REDSEN

Dossier de référence

Production de logiciel assistée par l'IA

« La productivité s'achète au token, la responsabilité ne se délègue pas »

Édition 2026 · version 1.1 · données arrêtées au 17 août 2026
Dossier complet : chapitres, annexes et bibliographie.

Production de logiciel assistée par l'IA
Dossier de référence.

Édition 2026 · version 1.1 · données arrêtées au 17 août 2026

Document de position et de cadrage publié par REDSEN. Il n'est ni un conseil juridique, ni un audit, ni une recommandation d'investissement. Les analyses réglementaires décrivent un état du droit à la date d'arrêté et ne dispensent pas de l'avis d'un conseil.

Rédaction : Emmanuel Bouchet, associé gérant, et Xavier Perrin, consultant senior. Avant-propos par Alain Lefebvre, qui n'a pas participé à la rédaction du corps du document.

Note sur l'usage de l'intelligence artificielle. Cet ouvrage a été conçu et rédigé sous la responsabilité de ses auteurs. Des outils d'intelligence artificielle générative ont été mobilisés à certaines étapes du travail : Perplexity pour l'exploration documentaire et l'identification de sources ; Claude pour l'assistance à la structuration, à la reformulation et à la relecture de certains passages. Les choix éditoriaux, l'analyse, la sélection des sources, la vérification des informations et la version finale relèvent de la responsabilité exclusive des contributeurs.

Toute correction, contestation ou remarque est bienvenue à l'adresse éditoriale REDSEN.

Dossier de référence REDSEN – Production de logiciel assistée par l'IA

La productivité s'achète au token, la responsabilité ne se délègue pas

Cadre de lecture et référentiel de maturité

Édition 2026 — version 1.1

. . .

À propos de ce document

Statut. Dossier de référence REDSEN, document de position et de cadrage destiné à être diffusé. Version 1.1, données arrêtées au 17 août 2026.

Destinataires. Directions des systèmes d'information et leurs équipes d'ingénierie logicielle : responsables d'études, architectes, responsables qualité et achats IT. Le document se lit sans prérequis technique.

Portée. Ce document n'est ni un conseil juridique, ni un audit, ni une recommandation d'investissement. Les analyses réglementaires décrivent un état du droit à la date d'arrêté et ne dispensent pas de l'avis d'un conseil ; les seuils, paliers et dispositifs proposés sont des repères à adapter. Les données chiffrées vieillissant vite et plusieurs des sources mobilisées étant révisées annuellement, ce document a vocation à l'être aussi.

Rédaction. Emmanuel Bouchet, associé gérant, et Xavier Perrin, consultant senior. L'avant-propos est signé par Alain Lefebvre, qui n'a pas participé à la rédaction du corps du document.

Chiffres et niveaux de preuve. Chaque affirmation chiffrée porte un niveau (**A**, **B** ou **C**) et renvoie à une entrée numérotée de l'annexe B. La grille, le traitement des sources d'origine fournisseur et la déclaration d'intérêts sont en annexe D ; les termes propres à ce dossier, en annexe E.

Toute correction, contestation ou remarque est bienvenue à l'adresse éditoriale REDSEN.

. . .

Avant-propos

Par Alain LEFEBVRE

Depuis l'apparition de ChatGPT (OpenAI), on est continuellement sous un déluge de démos et de promesses toutes plus impressionnantes que les autres. Il en résulte un effet de sidération qui perturbe la capacité d'analyse du grand public (ce qui n'est guère surprenant), mais aussi du milieu professionnel (ce qui est tout de même plus embêtant !).

Pendant toutes ces années, nous avons été abreuvés de soi-disant études qui attestaient toutes des progrès formidables apportés par l'IA générative dans tous les domaines et en particulier dans le développement d'applications informatiques. Problème : ces études provenaient toutes des fournisseurs de LLM comme OpenAI ou Anthropic !

Accorder crédit à ce type d'étude, c'est avaler tout cru les promesses marketing les plus extravagantes… C'est un choix, mais il ne faut pas se plaindre après coup d'avoir été trompé !

Et c'est pour cela que ce dossier réalisé par REDSEN est si précieux : il vous propose une plongée dans les *faits mesurables* en laissant de côté les promesses éthérées, ça fait une sacrée différence…

ILS TRAVESTISSENT LA RÉALITÉ, TOUS !

