, quelqu'un doit connaître les règles et les formaliser.

Sécurité

Dans le meilleur des cas : les prix, les marges, les recettes et les données clients restent protégés, et la production continue.

Dans le pire des cas : vol de données ou rançongiciel. Après la cyberattaque de 2025 contre Jaguar Land Rover, la production s'est arrêtée pendant environ cinq semaines. Le coût modélisé pour l'économie britannique atteignait 1,9 milliard de livres sterling (Cyber Monitoring Centre).

Verdict

Aujourd'hui

le meilleur agent écrit du code à la fois fonctionnel et sécurisé sur environ un tiers des tâches. Les applications créées ainsi exposent déjà des données et des clés d'accès à grande échelle.

De 2027 à 2030

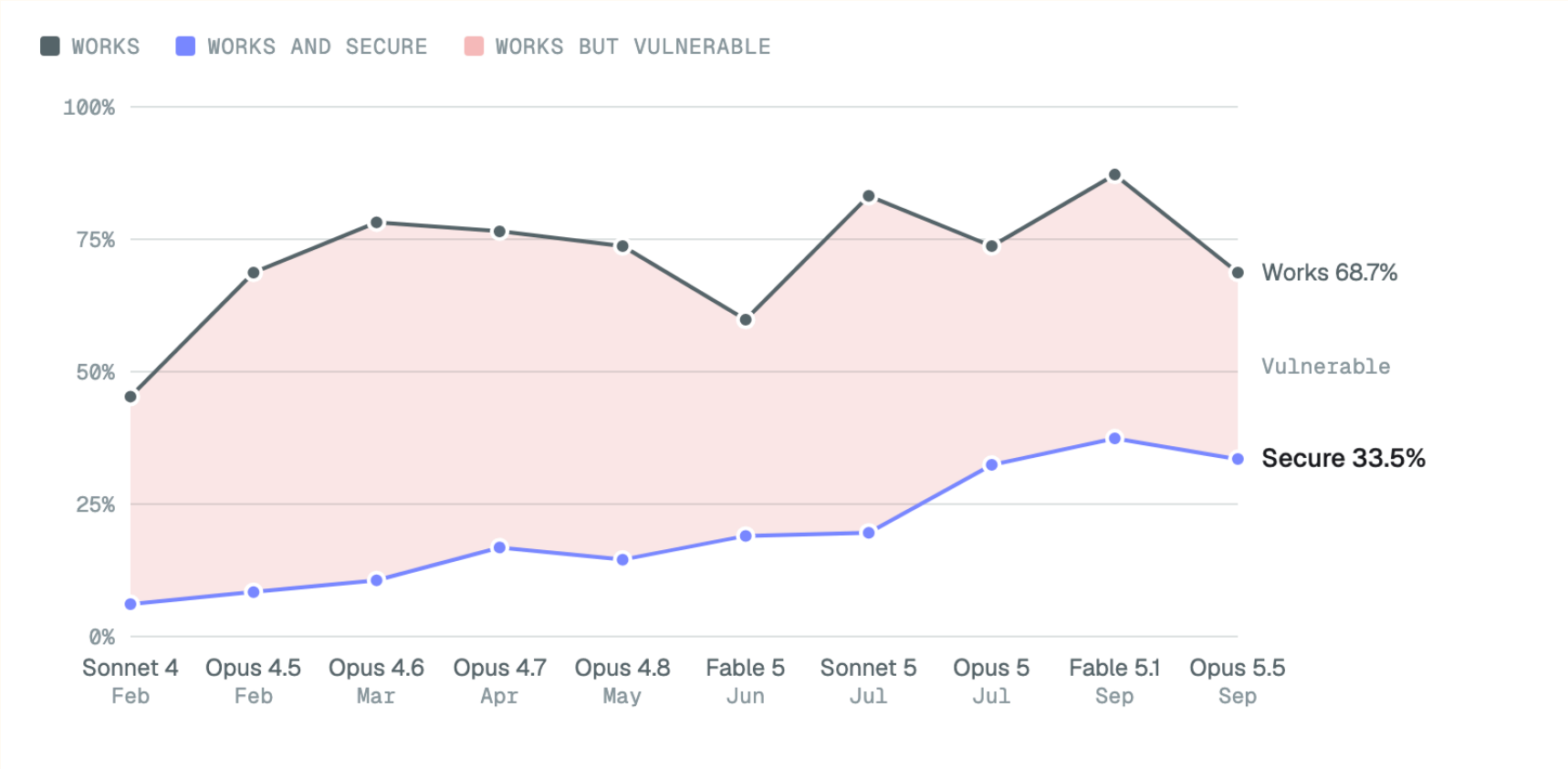
au rythme actuel, les agents écriront du code sécurisé sur la plupart des tâches avant 2030. Mais les attaquants utilisent les mêmes modèles, et en 2025, des failles étaient déjà exploitées avant qu'un correctif existe. Pour une entreprise sans équipe technique qui développe son ERP à partir de zéro, le principal risque est l'absence de suivi une fois le système opérationnel : aucune surveillance, aucun correctif appliqué, aucun plan en cas d'attaque. Une clé exposée ou une bibliothèque obsolète peut alors rester vulnérable pendant des mois, jusqu'à ce qu'une fuite de données ou un rançongiciel arrête la production.

Éléments de preuve

- Veracode teste les modèles sans agent. Sa moyenne restait à 55 % de code sécurisé en 2025 et 56 % en 2026, alors que plus de 95 % du code compile. Ajouter des consignes de sécurité au prompt n'a pas réduit les vulnérabilités de manière statistiquement significative (Kharma et al., mai 2026).
- L'analyse de 5 600 applications publiques créées par génération de code a trouvé plus de 2 000 vulnérabilités, plus de 400 secrets exposés et 175 expositions de données personnelles (Escape.tech, octobre 2025). Sur 1 645 applications Lovable, 170 permettaient à n'importe qui de lire leurs données (CVE-2025-48757).
- Les commits réalisés avec Claude Code exposaient des secrets dans 3,2 % des cas, contre 1,5 % en moyenne sur GitHub (GitGuardian, 2026).
- Les attaquants utilisent les mêmes modèles. Dans une campagne d'espionnage décrite par Anthropic en novembre 2025, l'IA a effectué 80 à 90 % du travail. Mandiant a mesuré qu'en 2025, les vulnérabilités étaient exploitées en moyenne sept jours avant l'existence d'un correctif.

La sécurité du code écrit par les agents progresse rapidement à partir d'un niveau faible. Les capacités d'attaque progressent en même temps.

- Sur le benchmark d'Endor Labs, la part du code à la fois fonctionnel et sécurisé est passée de 6 % à 37 % entre février et septembre 2026. Avec le meilleur couple agent-modèle mesuré, 57 % du code fonctionnel reste vulnérable.



Endor Labs, Agent Security League, environnement d'exécution Claude Code, 200 tâches Python et 77 types de vulnérabilités. Les dates correspondent aux tests réalisés en 2026.

Performance

Dans le meilleur des cas : les écrans et les calculs restent rapides malgré l'accumulation de plusieurs années de données : des millions de mouvements de stock, des dizaines d'utilisateurs lors des changements d'équipe, un calcul des besoins matières sur toute la nomenclature, un scan qui doit répondre instantanément sur la ligne.

Dans le pire des cas : des écrans lents dans l'atelier et des traitements nocturnes qui ne se terminent pas, découverts des mois après le go-live.

Verdict

Aujourd'hui

les agents optimisent bien le code lorsqu'on le leur demande avec un objectif mesurable. Par défaut, leur code est moins efficace que celui d'un expert. Ils échouent plus souvent en présence de contraintes de base de données et d'architecture, écrivent du code d'infrastructure correct moins d'une fois sur cinq et mesurent rarement les performances de leur propre initiative.

De 2027 à 2030

l'optimisation avec un objectif explicite progresse plus vite que toutes les autres mesures examinées ici et devrait atteindre le niveau expert. Rien dans les données ne montre que les agents définissent eux-mêmes ces objectifs ou prennent seuls les décisions d'infrastructure. Une entreprise sans équipe technique qui développe son ERP à partir de zéro obtiendra un système performant sur des données de test. Avec l'accumulation des mouvements de stock, les écrans ralentiront dans l'atelier et les traitements nocturnes ne finiront plus à temps. Personne ne saura s'il faut ajouter un index, une file de tâches ou changer d'architecture.

