PEGA

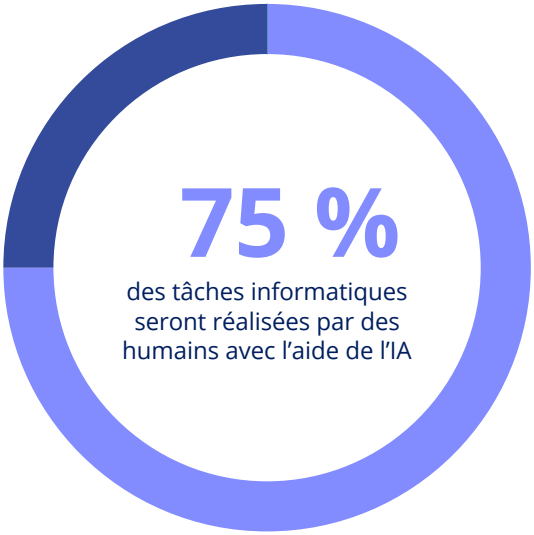
LIVRE BLANC

La taxe sur les tokens : pourquoi les agents IA autonomes menacent la rentabilité qu'ils devaient optimiser

Guide à l'intention des COO, CXO et VP Operations Leaders

L’opportunité est bien réelle

L’IA a profondément transformé le développement logiciel, qui est aujourd’hui plus rapide, plus accessible et plus performant que jamais.



Les équipes de développement peuvent concevoir et mettre en production des applications à un rythme encore impensable il y a un an. Aujourd’hui, 84 % des développeurs tirent parti de l’IA pour accélérer leur travail (Stack Overflow Developer Survey, 2025). Les équipes peuvent désormais créer en quelques jours des prototypes complets qui nécessitaient auparavant plusieurs semaines de travail. Le développement logiciel se démocratise à un rythme sans précédent. Et pour les DSI, la tendance est claire : d’ici 2030, Gartner prévoit que 75 % des tâches informatiques seront réalisées par des humains avec l’aide de l’IA, contre 25 % par l’IA seule. Autrement dit, aucune tâche informatique n’échappera à l’IA (Gartner, novembre 2025). Ce futur n’est pas si lointain : les services informatiques des grandes entreprises le préparent dès aujourd’hui.

Les investissements sont justifiés. Le cap est bon. Mais multiplier les mises en production ne permet pas encore d’obtenir les résultats métier révolutionnaires qu’attendent les conseils d’administration. Et c’est aux responsables informatiques d’expliquer pourquoi.

Le problème n’est pas technologique, il est structurel. Et il soulève trois questions essentielles auxquelles la plupart des entreprises peinent encore à répondre :

1. ***Le développement propulsé par l’IA finit-il par coûter plus cher que le recrutement d’ingénieurs supplémentaires ?***
2. ***Le code généré par l’IA est-il suffisamment fiable pour soutenir les opérations des entreprises à grande échelle ?***
3. ***Comment garder la visibilité et le contrôle sur ce que l’IA crée et déploie ?***

Coûts : investissez-vous davantage pour, au final, alourdir votre dette technique ?

La promesse du développement à l'aide de l'IA était claire : créer des logiciels plus rapidement et à moindre coût. Mais dans de nombreuses entreprises, la réalité économique s'avère bien différente de celle envisagée au départ.

L'écriture de code n'est pas la partie la plus coûteuse du développement logiciel. Elle ne représente qu'environ 10 % du coût total d'une application sur l'ensemble de son cycle de vie. Les 90 % restants couvrent la vérification du code, les tests, les correctifs de sécurité, le débogage et la refactorisation. À cela s'ajoute la maintenance nécessaire pour accompagner l'évolution des exigences métier. Les assistants de programmation propulsés par l'IA ont accéléré les 10 % consacrés à l'écriture de code. Mais les données tendent de plus en plus à montrer qu'ils ont aussi augmenté le coût des 90 % restants.