Pour illustrer mon propos, un exemple concret. À la conférence GTC de mars 2024, NVIDIA annonce l'architecture Blackwell — le superchip GB200/NVL72 — avec un argument spectaculaire : coût et consommation d'énergie divisés par 25 par rapport à la génération Hopper, inférence jusqu'à 30 fois plus rapide [1]. Aucun de ces deux chiffres n'est faux, et aucun ne compare ce qu'il prétend comparer : la démonstration oppose une baie GB200 en FP4 à un nombre équivalent de H100 en FP8, soit deux précisions de calcul différentes [2]. À isoprécision, les bancs indépendants ramènent le gain entre un facteur 1,6 et 4,3 par GPU [3] [4]. Sur l'énergie par jeton, la mesure ouverte donne un facteur voisin de 3 en interactivité courante et jusqu'à 7 à forte interactivité, sur des serveurs à huit cartes et non sur la baie NVL72 que revendique l'annonce [4] [5]. Nous retenons délibérément le bas de cette fourchette. Un facteur 3, c'est déjà remarquable.

Mais pour fixer les imaginations et provoquer la sidération de « l'effet démo », 25 était bien mieux que 3…

NVIDIA n'est pas un cas isolé. La bibliographie de ce dossier recense plusieurs autres exemples documentés : le classement LMArena de Meta, dont un dirigeant de l'entreprise a fini par reconnaître qu'il avait été arrangé [6] ; le graphique de xAI sur Grok 3, dont un mode de calcul avait simplement été masqué [7] ; le benchmark FrontierMath, financé en secret par OpenAI qui allait ensuite y annoncer un score record [8]. Une revue systématique de 445 benchmarks conclut d'ailleurs que 16 % seulement emploient des tests statistiques rigoureux [9].

NVIDIA n'est donc pas la seule à jouer avec ses propres chiffres. Il suffit de suivre les déclarations de Sam Altman (OpenAI) ou de Dario Amodei (Anthropic) pour se rendre compte que ces dirigeants ont un sens de l'exagération particulièrement développé. Là encore, un bon exemple est très parlant : Dario Amodei déclare que Claude (son modèle vedette) va remplacer tous les programmeurs de la planète à brève échéance (et ça fait deux ans qu'il répète cette « prédiction »). Mais vous savez bien, tout comme avec les politiques, qu'il vaut mieux « regarder ce qu'ils font plutôt qu'écouter ce qu'ils disent »…

Depuis la prédiction fracassante de son CEO en mars 2025, Anthropic n'a pas ralenti l'embauche de développeurs — au contraire, l'effectif de l'entreprise est passé d'environ 1 300 à environ 3 400 personnes en 9 mois [10], et 64 postes de *software engineers* sont à pourvoir au moment où j'écris ces lignes [11].

LA VALEUR DU DOSSIER REDSEN

Il est donc important de faire la part des choses en la matière et c'est justement ce grand vent d'air frais qu'apporte le dossier de REDSEN sur ce sujet brûlant. Mais ce n'est pas tout !

REDSEN ne se contente pas de revisiter les faits en s'appuyant sur les rares travaux dont le protocole est publié, les limites documentées et les conclusions défavorables à qui les finance — Cui et al., METR, GitClear : ce dossier remet en exergue un point important trop souvent occulté — la production logicielle ne se résume pas au *coding*.

En effet, il y a des phases en amont et en aval du *coding* qui sont tout aussi cruciales que l'écriture du code, qui n'est qu'un moyen, pas une fin.

Avec ce dossier, c'est la fin de « l'euphorie sans données ». Ce document rejette tout autant l'enthousiasme démesuré que le rejet dogmatique de l'IA générative appliquée au développement informatique, en s'appuyant sur des études indépendantes (Cui et al., METR, GitClear). Il en ressort deux constats clés : tout d'abord, l'IA générative accélère la production des tâches standardisées du développement, mais elle n'est pas « une machine de responsabilité », comprendre qu'on ne peut pas lui déléguer les décisions de projet.