Éléments de preuve

- Sur 1 000 problèmes, le code écrit par GPT-4 s'exécutait en moyenne 3,12 fois plus lentement que la solution de référence, et jusqu'à 13,9 fois plus lentement avec 43,9 fois plus de mémoire dans les cas les plus défavorables (EffiBench, NeurIPS 2024). Sur six langages, le meilleur modèle atteignait environ 62 % de l'efficacité d'un expert (EffiBench-X, 2025).
- Sur 498 tâches réelles d'optimisation, Claude 4.5 Opus obtenait 0,225 fois le gain de vitesse de l'expert, contre 0,024 pour Claude 3.7 Sonnet (SWE-fficiency, ICML 2026). Sur GSO, la part des tâches d'optimisation résolues est passée de 3,8 % en mai 2025 à 88,2 % en septembre 2026, en partie parce que l'environnement de test a commencé à demander aux agents de poursuivre l'optimisation.
- Les modèles de code échouent généralement sur les exigences non fonctionnelles, comme l'efficacité et la robustesse (NoFunEval, COLM 2024). L'ajout de contraintes d'architecture et de base de données réduisait le taux de réussite de 27 points en moyenne. Les contraintes de base de données étaient les plus pénalisantes (Constraint Decay, 2026).

- Une grande partie des performances d'un ERP dépend de choix techniques d'infrastructure que les demandes adressées aux agents ne mentionnent pas. Prenons deux exemples. Une opération longue, comme un calcul des besoins matières sur toute la nomenclature, peut s'exécuter en arrière-plan via une file de tâches ou bloquer l'écran jusqu'à sa fin. Un message entre deux systèmes peut être transmis au plus une fois, au risque d'être perdu ; au moins une fois, au risque d'être appliqué deux fois ; ou effectivement une seule fois grâce à la déduplication, au prix d'un travail d'ingénierie supplémentaire (Kleppmann, notes de cours de Cambridge). Les agents sont peu performants à ce niveau.
  - Le premier benchmark de code d'infrastructure pour AWS constatait que GPT-4 produisait un résultat correct dans 19,4 % des cas, contre 86,6 % sur des tâches Python ordinaires (IaC-Eval, NeurIPS 2024). L'ajout de connaissances sur la configuration faisait passer la réussite de 27 % à 63 %, mais l'adéquation au besoin réel de l'utilisateur cessait de progresser. Les auteurs résument ce constat ainsi : les modèles codent bien, mais conçoivent mal (Nekrasov et al., décembre 2025).
  - Dans un test de 2026, le code Terraform écrit par Claude Opus 4 passait le scanner de sécurité Checkov dans 23,1 % des cas (SBSeg 2026). Détailler les exigences non fonctionnelles améliore le code produit par les agents (Pereira et Garcia, août 2026). Par ailleurs, 35,2 % du code généré est moins robuste que le code humain, principalement faute de contrôles (Li et al., 2025).
- Les modifications écrites par des agents incluent moins souvent un contrôle de performance que celles écrites par des humains : 45,7 % contre 63,6 % (décembre 2025). Aucune étude publiée ne mesure le comportement sous charge d'une application métier écrite par un agent.

Les modèles progressent fortement lorsque la performance est un objectif explicite. Sinon, leur code reste moins efficace que celui d'un expert, et les choix d'infrastructure ne sont traités que si quelqu'un le demande.

Fiabilité & protection des données

Dans le meilleur des cas : l'ERP est disponible dès que l'usine fonctionne, et son historique reste intact.

Dans le pire des cas : une panne bloque les expéditions. Perdre l'historique des mouvements de stock ou le carnet de commandes en cours empêche l'entreprise de facturer ses livraisons, de valoriser son stock ou de retracer un lot lors d'un rappel. Pour une PME industrielle, une heure d'arrêt de production peut coûter jusqu'à 150 000 $ (Siemens, True Cost of Downtime 2024, mesure portant sur les arrêts de machines).

Verdict

Aujourd'hui

les agents et l'automatisation ont déjà détruit des données de production, aussi bien dans de petites entreprises que dans des groupes de la taille d'Amazon. Seules les données conservées hors de leur portée ont pu être récupérées. Les agents chargés de l'exploitation résolvent au mieux environ la moitié des incidents réels. L'adoption de l'IA reste associée à des mises en production moins stables.

De 2027 à 2030

les agents gagneront en autonomie et en droits d'accès. Une seule action erronée pourra donc affecter davantage de données. La protection dépend de la configuration : copies inaccessibles aux agents, suppression différée, droits restreints et restaurations testées régulièrement. Une entreprise sans équipe technique qui développe son ERP à partir de zéro risque d'accorder des accès étendus à l'agent pour qu'il travaille vite, et de laisser les sauvegardes accessibles avec ces mêmes droits, comme dans le cas de PocketOS. Elle peut ne découvrir une perte qu'au moment de rechercher les données pour une facture, un inventaire ou un rappel.

Éléments de preuve

- Dans l'enquête DORA 2024, menée auprès d'environ 3 000 professionnels, chaque hausse de 25 % de l'adoption de l'IA était associée à une baisse de 7,2 % de la stabilité des mises en production. Dans DORA 2025, avec près de 5 000 répondants, la relation avec la vitesse de livraison devenait positive, mais celle avec la stabilité restait négative.
- SWE-bench rassemble des bugs réels signalés dans des projets open source populaires. Un correctif d'IA est considéré comme réussi s'il passe les tests du projet (SWE-bench). Lorsque des responsables de projets ont examiné 296 correctifs d'agents ayant passé ces tests, environ la moitié auraient été refusés (METR, mars 2026).
- L'exploitation des systèmes est encore plus difficile. Les agents résolvaient 11,4 % des scénarios de fiabilité d'ITBench (ICML 2025). Sur 59 incidents Kubernetes, tous les modèles restaient sous 50 % au lancement de mai 2026 (ITBench-AA). Sur des pannes aux causes multiples ou de bas niveau, la réussite tombait à 15 à 28 % (SREGym, NeurIPS 2026).

Des agents et des automatismes ayant accès aux systèmes de production ont déjà détruit des données d'entreprise à grande échelle :

Oct. 2026

TV4

Le montage terminé d'une saison télévisée a été perdu, repoussant la première diffusion à 2027. Selon la presse, un agent IA disposant d'un accès au serveur l'aurait supprimé. TV4 nie l'intervention d'un agent autonome et indique que la sauvegarde a échoué (Metro Mode).

Données récupérées : Non

Avr. 2026

PocketOS

Un agent Cursor utilisant Claude Opus 4.6 a trouvé un jeton valable pour tout le compte et supprimé la base de production et ses sauvegardes en environ 9 secondes (Railway).

Données récupérées : Récupération par l'hébergeur, qui a ensuite ajouté une suppression différée

Mars 2026

Amazon retail

Pannes de la boutique, avec environ 120 000 et 6,3 millions de commandes perdues selon des documents internes rapportés par la presse. Une note interne mentionne des modifications assistées par IA. Amazon indique qu'un incident impliquait un conseil tiré par un agent d'un wiki obsolète, sans code écrit par IA (Fortune, Amazon).

Données récupérées : Ventes perdues, sans perte d'enregistrements. Validation d'un responsable expérimenté désormais requise pour les modifications assistées par IA

Mars 2026

Meta

Un agent interne a publié un conseil sans validation. Un salarié l'a suivi, rendant les données de l'entreprise et des utilisateurs accessibles à des ingénieurs non autorisés pendant environ deux heures. L'incident a été classé au deuxième niveau de gravité le plus élevé (TechCrunch).

Données récupérées : Aucun usage abusif signalé

Févr. 2026

DataTalks.Club

Claude Code a lancé une destruction d'infrastructure avec une configuration obsolète, supprimant une base de production contenant 2,5 ans de données ainsi que ses instantanés visibles (analyse de l'incident).

Données récupérées : Restauration après 24 heures à partir d'un instantané cloud caché

Déc. 2025

Amazon Web Services

Selon le Financial Times, un service est resté indisponible pendant 13 heures après que l'agent de programmation Kiro a décidé de supprimer puis de recréer son environnement (The Decoder). Amazon invoque une mauvaise configuration des contrôles d'accès et affirme qu'un seul service, dans une seule région, a été touché (Amazon).