En 2025, GitClear a analysé 211 millions de lignes de code issues de dépôts de Google, Microsoft, Meta et de grandes entreprises. Le constat est frappant : dans les projets développés à l'aide de l'IA, les blocs de code dupliqués étaient huit fois plus nombreux que deux ans auparavant. Autre signal : le « code churn », c'est-à-dire les lignes de code modifiées ou supprimées dans les deux semaines suivant leur création, a doublé entre 2020 et 2024. Bill Harding, fondateur de GitClear, compare le code généré par l'IA à « une toute nouvelle carte de crédit » permettant d'accumuler de la dette technique. La dépense semble productive sur le moment, mais la facture, elle, arrive plus tard. D'après les conclusions du rapport The State of Software Delivery 2025 de Harness, la majorité des développeurs consacrent désormais plus de temps à corriger le code généré par l'IA qu'à créer de nouvelles fonctionnalités.

La taxe sur les tokens vient encore alourdir la facture. Chaque ligne de code générée par une IA consomme des tokens. C'est là toute la différence avec les tokens consommés lors d'une interaction client : la génération de code crée un actif qui continuera à mobiliser des ressources dans la durée. Si cet actif accumule de la dette technique plus vite qu'il ne crée de valeur, sa rentabilité se dégrade dès le départ, et la dette continue de s'accumuler.

Autre difficulté : le coût des outils de développement basés sur l'IA devient lui aussi moins prévisible. En juin 2026, GitHub Copilot est passé à une facturation à l'utilisation. Son modèle forfaitaire d'unités de requêtes premium a laissé place à des crédits d'IA basés sur les tokens. GitHub reconnaissait lui-même les limites de l'ancien modèle. Une question rapide dans le chat et plusieurs heures de programmation autonome pouvaient coûter la même chose à l'utilisateur. Face à la hausse des coûts d'inférence, GitHub estimait que continuer à les absorber n'était plus viable (blog GitHub, avril 2026). Pour les entreprises, la donne change. Celles qui avaient établi leurs budgets sur la base de forfaits doivent désormais composer avec des coûts variables. Elles ont aussi moins de visibilité sur les dépenses associées à chaque workflow.

La question n'est donc pas de savoir si les outils de développement basés sur l'IA sont utiles. Ils le sont. La vraie question est ailleurs : générer plus de code, plus vite grâce à l'IA réduit-il réellement le coût total du développement ? Ou cette accélération permet-elle simplement de livrer davantage aujourd'hui, tout en créant une charge de maintenance qui pèsera sur les budgets de demain ?

Performance et sécurité : pouvez-vous vous fier à ce que crée l'IA ?

Accélérer les mises en production n'a de valeur que si les solutions déployées sont suffisamment fiables pour fonctionner à l'échelle de l'entreprise. Or, les données invitent les responsables informatiques à la prudence.

CodeRabbit a analysé 470 pull requests issues de projets réels. Le code généré par l'IA présentait 1,7 fois plus de problèmes que celui écrit par des humains. Et surtout, 2,74 fois plus de vulnérabilités de sécurité (CodeRabbit State of AI vs. Human Code Generation, décembre 2025). Selon des travaux de l'IEEE présentés à l'ISSRE 2025, le code généré par l'IA présente davantage de vulnérabilités de sécurité à haut risque, alors même que sa structure est plus simple. Le problème ne tient donc pas uniquement à la complexité du code, il tient aussi à la façon dont sa logique est construite.

Les outils d'IA répondent à des problèmes précis et immédiats. Mais ils n'ont pas nécessairement une vision d'ensemble du système dans lequel leur code doit s'intégrer : architecture, abstractions internes, utilitaires partagés ou règles de sécurité régissant les flux de données. Le code peut alors réussir les tests unitaires et franchir les premières étapes de validation. Ses failles, elles, ne se révèlent parfois que dans des scénarios atypiques ou lorsque le système est soumis à une forte charge en production.