Ensuite, on constate une certaine dégradation de la qualité du code lui-même. Les données réelles montrent une érosion de la lisibilité et de la maintenabilité (hausse du code copié-collé, baisse drastique du refactoring) si le code n'est pas strictement contrôlé, après coup, par un humain — expérimenté de préférence…

En résumé : l'IA générative libère du geste technique (au moins en partie), mais exige d'augmenter la rigueur du jugement humain. L'enjeu pour les DSI n'est plus d'adopter l'IA à tout prix, mais de piloter sa maturité pour maîtriser les risques de dette et de responsabilité.

Présenté ainsi, ça fait un peu « sec ». Mais c'est justement tout le mérite de ce document, qui privilégie une approche dépassionnée changeant agréablement de l'ambiance globale qui prévaut autour des questions liées à l'IA. Je voudrais finir par une affirmation personnelle : ce dossier est le meilleur texte que j'ai pu lire depuis trois ans au sujet de l'impact de l'IA sur le développement des applications informatiques, point.

Alain LEFEBVRE *Août 2026*

Alain Lefebvre a cofondé SQLI en 1990, quand le secteur des services numériques se constituait en France. Il est l'auteur de plusieurs ouvrages sur les grandes ruptures de l'informatique, dont *L'Architecture client-serveur*, prix AFISI du meilleur livre informatique 1994. Il a vu passer assez de ruptures annoncées pour distinguer celles qui tiennent de celles qui se racontent.

. . .

Préface – Lettre de conviction

Ce dossier n'est pas une prise de position optimiste sur l'IA générative, et ce n'est pas non plus un manifeste de prudence. Il défend un usage sous condition, et il vaut mieux le dire d'emblée que le laisser apparaître au troisième chapitre. **Nous soutenons que l'assistance générative est un levier réel, et qu'elle ne produit de valeur que dans un contexte où organisation, compétences, outillage et gouvernance permettent d'absorber ce qu'elle produit.** Ce qui suit est un ensemble de convictions écrites depuis la place que nous occupons : une entreprise de services numériques qui livre du code à des clients, sous contrat, avec des engagements de qualité et de délais.

Nous écrivons parce que la conversation publique sur l'IA dans le développement logiciel est aujourd'hui dominée par deux voix également indéfendables. La première célèbre des gains de productivité qui ne survivent pas à un examen sérieux. La seconde annonce la fin du métier d'ingénieur, en confondant régulièrement l'outil, l'agent et l'organisation qui les emploie.

La phrase portée en couverture n'est pas un slogan, et nous la tenons pour la plus importante de ce dossier. **La productivité s'achète au token, la responsabilité ne se délègue pas**. Elle produit vite ce que nous savons déjà produire. Elle nous décharge de certaines activités, elle ne nous libère pas de la validation. Elle rend l'ingénierie plus rapide, elle ne la rend pas moins exigeante.

Ce dossier s'adresse à l'ensemble des acteurs impliqués dans la construction de logiciels. Il propose une grammaire de lecture (CMMI), une analyse des zones de risque adossée à six fiches de détection, un référentiel de maturité opposable, des actions concrètes pour démarrer.

Nous ne cherchons pas à convaincre. Nous cherchons à donner à chacun les moyens de décider en connaissance de cause, et d'assumer sa décision devant un client, un régulateur ou un auditeur.

Emmanuel BOUCHET, associé gérant, et Xavier PERRIN, consultant senior *REDSEN — Août 2026*

Sommaire

Dossier de référence REDSEN — Production de logiciel assistée par l'IA *Édition 2026 · version 1.1 · données arrêtées au 17 août 2026*

. . .

. . .

Annexes

Annexe A — Matrice des décisions Trame à remplir : qui signe, qui arbitre, qui propose, quelle preuve. Vingt-huit décisions réparties sur les cinq dimensions du référentiel.

Annexe B — Bibliographie Trente-six sources numérotées par ordre de première apparition, avec URL et niveau de preuve. Comprend la liste des chiffres écartés faute de sources et les actions restant à conduire avant publication.

Annexe C — Fiches complètes des six mécanismes Le référentiel de détection : l'ensemble des signaux et de leurs seuils, les sept tests provoqués, les contre-mesures détaillées et leurs effets pervers.