Données récupérées : Aucune perte de données signalée ; cause contestée

Mai 2024

UniSuper

Un processus automatisé de Google Cloud a supprimé le cloud privé d'un fonds de pension de 647 000 adhérents, dans ses deux régions (communiqué commun).

Données récupérées : Oui, après environ deux semaines, grâce à des sauvegardes chez un autre prestataire

Août 2020

KPMG

Une politique de conservation destinée à un utilisateur a été appliquée à tout l'environnement, supprimant les conversations Teams d'environ 145 000 salariés (The Register).

Données récupérées : Non, selon une note interne

Janv. 2017

GitLab

Un ingénieur a effacé la base de production, et les cinq méthodes de sauvegarde ont toutes échoué (analyse de l'incident).

Données récupérées : Six heures de données définitivement perdues

Plusieurs récits concernant Amazon et TV4 sont contestés : le tableau présente les deux versions. Une étude de 344 incidents d'IA concernant les entreprises en a recensé 188 dans lesquels un système autonome avait causé des dommages sans intervention d'un attaquant, dont 65 suppressions de données ou de code (Cyera Research, mai 2026). La restauration n'est pas garantie non plus : parmi 900 organisations attaquées en 2024, seules 10 % ont récupéré plus de 90 % de leurs données (Veeam 2025).

Ces pertes sont survenues dans des entreprises disposant d'équipes techniques expérimentées. Acheter ne supprime donc pas le risque. Une entreprise sans cette expertise est plus exposée et peut ignorer la perte jusqu'au moment où elle a besoin des données. Dans tous les cas de récupération, les données provenaient d'une copie hors de portée du système défaillant : instantané caché, suppression différée ou sauvegarde chez un autre prestataire.

Intégrations

Dans le meilleur des cas : les données saisies une seule fois sont transmises à la comptabilité, aux clients, aux transporteurs et aux machines.

Dans le pire des cas : une facture créée en double, un avis d'expédition envoyé deux fois, des écarts silencieux entre l'ERP et le logiciel comptable, et des équipes qui ressaisissent les données à la main.

Verdict

Aujourd'hui

les agents écrivent facilement un connecteur. La difficulté consiste à décider ce qui se passe lorsqu'un message est perdu, reçu plusieurs fois ou en retard, puis à appliquer ces règles de manière uniforme. Rien de tout cela ne figure dans une simple demande de connexion.

De 2027 à 2030

le nombre de systèmes à connecter augmente, et chaque tiers continue de faire évoluer son interface. Ce travail de conception s'accroît donc, quelles que soient les capacités de génération de code des modèles. Une entreprise sans équipe technique qui développe son ERP à partir de zéro obtiendra des connexions fonctionnelles au premier jour. Lors des premières défaillances, comme un délai d'attente dépassé pendant l'export d'une facture ou un message de transporteur reçu deux fois, elle découvrira des doublons et des écarts entre l'ERP et le logiciel comptable. Personne ne saura quel système sert de référence.

Éléments de preuve

L'intégration relève des systèmes distribués, dont les difficultés sont bien documentées.

- En 1994, Peter Deutsch a recensé les hypothèses erronées des ingénieurs sur les réseaux, connues sous le nom d'illusions de l'informatique distribuée. La première consiste à croire que le réseau est fiable, alors que des messages se perdent en pratique (Rotem-Gal-Oz). Lorsqu'un système envoie une requête sans recevoir de réponse, il ne peut pas savoir si elle a été traitée.
- Aucun nombre fini de messages ne permet à deux systèmes d'être certains de leur accord. Les systèmes réels contournent cette limite en rendant les opérations répétables sans effet supplémentaire et en vérifiant ensuite leur statut (Kleppmann, notes de cours de Cambridge). Un message renvoyé jusqu'à accusé de réception peut être appliqué deux fois. Le cours prend l'exemple d'un débit de 1 000 £ prélevé deux fois.
- Stripe a créé des clés d'idempotence pour éviter qu'un paiement relancé soit débité deux fois. La clé doit être vérifiée par rapport à la requête initiale et n'est conservée que pour une durée limitée (Stripe, documentation de l'API). Bien concevoir ces mécanismes exige de l'expertise : déterminer ce qui rend deux requêtes identiques, combien de temps les mémoriser et comment traiter leurs différences. Rien de cela ne figure dans une demande comme « connectez-nous à notre logiciel comptable ». Garantir qu'un message est transmis exactement une fois exige la coopération des deux systèmes (Confluent, 2017).

- À titre de contexte, les grandes entreprises utilisent en moyenne 957 applications, dont 27 % sont intégrées (MuleSoft 2026). Leurs développeurs consacraient 39 % de leur temps à créer de nouvelles intégrations en 2025 (MuleSoft 2025).
- Les échecs d'intégration et de migration ont des coûts mesurés. La migration de plateforme de TSB en 2018 a entraîné 225 492 réclamations, 330,2 millions de livres sterling de coûts après migration et 48,65 millions de livres d'amendes (décision finale de la FCA, résultats de TSB). Le lancement de l'ERP de Revlon dans une usine en 2018 a empêché des expéditions représentant environ 64 millions de dollars de ventes nettes et entraîné 53,6 millions de dollars de charges supplémentaires (Revlon 10-K). Aucun de ces échecs n'impliquait l'IA. Tous deux tenaient à la planification, aux tests et au déploiement de la migration. Il s'agit d'ingénierie et de gestion de projet, que la génération plus rapide de code ne remplace pas.

Générer un connecteur est facile. Définir le comportement de chaque connexion en cas de message perdu, répété ou tardif, puis appliquer cette règle de façon uniforme, relève d'un travail d'ingénierie en amont du code. Les chiffres de MuleSoft concernent des entreprises de plus de 1 000 salariés, pas de petits industriels.

Conformité

Dans le meilleur des cas : les factures sont acceptées, les contrôles fiscaux se passent bien et les produits sont traçables.

Dans le pire des cas : des factures rejetées, des pénalités et des rappels dont le périmètre ne peut pas être établi.

Verdict

Aujourd'hui

la plupart des obligations d'un ERP, comme la piste d'audit, l'export comptable, la traçabilité, la conservation ou les blocages qualité, ne disposent d'aucun outil de vérification automatique. Sans cet outil, les agents se trompent souvent : 44 à 76 % de réponses correctes sur des règles exprimées sous forme de texte, contre près de 100 % lorsqu'une suite de tests officielle existe.

De 2027 à 2030

les agents deviendront fiables dans les domaines dotés de validateurs officiels, comme les formats de facture électronique (Vericoding, 2025 montre une progression de 68 % à 96 % de code vérifié en un an). Ce ne sera pas le cas ailleurs, et de nouvelles obligations continuent d'apparaître : facturation électronique des PME françaises et passeports de batterie en 2027, passeports numériques de produits dès 2026. Une entreprise sans équipe technique qui développe son ERP à partir de zéro aura un système apparemment conforme, jusqu'à ce qu'un contrôle fiscal, une facture rejetée ou un rappel démontre le contraire. Chaque nouvelle obligation imposera de relire la réglementation et de la traduire en contrôles, sans personne dans l'entreprise qui ait déjà fait ce travail.

Éléments de preuve

Une entreprise industrielle en Europe ou aux États-Unis doit respecter un nombre croissant d'obligations que son ERP doit prendre en charge.

