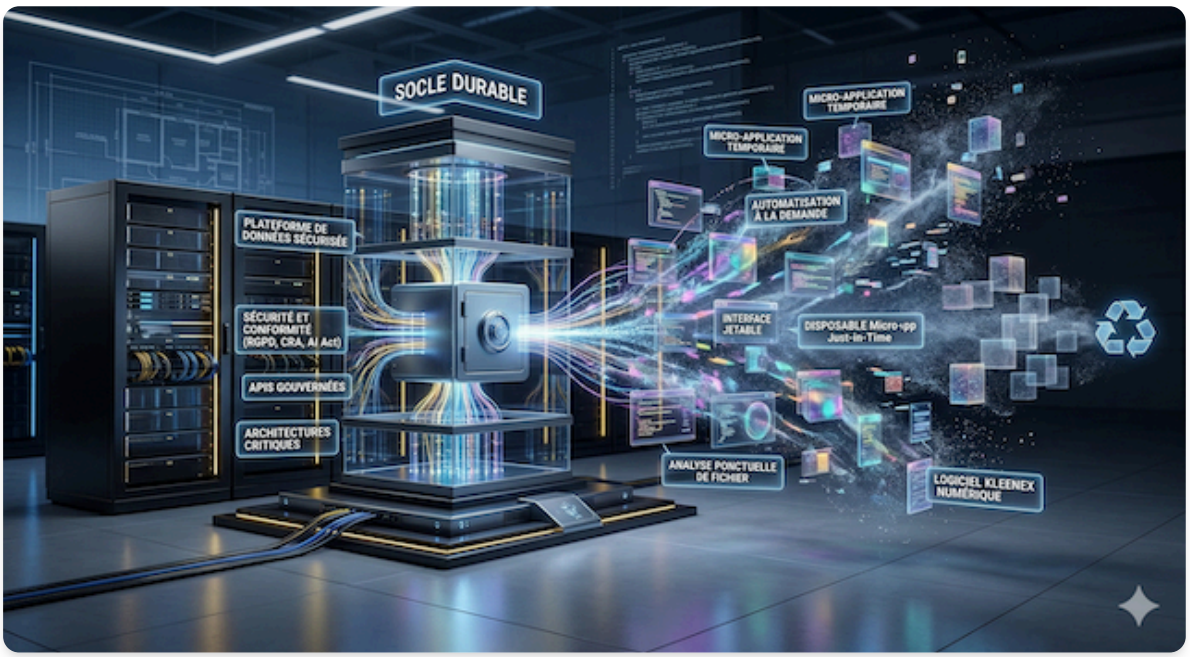
Logiciel à usage unique

8 March 2026 • 4 min de lecture

Code Generation



> *Plus besoin d'être un "pisseur de code" pour bricoler un outil. Aujourd'hui, même papi peut commander à la machine : il ajuste son dentier, articule un prompt à peu près compréhensible ... et l'IA lui génère une application. La vitesse de production dépasse l'entendement humain : quelques phrases suffisent pour matérialiser une micro-application qui automatise une tâche, analyse un fichier ou connecte deux systèmes.*

insight_epistemic

Une partie de l'ingénierie logicielle "à la papa" est clairement en train de vaciller.

Les cycles en V de 18 mois pour produire un formulaire CRUD ? Les backlog Jira qui ressemblent à la file d'attente d'un parc d'attraction ?

Tout cela commence à sentir la naphtaline. Nous entrons probablement dans l'ère des applications "Just-in-Time".

Un utilisateur a un problème, il décrit la tâche et l'IA génère une micro-app pour l'exécuter. Une fois le travail terminé, on sort la poubelle; le logiciel devient parfois un consommable : un Kleenex numérique utilisé pour résoudre un problème ponctuel.

Et dans beaucoup de cas ces scripts, automatisations, petits outils internes sont une révolution de productivité.

Mais comme souvent avec les révolutions technologiques, la réalité est un peu moins simple que le slogan. Non, nous n'allons pas vivre dans un monde où tout le code est jetable : une entreprise ne fonctionne pas uniquement avec des petits scripts improvisés entre deux cafés.

Elle repose sur des fondations solides composées de données, identités, règles métier, sécurité et de de plus en plus de réglementation complexes induisant des responsabilités (AI Act, RGPD, CRA, PLD, NIS2, DORA, ...)

Ce socle central ne peut pas être régénéré à chaque prompt comme un PowerPoint un peu moche, il doit être stable, auditable et gouverné.

Le modèle qui se dessine ressemble plutôt à une architecture à deux vitesses.

- un socle durable : plateformes de données, APIs, sécurité, systèmes critiques où l'ingénierie reste un discipline sérieuse avec peu de magie, beaucoup de responsabilité.
- une explosion d'outils générés pour de l'automatisation à façon, des interfaces temporaires, des analyses ponctuelles, des micro-apps produites à la demande. Ce sont des artefacts numériques que l'on crée pour résoudre un problème puis que l'on jette.

Cela ne supprime pas les problèmes mais en crée même de nouveaux car si tout le monde génère des apps qui contrôlent les accès aux données ? Qui

audite ce que fait réellement l'outil ? Qui garantit que l'IA n'a pas improvisé une règle métier douteuse ?

La dette technique ne disparaît pas non plus mais change simplement de forme.

Une chose est néanmoins assez claire, dans ce nouveau monde, la rareté ne sera probablement plus le développeur qui sait coder un formulaire. La rareté sera l'architecte capable de concevoir le socle sur lequel toutes ces apps jetables vont tourner sans faire exploser l'entreprise.

En résumé, le logiciel critique restera un actif durable et une grande partie des outils deviendra éphémère. Entre les deux, des milliers de petites applications générées à la volée par des humains et des agents.

Le futur du logiciel ressemble peut-être moins à une cathédrale de code maintenue pendant vingt ans et un peu plus à un immense distributeur de Kleenex numériques; même papi pourra s'en servir à condition que quelqu'un, quelque part, ait construit correctement la plomberie derrière le mur.

Les opinions exprimées dans cet article sont strictement personnelles et ne reflètent pas nécessairement celles de mon employeur. Les contenus sont fournis à titre informatif et ne constituent pas un conseil juridique. Cet article explore des concepts architecturaux émergents et analyse des tendances de marché (Gartner, Forrester). Les solutions technologiques citées le sont à titre d'exemple et ne préjugent pas des choix technologiques ou des partenariats de mon employeur

Eric Blaudez - AI Architect | Responsible AI · AI Act · AI Governance | Bringing R&D AI to Production (TRL 3→6) | EU & Sovereign Programs

Logiciel à usage unique

© 2026 Eric Blaudez. All rights reserved.

Les opinions exprimées sur ce site sont strictement personnelles et ne reflètent pas nécessairement celles de mon employeur. Les contenus sont fournis à titre informatif et ne constituent pas un conseil juridique.