Annexe D — Méthode, sources et intérêts Règle de sobriété évidentielle, niveaux de preuve, traitement des sources d'origine fournisseur, périmètre et limites, déclaration d'intérêts.

Annexe E — Glossaire Les termes propres à ce dossier, et ceux dont l'usage courant prête à confusion.

. . .

Parcours de lecture

Un dirigeant d'une direction des systèmes d'information lit l'ensemble, et consulte l'annexe C au moment d'installer les dispositifs.

Une direction de projet trouve l'essentiel de ce qui la concerne au chapitre 3, au §5.1 et à l'annexe C.

Un lecteur qui veut d'abord vérifier commence par l'annexe D, qui dit selon quelle règle les chiffres ont été retenus et à qui ce document profite, puis par l'annexe B.

Chapitre 1 — Sortir de la sidération

Ce que l'IA générative fait et ne fait pas dans le développement logiciel

1.1 Deux ans de hype, un an de reflux

Nous entrons dans une nouvelle phase. Les premières années de l'IA générative dans le développement ont été dominées par des annonces spectaculaires : gains de productivité, projections d'un métier d'ingénieur transformé en 18 mois, promesses d'agents autonomes remplaçant des équipes entières. Ces annonces étaient portées par les acteurs qui commercialisent les outils, relayées sans filtre par la presse technologique, et rarement confrontées à des mesures indépendantes.

L'année 2025 marque un point de bascule. Les premières études empiriques rigoureuses arrivent, et elles ne confirment pas les chiffres publicitaires. Certaines de ces études sont elles-mêmes révisées en 2026 à mesure que les auteurs constatent les limites de leurs propres protocoles [12]. La conversation publique se polarise entre deux discours : celui qui célèbre la révolution productiviste sans données solides, et celui qui annonce la fin du métier sans lire les études qui contredisent ses prédictions. Notre position est de refuser les deux, en revenant aux mesures indépendantes disponibles, en les datant, en les attribuant, et en les contextualisant précisément.

Sortir de la sidération n'est pas se ranger dans le camp des sceptiques. C'est cesser de raisonner par slogans, et de refuser autant l'euphorie sans données que la défiance sans lectures. C'est aussi admettre que l'IA générative change certaines choses, mais n'en change pas d'autres. En tant que spécialistes des services, notre tâche consiste à déterminer lesquels.

1.2 Les chiffres attaquables, datés et attribués

Quatre familles de chiffres circulent aujourd'hui et méritent d'être manipulées avec précaution.

« 55 % DE TEMPS EN MOINS POUR ÉCRIRE DU CODE »

Le chiffre provient d'une étude GitHub 2022 sur 95 développeurs recrutés pour l'expérience, à qui on demande d'implémenter un serveur HTTP en JavaScript avec et sans Copilot [13]. Il ne mesure pas la productivité en contexte de production. Il mesure la vitesse d'exécution d'une tâche synthétique, sur des développeurs qui n'ont pas d'historique dans la base de code, sans coût de revue aval, sans mesure de la qualité. Le chiffre est vrai dans son contexte, il est faux hors de son contexte. Il ne peut pas servir à budgéter une équipe pluriannuelle.

Le glissement typique consiste à extraire ce chiffre de son étude et à le présenter comme une propriété générale de l'IA. C'est un raisonnement fallacieux : il transforme une mesure de laboratoire en promesse commerciale. Un dirigeant qui fonde son plan d'affaires sur ce chiffre prend le risque de constater, à la fin de l'exercice, que la productivité réelle mesurée n'a rien à voir avec la projection initiale.

« 26 % DE TÂCHES COMPLÉTÉES EN PLUS » OU « 19 % PLUS LENTS »

L'étude Cui, Demirer, Jaffe, Musolff, Peng et Salz (chercheurs Microsoft, MIT, Princeton et Wharton), publiée en 2024, révisée en 2025 et parue dans *Management Science* en 2026, agrège trois essais contrôlés randomisés menés en conditions réelles chez Microsoft, Accenture et une entreprise anonyme du Fortune 100, sur **4 867 développeurs** [14]. Elle mesure une **augmentation de 26,08 % du nombre de tâches complétées** (mesuré sur les pull requests) chez les développeurs équipés d'un assistant IA, avec une erreur type de 10,3 %, les gains les plus élevés concernant les développeurs les moins expérimentés.