- **États-Unis :** depuis l'arrêt South Dakota v. Wayfair de 2018, les 45 États appliquant une taxe sur les ventes imposent les vendeurs à distance au-delà d'un seuil de ventes. L'ERP doit donc calculer la taxe selon la destination (Cour suprême). Les documents relatifs aux biens soumis au contrôle des exportations sont conservés cinq ans (ITAR, EAR). Les fournisseurs de la défense détenant des informations non classifiées contrôlées doivent respecter les 110 exigences du NIST SP 800-171 dans le cadre de la Cybersecurity Maturity Model Certification (CMMC), dont la première phase a commencé le 10 novembre 2025 (32 CFR 170).
- **Agroalimentaire aux États-Unis :** la règle FSMA 204 exige un code de lot de traçabilité à chaque événement critique de suivi et un fichier électronique pouvant être trié dans les 24 heures suivant une demande de la FDA. Son application est prévue au plus tôt le 20 juillet 2028 (21 CFR 1.1455, Public Law 119-37).
- **Dispositifs médicaux :** depuis le 2 février 2026, la réglementation de la FDA sur les systèmes qualité intègre l'ISO 13485 et exige l'enregistrement de l'identifiant unique pour chaque dispositif ou lot (FDA, 21 CFR 820.10). Le logiciel utilisé dans le système qualité doit lui-même être validé (21 CFR 11.10, FDA Computer Software Assurance).
- **Aéronautique et défense :** l'AS9100D impose la gestion de configuration, la prévention des pièces contrefaites et la traçabilité (IAQG). Les règles américaines de défense exigent aussi de retracer les composants électroniques jusqu'à leur fabricant d'origine et de mettre les pièces suspectes en quarantaine (DFARS 252.246-7007).

- **Automobile :** l'IATF 16949 exige des plans de traçabilité documentés permettant d'identifier et d'isoler les produits suspects dans les délais de réponse de chaque constructeur. S'y ajoutent les exigences spécifiques publiées par BMW, Ford, Renault, Stellantis, Volkswagen et d'autres (IATF).
- **Pharmacie :** l'annexe 11 des BPF européennes impose des systèmes validés, un contrôle des accès et une piste d'audit enregistrant le motif de chaque modification. Elle demande aussi un processus qualité formalisé pour les systèmes développés sur mesure ou personnalisés (annexe 11). Dans l'UE, les boîtes de médicaments portent un identifiant unique comprenant le code produit, le numéro de série, le lot et la date de péremption (règlement 2016/161). Aux États-Unis, la loi exige une traçabilité électronique à la boîte depuis 2025 (FDA, DSCSA).
- **Facturation électronique :** toutes les entreprises françaises doivent pouvoir recevoir des factures électroniques depuis le 1er septembre 2026, et les PME doivent en émettre à partir du 1er septembre 2027 (impots.gouv.fr). Les outils de validation EN 16931 comprennent 223 règles métier, plus environ 759 règles propres au format UBL et 587 au format CII, révisées environ deux fois par an (CEN/TC 434). En Italie, où la facturation électronique est obligatoire depuis 2019, 1,4 % des 2,1 milliards de factures étaient encore rejetées en 2021 (Corte dei conti).
- **Traçabilité :** les entreprises agroalimentaires doivent identifier le fournisseur et le client immédiats de chaque lot (règlement 178/2002, art. 18) et signaler 14 groupes d'allergènes (règlement 1169/2011). Les fabricants de médicaments et de dispositifs médicaux vendant aux États-Unis doivent disposer de pistes d'audit horodatées qui ne masquent jamais les valeurs précédentes (21 CFR 11.10).
- **Contrôles fiscaux :** les entreprises françaises doivent remettre leurs écritures comptables dans un fichier comprenant 18 champs dans un ordre fixe, avec une numérotation continue et chronologique (LPF, art. A47 A-1), sous peine de 5 000 € d'amende ou de 10 % des droits rappelés. Une piste d'audit fiable doit relier les factures aux livraisons correspondantes (BOFiP).
- **Conservation :** les documents comptables sont conservés dix ans (Code de commerce, art. L123-22). Cette obligation prévaut sur le droit à l'effacement du RGPD pendant cette période, puis impose leur suppression.
- **Nouvelles obligations de données :** un passeport de batterie à partir du 18 février 2027 (Commission européenne) et des passeports numériques de produits pour l'acier, les textiles, les pneumatiques et l'aluminium de 2026 à 2027 (ESPR).

La capacité des agents à appliquer correctement une norme dépend de l'existence d'un outil de vérification automatique.

- Lorsqu'une suite de tests complète ou un validateur existe, les agents atteignent désormais une conformité presque totale. Une équipe de 16 agents Claude Opus 4.6 a créé un compilateur C qui passe 99 % des suites de tests standard pour compilateurs. Son auteur souligne que le résultat dépend d'un vérificateur presque parfait (Anthropic, février 2026). Un analyseur HTML5 porté avec GPT-5.2 en environ 4,5 heures a réussi 6 810 tests de tokenisation sur 6 810 (Willison, décembre 2025). Pour transformer une note clinique en dossier de santé FHIR, les modèles obtenaient au mieux 50 % seuls et jusqu'à 100 % avec accès au validateur officiel (Flexpa, 2025, 14 cas).

- Lorsque la norme n'existe que sous forme de texte, les résultats sont bien inférieurs. Sur 100 cas fiscaux vérifiés au dollar près, GPT-5 répondait correctement 76 fois (Jurayj et al., 2026). Sur 6 232 tâches d'application de règles en droit fiscal, de l'immigration et du logement, le meilleur résultat sur le sous-ensemble difficile est de 44 % (DeonticBench, avril 2026). Les modèles de pointe calculaient correctement moins d'un tiers des déclarations fiscales fédérales américaines, même avec toutes les données fournies (TaxCalcBench, 2025). Le meilleur modèle attribue les bons codes de facturation médicale dans 64 % des cas (Vals.ai MedCode). Aucun des cinq chatbots testés n'a produit de code conforme à la norme de sécurité MISRA C++, malgré des prompts explicites (Umer, IEEE Access 2025).
- Réussir les contrôles ne signifie pas avoir raison. La vérification formelle prouve que le code respecte la spécification fournie, pas que cette spécification reflète la loi (Vericoding, 2025).

Certaines obligations de conformité d'un ERP peuvent être vérifiées par des validateurs officiels, comme les règles EN 16931 pour la facturation électronique. Les agents peuvent donc plus facilement les respecter, même si aucune étude ne mesure encore les factures électroniques générées par un agent. La facturation électronique pose aussi un problème d'intégration et de systèmes distribués : les factures transitent par des plateformes d'échange agréées, avec des accusés de réception, des rejets et des mises à jour de statut à traiter. Elle cumule donc les deux difficultés. Une grande partie des obligations n'a aucune suite de tests : piste d'audit fiable, contenu et signification d'un export comptable, traçabilité des lots, durées de conservation. Quelqu'un doit alors traduire le texte en contrôles et les maintenir à jour face à des règles qui évoluent chaque année.

Maintenance

Dans le meilleur des cas : le système évolue avec l'entreprise.

Dans le pire des cas : l'essentiel du budget sert à maintenir le système en fonctionnement, et la connaissance de son fonctionnement se perd.

Verdict

Aujourd'hui

le volume de code écrit avec l'IA augmente plus vite, avec davantage de duplications et une complexité accrue. Les personnes qui apprennent avec l'IA le comprennent moins bien.

De 2027 à 2030

les agents prendront en charge davantage de corrections courantes. Les données ne montrent pas que le code devient plus facile à comprendre, alors que cette compréhension est indispensable à la maintenance. Une entreprise sans équipe technique qui développe son ERP à partir de zéro dépendra de l'agent pour chaque correction. Chacune ajoutera du code, souvent copié depuis une autre partie du système. Après quelques années, personne dans l'entreprise ne comprendra le fonctionnement d'ensemble.

Éléments de preuve

- La maintenance représente 40 à 80 % du coût d'un logiciel sur sa durée de vie (Robert Glass, *Facts and Fallacies of Software Engineering*, 2002, fait 41). Les développeurs estiment y consacrer 42 % de leur temps (Stripe, 2018). Ces chiffres sont antérieurs aux agents de programmation.
- GitClear a analysé 623 millions de modifications de code. La part des modifications améliorant le code existant est passée de 21 % en 2022 à 3,8 % en 2026, tandis que celle du code copié-collé passait de 9,4 % à 15,7 % (GitClear 2026, données d'un fournisseur ne comparant pas utilisateurs et non-utilisateurs de l'IA). Lorsqu'un même code existe à plusieurs endroits, un bug corrigé à un endroit persiste souvent ailleurs.
- Des chercheurs ont comparé 806 projets open source ayant adopté l'éditeur IA Cursor à 1 380 projets similaires ne l'ayant pas adopté. Dans les premiers, la production de code augmentait fortement le premier mois, puis retrouvait son rythme habituel. Leur code devenait cependant plus complexe et déclenchait davantage d'alertes des outils automatiques de qualité, sans amélioration ultérieure (Carnegie Mellon, MSR 2026).
- Dans une étude contrôlée avec 151 participants, le code écrit avec l'IA n'était pas plus difficile à modifier ensuite par d'autres développeurs (Echoes of AI, 2026).
- Dans un essai contrôlé avec 52 ingénieurs principalement juniors, ceux qui avaient réalisé une tâche de programmation avec l'IA obtenaient 50 % à un questionnaire de compréhension ultérieur, contre 67 % pour ceux qui avaient codé seuls. L'écart le plus marqué concernait le débogage (Anthropic, janvier 2026).