76 % des développeurs utilisant des outils de programmation basés sur l'IA déclarent avoir déjà généré du code qu'ils ne comprenaient pas entièrement (Stack Overflow, 2026). Ce chiffre a des conséquences opérationnelles très concrètes. Lorsqu'une autorité de contrôle, une équipe juridique ou un conseil d'administration demande à un DSI pourquoi un système s'est comporté d'une certaine manière, « l'IA a généré le code et le développeur n'en comprenait pas précisément le fonctionnement » n'est pas une réponse acceptable. Pourtant, c'est précisément le risque que prennent les entreprises lorsqu'elles déploient du code généré par l'IA sans cadre de gouvernance adapté.

Gartner va plus loin. Parmi ses principales prévisions pour 2026, le cabinet estime que plus de 2 000 actions en justice liées à des décès imputables à l'IA seront engagées dans le monde. En cause : une gouvernance et des contrôles des risques insuffisants sur les résultats générés par l'IA (Gartner, octobre 2025). Et ce risque n'appartient pas à un futur hypothétique. Les entreprises qui confient aujourd'hui au code généré par l'IA des décisions importantes dans des workflows réglementés ou en contact avec les clients y sont d'ores et déjà exposées.

Fiabilité : savez-vous vraiment ce que crée l'IA ?

Si la rapidité et la sécurité sont des enjeux opérationnels, l'auditabilité est un enjeu structurel. Et pour les DSI et les architectes d'entreprise, c'est peut-être aussi le plus difficile à intégrer après coup.

Lorsqu'une application prend une décision, qu'il s'agisse d'acheminer un dossier, de déterminer une éligibilité, d'accorder un accès ou de générer du contenu pour les clients, c'est l'équipe informatique qui en assume la responsabilité, pas l'IA qui a écrit le code. L'entreprise a déployé l'application, les responsables l'ont validée : à eux de pouvoir en justifier les décisions.

Dans un environnement de développement traditionnel, cette responsabilité reste maîtrisable, car la logique peut être examinée. Si un auditeur demande pourquoi telle décision a été prise, le développeur peut suivre le chemin d'exécution du code et identifier ce qui l'a déclenchée. Mais avec du code généré par l'IA, cette traçabilité n'est plus garantie. Le code peut produire le résultat attendu sans que les choix qui ont guidé sa conception soient documentés comme ils le seraient dans le cadre d'une conception humaine délibérée. Et si un auditeur demande pourquoi le système a pris telle décision dans une situation particulière, « parce que l'IA l'a écrit ainsi » n'est pas une réponse défendable.

C'est là que se joue véritablement la fiabilité à l'échelle de l'entreprise. Il ne suffit pas que le code fonctionne correctement dans la majorité des cas. Chaque résultat doit pouvoir être expliqué, à tout moment, à toute personne qui en fait la demande. Et cette explication doit être suffisamment claire pour permettre d'agir.

Une logique applicative visuelle et déclarative permet de résoudre le problème à la source. Les règles métier, les tables de décision et le routage des workflows ne sont plus enfouis dans du code généré. Ils prennent la forme de configurations qui peuvent être consultées et vérifiées. Chaque décision devient ainsi traçable sans devoir décortiquer le code pour en reconstituer le cheminement. La logique devient accessible aux équipes de conformité et juridiques, ainsi qu'aux équipes métier, plus seulement aux ingénieurs. L'audit change alors de nature : il ne s'agit plus d'examiner le code pour comprendre ce qui s'est passé, mais simplement d'analyser la configuration.

L’architecture qui répond aux trois enjeux

Pour maîtriser à la fois les coûts, la sécurité et la fiabilité, il ne s’agit pas de limiter l’usage de l’IA. Il faut repenser son rôle dans la pile de développement.

C’est dans la conception des structures applicatives de haut niveau que l’IA apporte le plus de valeur. Modèles de données, définitions de workflows, tables de décision, logique de routage sous forme déclarative : ces artefacts décrivent ce que l’application doit faire, sans traduire directement son fonctionnement en code impératif. Contrairement au code généré, ils sont lisibles, auditables et faciles à maintenir. Lorsque les exigences métier évoluent, le modèle s’adapte et le comportement de l’application avec lui. Pas de modifications en cascade dans le code ni les risques de régression qui les accompagnent.