L'étude METR de juillet 2025 mesure sur 16 développeurs open source expérimentés un **ralentissement de 19 %** avec Cursor Pro sur leurs propres bases de code, en tâches réelles, en essai contrôlé randomisé [15]. Ce résultat a fait débat, et **METR a publié en février 2026 une note d'actualisation** [12] dont l'enseignement est méthodologique avant d'être quantitatif. Le réexamen n'est pas parvenu à confirmer ni à infirmer les −19 % avec une confiance suffisante : l'estimation ponctuelle sur les développeurs revenus dans l'étude reste un ralentissement, de 18 %, mais avec un intervalle de confiance si large qu'il englobe une accélération. METR qualifie lui-même ces données de signal peu fiable. La raison est liée au terrain plutôt qu'au protocole : **maintenant, de 30 à 50 % des développeurs refusent de travailler sans assistance**, ce qui entraîne un biais de sélection et rend la reproduction difficile.

Cette révision est structurante, et pas dans le sens qu'on attendrait. Elle ne corrige pas un chiffre à la hausse ou à la baisse : elle montre que **la mesure de la productivité IA se dérobe à mesure que l'IA se normalise**. Mettre en place un protocole qui compare le travail avec et sans assistance devient impossible lorsque la moitié des sujets refusent la condition témoin. Cela constitue une raison de plus de piloter sur des mesures internes datées.

Ces deux chiffres coexistent parce qu'ils ne mesurent pas la même chose. L'étude Cui et al. mesure des tâches sans distinction fine de complexité, sur une population large et diverse, avec les développeurs les moins expérimentés qui gagnent le plus. L'étude METR mesure des développeurs experts, sur des bases de code qu'ils maîtrisent depuis des années. Elle montre que le gain IA disparaît, voire s'inverse, lorsque le développeur maîtrise déjà mieux le contexte que l'IA.

L'IA générative amplifie les compétences quand elle apporte un contexte que le développeur n'a pas, et elle le freine quand elle propose un contexte moins pertinent que celui déjà intégré par l'ingénieur.

« L'IA AMÉLIORE LA QUALITÉ DU CODE »

Aucune étude indépendante de grande ampleur ne confirme cette affirmation à ce jour. Le rapport GitClear 2026, actualisation de l'analyse 2025, porte sur **623 millions de changements de code analysés entre 2023 et le premier semestre 2026** [16] et documente une dégradation mesurable. Les quatre séries sont données ici avec leurs bases : le reste du dossier y renvoie sans les répéter.

Indicateur	Référence	Mesure 2026	Variation
Code copié-collé	9,4 % (2022)	15,7 % (S1 2026)	+67 % en 4 ans
Refactoring (déplacement de lignes)	21 % (2022)	3,8 % (2026)	−70 % selon le chiffre publié par GitClear ; **l'arithmétique directe sur ces deux points donne −82 %**, et nous signalons l'écart plutôt que de trancher à la place de l'éditeur
Duplication de blocs	40,3 par million de lignes modifiées (2023)	73,0 (2026)	+81 % en 3 ans
Maintenance du code hérité	1,7 % (2022)	0,46 %	−72,9 %

Les indicateurs mesurés sur des commits réels montrent que l'utilisation de l'IA entraîne une dégradation de la lisibilité et de la maintenabilité du code.

Un mécanisme cognitif accompagne cette érosion : la sous-estimation du temps de contrôle, fragmenté et peu visible, et un biais d'optimisme sur la qualité de ce qui a été généré. Ce mécanisme est indépendant du type de tâche et se retrouve dans nos observations sur nos projets chez REDSEN comme dans l'étude METR de juillet 2025 [15].

Les enquêtes internes sur l'utilisation de l'IA produisent des chiffres flatteurs qui ne reflètent pas la réalité. Les développeurs perçoivent une accélération, mais celle-ci est due au temps de génération plus rapide, tandis que le temps de contrôle est invisible, car fragmenté. Le pilotage doit se baser sur des mesures objectives (temps de cycle, qualité livrée, dette accumulée).