Aucune étude ne mesure encore la part de maintenance d'un système créé par des agents. Dans plusieurs jeux de données, les travaux ci-dessus relèvent une hausse du volume de code, des duplications et de la complexité après l'adoption de l'IA. Une expérience contrôlée ne constate toutefois aucun effet sur la facilité de modification du code. D'autres travaux montrent des compétences de débogage plus faibles chez ceux qui s'appuient sur l'IA pour apprendre.

Coût

Dans le meilleur des cas : développer coûte moins cher que jamais.

Dans le pire des cas : l'essentiel des dépenses intervient après le développement : maintenance, équipes chargées de vérifier le travail des agents et utilisation des modèles de pointe, plus coûteuse à chaque génération.

Verdict

Aujourd'hui

le coût d'un ERP provient surtout du travail humain : le développer, puis le maintenir sur toute sa durée de vie, porte le total à 1,7 à 5 fois son coût de développement. Les modèles n'en représentent qu'une petite part. Le modèle le moins cher atteignant un score donné sur un benchmark voit son coût divisé par environ 13 chaque année. Ce chiffre est amplifié par l'entraînement des modèles sur ces benchmarks. En parallèle, chaque tâche de programmation réalisée avec le meilleur modèle coûte plus cher qu'il y a un an.

De 2027 à 2030

un ERP est développé et maintenu avec le meilleur modèle de son époque. Le coût des modèles suit donc une courbe ascendante. À cela s'ajoute la rémunération des personnes chargées de vérifier le travail. Une entreprise sans équipe technique qui développe son ERP à partir de zéro constatera un faible coût initial, puis un coût d'exploitation qui augmente avec chaque nouveau modèle et chaque correction, et qu'elle assume seule. L'option la moins chère dépendra surtout du nombre d'utilisateurs et de l'existence d'une équipe de développement : l'achat préserve la trésorerie et se paie par utilisateur ; le développement mobilise durablement des fonds et des salariés, mais peut coûter moins cher avec de nombreux utilisateurs.

Éléments de preuve

Sur toute la durée de vie d'un ERP, le travail humain représente l'essentiel du coût. Les modèles n'en constituent qu'une petite part.

- La maintenance représente 40 à 80 % du coût total d'un système (Glass, voir Maintenance). Un système développé en interne coûte donc 1,7 à 5 fois son coût de développement sur sa durée de vie. Les agents réduisent surtout la phase de développement, et les données montrent une baisse moindre de la maintenance.
- Le développement se paie surtout en salaires. Le salaire médian d'un développeur aux États-Unis était de 135 980 $ en mai 2025 (BLS). Les salaires représentent 68,5 % du coût employeur d'un salarié à temps plein du secteur privé (BLS, juin 2026). Un développeur coûte donc environ 200 000 $ par an. À 13 $ par jour d'utilisation, un agent de programmation IA ajoute environ 3 000 $ par an et par développeur.
- L'achat se paie principalement par un abonnement par utilisateur, auquel s'ajoute la mise en place de la plateforme. Plus d'un quart des entreprises dépassent le budget de cette mise en place (Panorama Consulting, 2026).

Les deux options n'utilisent pas la trésorerie de la même façon. L'achat demande peu de fonds au départ et transforme l'ERP en coût récurrent par utilisateur. Le développement exige immédiatement de la trésorerie pour les salaires, puis des charges salariales fixes tant que le système fonctionne. Selon les IFRS, les coûts de développement d'un logiciel interne peuvent être immobilisés dès que le projet remplit les conditions d'IAS 38. La configuration d'un logiciel par abonnement est généralement comptabilisée en charges au fur et à mesure des travaux (IFRIC, 2021). Les normes US GAAP permettent d'immobiliser certains coûts de mise en œuvre des logiciels cloud (ASU 2018-15).

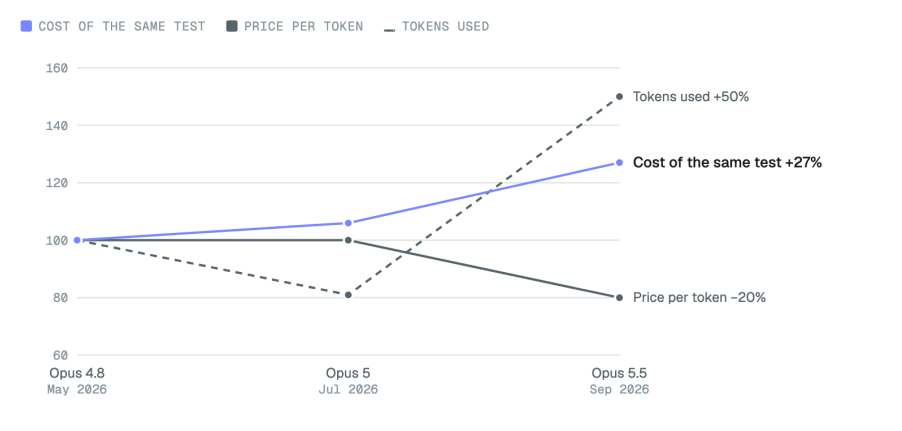
L'option la moins coûteuse dépend surtout du nombre d'utilisateurs : le coût de l'abonnement augmente avec le nombre d'utilisateurs, celui du système interne avec la taille de l'équipe. À titre d'illustration, si un développeur pouvait créer et maintenir seul l'ERP, l'abonnement annuel coûterait autant que ce développeur à environ 330 utilisateurs pour 50 $ par utilisateur et par mois, 170 utilisateurs à 100 $ et 80 utilisateurs à 200 $. Cette illustration exclut le coût de mise en place de la plateforme d'un côté et la probabilité qu'un seul développeur ne suffise pas de l'autre.



La courbe bleue représente le coût d'une tâche de programmation avec le meilleur modèle disponible à chaque date (données MIT FutureTech sur SWE-bench Verified). La courbe verte représente le coût d'une question scientifique à niveau de qualité constant, celui du meilleur modèle de janvier 2025 (Epoch AI), mesuré sur un benchmark public sur lequel les laboratoires peuvent entraîner leurs modèles. L'échelle verticale est logarithmique : chaque graduation vaut 10 fois la précédente.

Le coût des modèles s'ajoute à ces dépenses et suit deux tendances simultanées.

- Le prix d'un niveau de capacité donné baisse d'environ 47 % par trimestre, soit une division par environ 13 par an (Epoch AI, septembre 2026).
  - Répondre à une question scientifique au niveau d'o3 coûtait 0,30 $ en janvier 2025 et 0,0004 $ à la mi-2026. Les auteurs reconnaissent que ce chiffre peut être influencé par l'optimisation des modèles pour les benchmarks. Il correspond au coût du modèle le moins cher atteignant un score donné, et les laboratoires peuvent entraîner leurs modèles sur certains de ces tests. Sur le seul dont le contenu reste secret, le prix baisse moins vite.
  - Le modèle le moins cher atteignant un score est rarement un modèle de pointe. Lorsqu'un test public de mathématiques a été réécrit avec de nouvelles questions de même difficulté, plusieurs familles de modèles ont montré un surapprentissage systématique, contrairement aux modèles de pointe qui en montraient peu (GSM1k, 2024). Le calcul suppose aussi que l'utilisateur passe au modèle le moins cher chaque trimestre. Ce chiffre montre que le coût nécessaire pour atteindre un score de benchmark baisse, pas que le même travail coûte moins cher.
- Le coût du meilleur modèle disponible augmente sur les tâches de programmation, car chaque génération utilise beaucoup plus de tokens par tâche. Une exécution de SWE-bench Verified avec le meilleur modèle coûtait 88 $ en octobre 2024 et 2 323 $ en février 2026 (MIT FutureTech, données). Les réponses des modèles de raisonnement deviennent environ cinq fois plus longues chaque année (Epoch AI). Sur les questions scientifiques, le coût du meilleur modèle a culminé en décembre 2024, puis s'est maintenu autour de 0,07 à 0,10 $ par question.