C’est tout le principe d’une approche « model-first ». L’IA intervient là où elle excelle : traduire une intention en structure, transformer le langage naturel en configuration et accélérer la définition de l’architecture. Il en résulte un modèle, et non du code. Et lorsqu’il faut générer du code, cette étape reste en arrière-plan. Elle relève de l’implémentation et ne constitue plus le principal livrable.

Les bénéfices opérationnels sont immédiats. Les équipes de développement passent moins de temps à assurer la maintenance, puisqu’elles font évoluer le modèle plutôt que la base de code. Les contrôles de sécurité sont simplifiés grâce à une logique métier exprimée de manière déclarative. Les audits de conformité sont plus rapides, car les décisions peuvent être examinées sans faire systématiquement appel aux équipes d’ingénierie. Même les dépenses en tokens sont mieux ciblées. Elles se concentrent sur ce qui crée le plus de valeur : définir ce que l’application doit faire, plutôt que générer tout le code nécessaire à son fonctionnement.

Selon l’enquête I&O menée par Gartner fin 2025, seuls 28 % des cas d’usage de l’IA dans les infrastructures répondent pleinement aux attentes en matière de ROI. Ce décalage entre promesse et réalité ne vient pas d’un manque de capacités de l’IA. Il tient à l’architecture. Les entreprises qui parviennent à rentabiliser l’IA dans le développement n’utilisent pas nécessairement des modèles plus avancés. Elles l’utilisent pour créer des artefacts qui conservent leur valeur plus longtemps que le code généré.

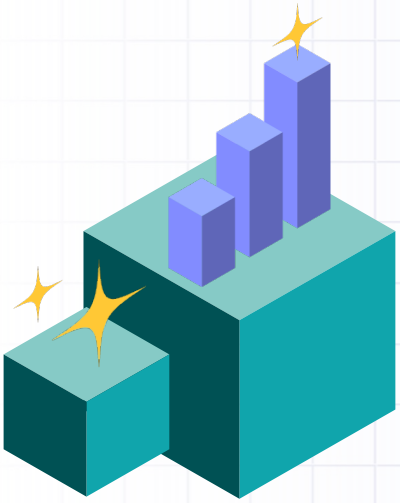
Le choix d’architecture

Les responsables informatiques n’ont pas à choisir entre rapidité et sécurité. Poser le problème en ces termes revient à accepter un compromis qui n’a pas lieu d’être. Le véritable choix porte sur l’architecture, avec des conséquences très différentes sur les coûts et les risques à long terme.

Une architecture centrée sur le code généré par l’IA permet d’obtenir rapidement de premiers résultats. Mais cette rapidité a des conséquences dans la durée : la maintenance s’alourdit, l’auditabilité diminue, et les risques de sécurité augmentent à mesure que la base de code s’étend. L’approche « model-first » change la donne . En intégrant l’IA dès la conception, elle permet de créer des applications auditables et faciles à maintenir au niveau du modèle. Les dépenses en tokens se concentrent alors sur les étapes du développement qui créent le plus de valeur dans la durée.

Les responsables informatiques n’ont pas à renoncer aux gains de rapidité que l’entreprise attend de l’IA. L’idée n’est pas de ralentir, mais d’adopter une approche de développement viable à long terme, tout en donnant aux équipes informatiques les garanties dont elles ont besoin.

Investir judicieusement dans l’IA, ce n’est pas chercher à écrire du code plus vite, mais plutôt concevoir des modèles applicatifs qui n’auront pas besoin d’être réécrits.



Merci

À propos de Pega

Pega offre la plateforme basée sur l'IA leader pour accompagner la transformation des entreprises. Les organisations les plus influentes au monde font confiance à notre technologie pour repenser leur façon de travailler, avec l'automatisation des workflows, la personnalisation de l'expérience client et la modernisation des systèmes legacy. Depuis 1983, notre architecture évolutive et flexible favorise l'innovation continue et aide nos clients dans leur parcours vers l'entreprise autonome.
PEGA.COM/FR