Modèles Claude Opus successifs en 2026, base 100 pour Opus 4.8 : coût d'exécution du même test (Artificial Analysis Intelligence Index v4.3.2, effort maximal), prix par token de sortie et nombre de tokens utilisés pour ce test.

Entre Claude Opus 4.8 et Opus 5.5, le prix par token a baissé de 20 %, le nombre de tokens utilisés a augmenté de 50 % et le coût du même test a augmenté de 27 % (Artificial Analysis). Anthropic estime le coût de Claude Code à environ 13 $ par développeur et par jour d'utilisation (Claude Code).

Les progrès des modèles réduisent le coût nécessaire pour atteindre les scores de benchmark d'hier, tandis que l'utilisation des modèles de pointe coûte plus cher par tâche. La seconde tendance est la plus solide : elle mesure ce que les modèles consomment, plutôt que leurs scores. Un ERP a besoin du modèle le plus fiable disponible, car ses erreurs coûtent cher.

Adaptabilité

Dans le meilleur des cas : le logiciel prend en charge les processus de l'entreprise au lieu d'imposer des processus génériques.

Dans le pire des cas : les processus sont déformés pour s'adapter au logiciel, ou les modifications du cœur partagé provoquent des régressions et exposent des données.

Verdict

Aujourd'hui

un ERP développé sur mesure correspond exactement à l'entreprise. Les plateformes diffèrent : certaines ne peuvent être adaptées qu'en modifiant le code de leur cœur, ce qui impose de reprendre ces modifications à chaque montée de version. D'autres, comme Bonx, permettent de créer des adaptations sur les données propres à l'entreprise et de les conserver après les mises à niveau.

De 2027 à 2030

l'IA réduit le coût des adaptations dans les deux cas. Les plateformes conçues pour être personnalisées en toute sécurité se rapprochent ainsi du sur-mesure. Une entreprise sans équipe technique qui développe son ERP à partir de zéro obtient un système parfaitement adapté à ses processus au premier jour. Qu'il le soit encore cinq ans plus tard dépend des autres critères : une architecture évolutive, des règles qui restent correctes et quelqu'un capable d'assurer la maintenance.

Éléments de preuve

- Dans une enquête de 2013 auprès d'utilisateurs d'ERP, 38 % avaient modifié leurs processus pour s'adapter au logiciel, et 29 % avaient modifié le logiciel pour conserver leurs processus (Panorama 2013).
- Aucune étude publiée ne mesure la part des besoins d'une entreprise couverte par les fonctions standard d'un ERP. Le chiffre souvent cité de 70 % n'a pas de source identifiable.
- Odoo, un ERP open source largement utilisé, peut être adapté de deux façons :
  - Odoo Studio ajoute des champs, des vues, des modèles, des automatismes et des rapports sans code (Odoo Studio).
  - Au-delà, les adaptations passent par des modules Python qui étendent le cœur.
  - Odoo publie une version majeure par an et impose une mise à niveau au moins tous les deux ans. Le responsable de chaque module spécifique doit le rendre compatible avec la nouvelle version, sauf si le client paie Odoo un forfait mensuel par tranche de 100 lignes de code spécifique pour ce travail (documentation des mises à niveau Odoo, contrat Odoo Enterprise).

Un ERP développé sur mesure correspond par construction à l'entreprise : c'est son principal avantage. Sur une plateforme, l'adaptabilité dépend de mécanismes permettant d'ajouter ses propres champs, tables, écrans et règles sur ses données sans toucher au cœur partagé. Des droits d'accès et une isolation adéquats doivent empêcher qu'une adaptation perturbe d'autres fonctions ou expose des données.

L'IA réduit le coût de création de ces adaptations. Leur sécurité vient des mécanismes de la plateforme, pas du modèle. Leur mode de réalisation détermine aussi leur coût à long terme. Lorsqu'elles prennent la forme de code dans le cœur de la plateforme, chacune impose du travail à chaque mise à niveau, comme le montrent les conditions d'Odoo. Lorsqu'elles sont réalisées sur les données de l'entreprise avec des mécanismes maintenus par la plateforme, elles sont conservées sans réécriture.

Nouvelles fonctionnalités

Dans le meilleur des cas : le logiciel exécute une partie du travail au lieu de seulement l'enregistrer. Par exemple : des plannings recalculés lorsqu'une commande change, des documents fournisseurs lus et leurs données saisies sans intervention manuelle, des tâches administratives courantes réalisées sans intervention humaine.

Dans le pire des cas : une entreprise qui développe seule reste limitée à ce que son équipe sait imaginer et vérifier. Cela revient généralement à recréer un ERP standard pendant que les autres avancent.

Verdict

Aujourd'hui

les ingénieurs expérimentés bénéficient de la quasi-totalité des gains de productivité liés à l'IA et des progrès qu'elle permet dans l'exploration de nouveaux domaines. Évaluer ses résultats exige de l'expertise. Dans l'industrie, l'IA a produit des gains après une baisse initiale, et seulement avec un travail d'organisation autour de son adoption.

De 2027 à 2030

tout le monde disposera des mêmes modèles. La différence viendra de ce qui sera construit avec eux. Comparons deux industriels semblables pour illustrer ce point. Il s'agit d'un scénario fondé sur les données de cette analyse, pas d'une mesure.

Période	Développement avec des agents, sans équipe d'ingénierie	Achat d'une plateforme adaptable
Premiers mois	Premiers écrans en quelques semaines : les agents parviennent presque toujours à créer une nouvelle application.	Départ avec des fonctions déjà utilisées et testées par d'autres industriels, dans le cadre d'un projet de mise en place. Plus d'un quart de ces projets dépassent leur budget : ils méritent donc autant de soin qu'un développement.
Années 1 à 2	Chaque modification risque de provoquer une régression : 34 % de réussite pour faire évoluer une application, et une régression dans 39 % des échecs. Les efforts se concentrent sur les corrections.	La plateforme est adaptée aux processus sans toucher au cœur. L'éditeur fournit les correctifs, les mises à jour de sécurité, les évolutions réglementaires et les nouvelles fonctions créées par une équipe qui exploite mieux les mêmes modèles.
Année 5	La maintenance absorbe l'essentiel du budget, soit 40 à 80 %. Le système fait ce que l'entreprise savait spécifier en première année.	L'ERP a continué de progresser sans que l'entreprise le développe ou le maintienne. Son équipe se concentre sur ses processus. Elle dépend de l'éditeur pour les mises à niveau, les tarifs et l'assistance : le choix de l'éditeur devient donc la décision principale.

Verdict

Projection : l'écart est le plus grand si la plateforme est réellement native IA, avec une IA qui exécute seule l'essentiel des tâches administratives. Aucune étude publiée ne l'a encore mesuré dans un véritable ERP. La projection ci-dessous repose donc sur une hypothèse : l'IA de l'ERP réalise 80 % du travail administratif, et ses propres contrôles remplacent l'essentiel de la vérification humaine. C'est une hypothèse forte. Avec les modèles de 2025, vérifier et reprendre le travail de l'IA annulait la majeure partie du temps gagné (GDPval, voir l'hypothèse centrale).

Dans l'industrie manufacturière américaine, les métiers de gestion, de finance, de bureau et d'administration représentaient 12,6 % des salariés et 90 milliards de dollars de salaires en 2021 (BLS), pour une valeur totale des livraisons de 6 100 milliards de dollars (Census), soit environ 500 000 $ par salarié. Charges employeur comprises, les fonctions administratives coûtent environ 2 % du chiffre d'affaires.

Prenons un industriel de 100 salariés, réalisant 50 millions de dollars de chiffre d'affaires avec une marge opérationnelle de 8 %. Cette marge se situe dans la fourchette observée chez les industriels américains cotés, de 4 % dans l'acier à 16 % dans les machines (Damodaran, janvier 2026). Ses fonctions administratives comptent environ 13 personnes et coûtent environ 1,1 million de dollars par an. Nous supposons que l'entreprise qui développe paie un développeur et n'atteint pas le même niveau d'automatisation, car elle consacre ses efforts aux fonctions standard. Celle qui achète paie, par hypothèse, 100 $ par utilisateur et par mois pour 30 utilisateurs.

Indicateur	Développement à partir de zéro	Achat d'un ERP natif IA
Coût des fonctions administratives	Environ 1,1 million de dollars	Environ 200 000 $, soit l'équivalent de 10 personnes réaffectées ou non remplacées
Coût de l'ERP	Environ 200 000 $ pour un développeur	Environ 36 000 $ d'abonnement
Résultat opérationnel	Environ 3,8 millions de dollars	Environ 4,8 millions de dollars
Marge opérationnelle	Environ 7,6 %	Environ 9,7 %

Avec ces hypothèses, l'entreprise qui achète réalise chaque année un résultat opérationnel supérieur d'environ un quart à celui de l'entreprise qui développe. L'écart atteint environ 55 % si toutes deux partent d'une marge de 4 %. La projection exclut les coûts de mise en place des deux côtés et tout effet sur le chiffre d'affaires, comme des devis plus rapides ou moins de livraisons en retard, qu'aucune étude n'a encore mesuré. Les données industrielles présentées ci-dessous suggèrent aussi que les gains apparaissent après une période d'adaptation.

Éléments de preuve

Tout le monde a accès aux mêmes modèles. Ce qui diffère, c'est ce qui est créé avec eux, et par qui.

- Sur 160 097 développeurs sur GitHub, l'IA écrit environ 29 % des fonctions Python aux États-Unis. Les développeurs qui utilisent l'IA explorent aussi plus facilement de nouveaux domaines logiciels. La quasi-totalité des gains de productivité et des progrès dans l'exploration de nouveaux domaines concerne les développeurs expérimentés, tandis qu'aucun gain significatif n'apparaît chez ceux en début de carrière (Daniotti et al., Science 2026).
- Des essais contrôlés antérieurs sur des tâches courantes vont dans l'autre sens. Chez Microsoft, Accenture et une entreprise du Fortune 100, les développeurs moins expérimentés adoptaient davantage Copilot et en tiraient plus de gains, sans que l'écart soit statistiquement significatif (Cui, Demirer et al.). Dans le support client, les débutants gagnaient 34 %, contre 14 % en moyenne (Brynjolfsson et al., NBER 2023).
- Évaluer les résultats de l'IA exige de l'expertise. Les responsables de projets refuseraient environ la moitié des correctifs d'agents passant les tests automatiques (METR, mars 2026). Corriger une modification d'agent prenait en moyenne 26 à 42 minutes (METR, août 2025). Les modifications soumises par des développeurs moins expérimentés travaillant avec l'IA recevaient 4,52 fois plus de commentaires de revue et étaient acceptées 31 % moins souvent (étude de 22 953 demandes d'intégration de code, 2026).
- Dans l'industrie américaine, les usines ayant adopté l'IA industrielle ont d'abord perdu en productivité, avec davantage d'encours de production, des investissements dans les robots et des réductions d'effectifs, avant de croître plus vite que les autres. Les pertes se concentraient dans les entreprises les plus anciennes et étaient atténuées par des stratégies de croissance (McElheran, Yang, Kroff et Brynjolfsson, document de travail, 2025). L'IA n'améliore les performances d'une usine qu'avec le travail d'organisation qui l'accompagne.
- Le rapport DORA 2025 décrit l'IA comme un amplificateur des forces et des faiblesses existantes d'une organisation (DORA 2025).

Les résultats diffèrent selon ce qui est mesuré. Les débutants progressent davantage sur les tâches courtes et courantes. Les experts gagnent davantage sur le travail réel, explorent de nouveaux domaines et savent vérifier le résultat. Créer des fonctions qui n'existaient pas auparavant relève de ce second type de travail.

Ce qui se passe avec le temps si vous développez en interne

Si vous décidez de développer, le tableau suivant montre comment la situation évolue généralement des premiers mois à la cinquième année. Il présente les compromis à prévoir et les problèmes à anticiper dès la conception.

Critère	Premiers mois	Après 1 à 2 ans	Après 5 ans
Conception technique	Livraison rapide des fonctionnalités	Évolutions de plus en plus lentes et risquées	Réécriture partielle ou totale
Règles métier	Petites erreurs de totaux et de stock	Marges et valeurs de stock erronées en comptabilité	États financiers erronés, rappels impossibles à délimiter
Sécurité	Données ou clés exposées	Intrusion ou rançongiciel	Production arrêtée pendant des semaines après un seul incident
Performance	Rapidité sur les données de test	Écrans lents, traitements nocturnes trop longs	Refonte de l'architecture
Fiabilité & protection des données	Pannes lors des mises en production	Perte de données lors d'un incident	Lacunes dans l'historique des stocks et des commandes
Intégrations	Ressaisies manuelles et doublons	Écarts silencieux avec la comptabilité et les clients	Chaque nouvelle connexion coûte autant que la première
Conformité	Factures rejetées	Pénalités lors d'un contrôle fiscal	Nouvelles obligations non mises en œuvre
Maintenance	Peu de problèmes	Effort consacré surtout aux corrections	Système que personne ne comprend
Coût	Faible coût de développement	Coût d'exploitation supérieur au développement	Coûts d'exploitation majoritaires dans le coût total
Adaptabilité	Contournements dans des tableurs	Processus déformés pour s'adapter au logiciel	Logiciel contourné ou abandonné
Nouvelles fonctionnalités	Fonctions standard recréées une par une	Aucune fonction au-delà de ce qui a été spécifié	Retard sur les concurrents dont l'ERP a continué de progresser

Ce que cela implique pour vous

Les modèles continueront de réduire le coût du code. Ils ne remplacent pas l'expérience de l'exploitation d'un ERP sur plusieurs années : maintenir la cohérence de l'architecture, appliquer correctement les règles métier et faire fonctionner le système. La question est donc moins de savoir si l'IA peut écrire votre ERP que de savoir qui en sera responsable, qui le testera et qui le fera évoluer une fois créé.

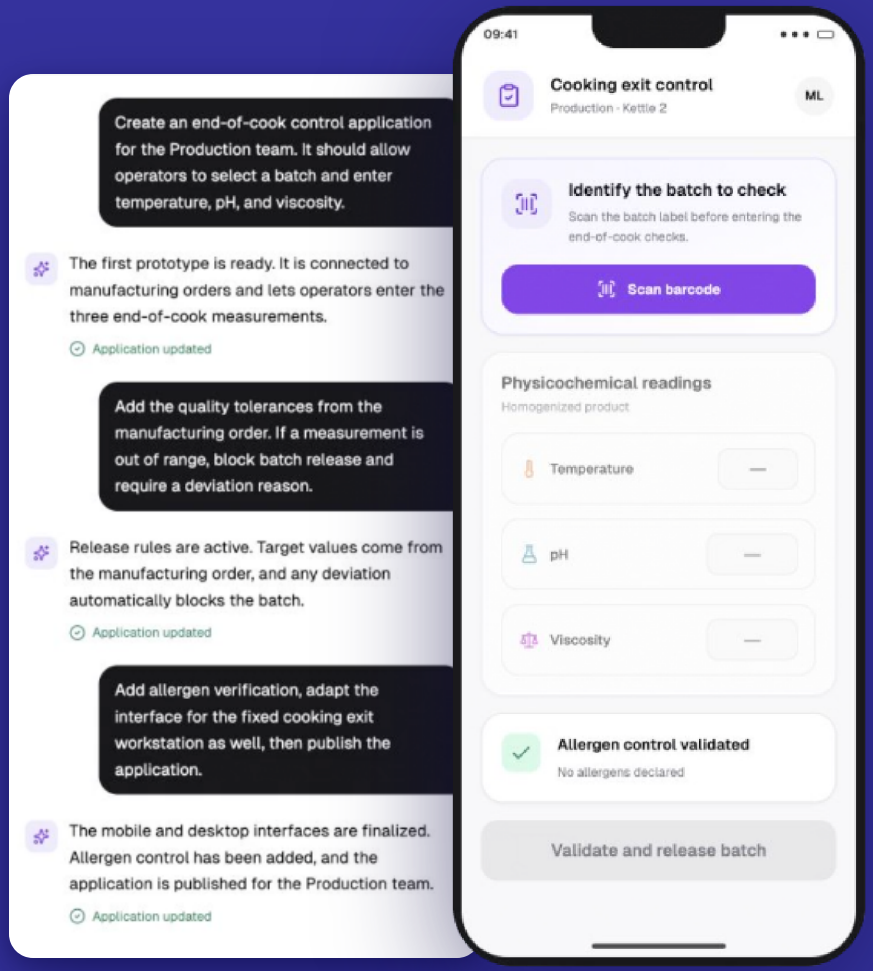
À retenir

- L'IA simplifie les tâches faciles, mais pas les plus difficiles : cohérence d'une architecture sur des années d'évolution, règles métier non écrites, exploitation, intégrations et conformité.
- Les équipes expérimentées tirent davantage parti de l'IA. L'écart entre une équipe qui exploite des ERP depuis des années et une entreprise qui crée son premier ERP tend donc à se creuser.
- Développer se justifie lorsque la plupart de vos besoins sont spécifiques. Dans les autres cas, achetez une plateforme adaptable et développez sur cette base les fonctions spécifiques dont vous avez besoin.
- Utilisez les 11 critères ci-dessus pour évaluer votre situation et anticipez les problèmes du tableau avant qu'ils n'apparaissent.

Bonx

Demander une démo personnalisée

Achetez la plateforme. Développez la part qui vous est propre.



Pour la plupart des industriels, les recherches mènent à une même réponse : acheter une plateforme adaptable, puis développer dessus, avec l'IA, la partie qui vous est propre. Bonx est l'ERP industriel natif IA, conçu précisément pour cela.

Il s'adapte à votre usine, pas l'inverse

Vos responsables modifient eux-mêmes le modèle de données, les processus, les automatismes, les agents IA et les écrans, sans code et sans consultant. Environ 90 % des opérations reposent sur des modules standard. Le reste peut être développé sous forme d'application dans Bonx, sur les mêmes données.

Il se déploie en semaines, pas en années

Une première mise en production en 10 semaines en moyenne et une mise en production complète en 4 à 6 mois, contre 12 à 24 mois pour un ERP traditionnel. Comment Bonx se déploie bien plus vite qu'un ERP traditionnel

Il s'intègre à votre environnement existant

Un seul système pour les commandes, les stocks, les achats, la planification, la production, la qualité, la logistique et la facturation, connecté à votre CRM, à votre e-commerce et à vos outils de comptabilité. Nervures a réduit de 95 % les ressaisies manuelles de données dans la production et la logistique.



« Bonx est à la fois extrêmement complet et remarquablement facile à personnaliser. Nous l'avons adapté précisément à nos processus industriels sans ajouter de complexité. »

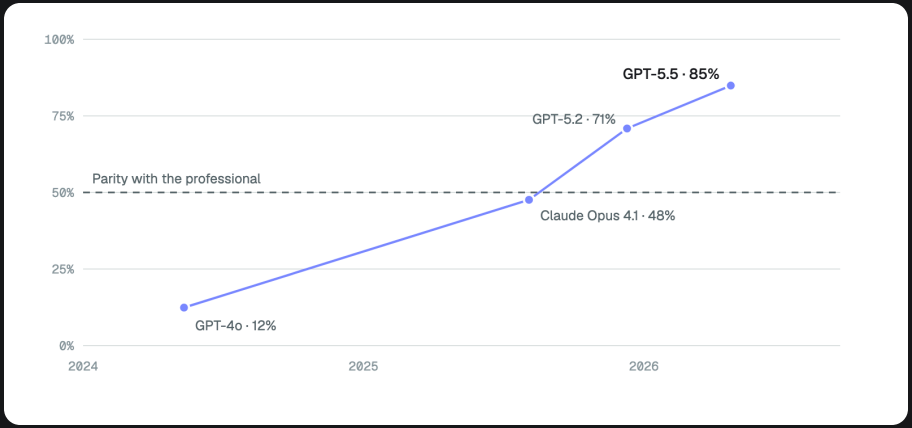
Romaric Duretz, directeur d'usine de La Maillecotech

Hypothèse centrale, limites & transparence

L’analyse suppose que les grands modèles de langage (LLM) continuent de progresser. Leurs progrès ont été rapides au cours des 24 à 36 derniers mois. Nous supposons qu’ils se poursuivront pendant les 24 à 48 prochains mois.

GDPval mesure ces progrès sur des tâches professionnelles réelles. Des professionnels de 44 métiers, répartis dans les neuf secteurs les plus importants de l’économie américaine, dont l’industrie manufacturière, ont fourni des tâches issues de leur propre travail. D’autres professionnels ont ensuite comparé à l’aveugle le livrable produit par une IA et celui produit par un professionnel.

Le travail du meilleur modèle a été jugé au moins aussi bon que celui du professionnel sur 12 % des tâches en mai 2024, contre 85 % en avril 2026 (GDPval, OpenAI 2025, GPT-5.2, GPT-5.5). Ces chiffres appellent à la prudence : les deux derniers sont publiés par OpenAI sur ses propres modèles et son propre benchmark. De plus, 220 tâches sont publiques, ce qui permet aux laboratoires d’entraîner leurs modèles dessus. La version indépendante d’Artificial Analysis utilise des IA comme évaluateurs plutôt que des professionnels : ses scores ne sont donc pas comparables (GDPval-AA).



GDPval : part des tâches pour lesquelles le livrable du meilleur modèle est jugé au moins aussi bon que celui d’un professionnel, selon le meilleur score publié à chaque date. Les deux premiers points proviennent de l’étude GDPval, les deux derniers des annonces d’OpenAI sur ses propres modèles.

Ces tâches fournissent tout le contexte nécessaire et sont réalisées en une seule fois. C’est l’inverse d’un ERP qui évolue pendant des années. La même étude mesure aussi le temps nécessaire à un professionnel pour vérifier le travail de l’IA et le reprendre s’il est insuffisant. En intégrant ce temps, un professionnel utilisant les modèles testés en 2025 terminait à peu près aussi vite qu’un professionnel travaillant seul : sa vitesse était multipliée par 0,87 à 1,12, contre 90 à 327 lorsque la vérification était exclue.

Notes

Le coût d'un niveau d'intelligence donné baisse rapidement. Répondre à une question scientifique de niveau doctorat aussi bien que le meilleur modèle de janvier 2025 coûtait alors 0,30 $, contre 0,0004 $ à la mi-2026 (Epoch AI). Cette mesure repose sur un benchmark public sur lequel les laboratoires peuvent entraîner leurs modèles. Une partie de la baisse peut donc venir d'une optimisation pour ces questions plutôt que d'une intelligence moins coûteuse. Le meilleur modèle disponible à chaque date ne suit pas la même tendance : le coût était d'environ 0,04 $ par question début 2024, puis de 0,07 à 0,10 $ depuis août 2025, car chaque génération raisonne plus longtemps avant de répondre (MIT FutureTech, Epoch AI). La section consacrée au coût examine les conséquences pour un ERP.



Coût d'une réponse à une question de GPQA Diamond, un ensemble de 198 questions scientifiques de niveau doctorat. Les questions sont publiques et les laboratoires peuvent entraîner leurs modèles dessus : la courbe verte peut donc baisser plus vite que le coût des capacités réelles (Epoch AI). L'échelle verticale est logarithmique : chaque graduation vaut 10 fois la précédente. En bleu : le modèle ayant le meilleur score à chaque date (données MIT FutureTech). En vert : le modèle le moins cher répondant aussi bien qu'o3 en janvier 2025 (Epoch AI).

Pour limiter notre biais, chaque affirmation ci-dessous s'appuie sur des données publiées par des tiers, avec un lien vers la source. Nous rapportons aussi les données qui contredisent notre position.

Cette démarche ne supprime pas les limites de l'analyse. La plupart des mesures proviennent de benchmarks logiciels, pas d'ERP. Une grande partie des travaux de 2026 sont des prépublications. Les enquêtes sur les intégrations concernent de grandes entreprises. Enfin, ce document est écrit par un éditeur d'ERP : le recours exclusif à des sources tierces réduit ce biais sans l'éliminer.