« 90 % DES DÉVELOPPEURS UTILISENT L'IA ET PLUS DE 80 % SE DISENT PLUS PRODUCTIFS »

Ces deux chiffres proviennent de l'enquête DORA 2025 [17], menée auprès de près de 5 000 professionnels. Ce sont des chiffres d'adoption et de **perception**, pas de mesure : une adoption plus forte est associée à une hausse du débit de livraison et de l'instabilité de livraison.

Deux précautions accompagnent cette source, et il est important de les souligner. Le programme est **opéré par Google Cloud**, acteur qui vend de l'IA et des plateformes ; plusieurs de ses conclusions sont défavorables à un déploiement naïf, ce qui satisfait le critère de recevabilité posé en annexe D. Les données sont **déclaratives et corrélationnelles** : l'association entre adoption et instabilité n'établit pas de causalité.

L'écart entre ces 80 % de perception et la mesure METR [15] est le même écart que celui documenté par Microsoft [18] sur des tâches d'écriture. Il traverse ce dossier de bout en bout : **la croyance en un gain est quasi universelle, la mesure du gain est contextuelle.**

L'ÉCONOMIE DU TOKEN – UN COÛT STRUCTURELLEMENT NON PRÉDICTIBLE

Le token est l'unité de facturation de l'IA générative : chaque appel est facturé au volume de tokens consommés en entrée (le contexte transmis) et produits en sortie (la réponse générée). Cette unité a une propriété qu'aucune autre unité d'achat informatique ne partage : le volume facturé n'est pas spécifié par l'acheteur. La longueur de la réponse est décidée par le modèle, appel par appel ; les modes de raisonnement avancés consomment des tokens intermédiaires invisibles dans la réponse finale ; et les tarifs au million de tokens sont révisables par le fournisseur sans engagement de durée. Un DSI qui achète des tokens achète une unité dont ni la quantité par tâche, ni le prix dans le temps ne sont contractuellement fixés.

La non-prédictibilité opère donc à plusieurs niveaux. Par appel : deux appels de spécification identique produisent des sorties de longueur et de coût différents, faute de reproductibilité garantie par les fournisseurs (cf. §1.3). Par exercice : la politique tarifaire du fournisseur peut changer en cours d'année, sans que le client dispose d'un levier contractuel comparable à ceux des licences logicielles classiques.

La conséquence est directe : le coût token ne se prédit pas, il se maîtrise. La budgétisation par enveloppe (par équipe, par mois), les seuils d'alerte et le suivi de la variance mensuelle, développés au §3.4 (mécanisme 6, fiche complète en annexe C.6) et au §5.1, sont les moyens de contrôler une unité d'achat qui échappe à la prévision.

1.3 Trois natures d'IA qu'il faut cesser de confondre

Le débat public souffre d'une confusion permanente entre trois objets différents qui n'appellent ni les mêmes usages, ni les mêmes contrôles, ni les mêmes contrats.

L'IA prédictive appartient à la famille des modèles de classification, de détection d'anomalies et de scoring. Elle s'est développée, s'est montrée mesurable, s'est vu encadrer par des méthodologies éprouvées depuis quinze ans. Elle produit des résultats reproductibles à contexte constant. Elle relève d'un contrat de type performance mesurable : accuracy, précision, rappel, latence.

L'IA générative produit du texte, du code, des images, du son. Elle est probabiliste par nature. Deux appels identiques peuvent produire deux résultats différents. Elle ne possède pas de mémoire propre entre appels, elle hallucine, elle ne connaît pas la vérité, seulement la plausibilité statistique de séquences de tokens. Elle relève d'un contrat de type prestation avec obligation de moyens et de traçabilité, pas de résultat reproductible.

L'IA agentique compose des chaînes d'appels IA générative avec des outils (accès web, exécution de code, appels d'API) pour accomplir des tâches multi-étapes en autonomie. Elle hérite de toutes les fragilités de l'IA générative, avec en plus la démultiplication des risques par la composition. Cette distinction a une conséquence contractuelle concrète. Un fournisseur de LLM ne peut pas garantir la reproductibilité d'une génération : c'est une propriété exclue par la nature du produit. L'IA agentique apporte enfin des moyens de contrôler les artefacts produits par l'IA elle-même : revue automatisée, tests générés, vérification de conformité. Ces contrôles ont un coût, et ce coût est lui-même une consommation de jetons dont ni le volume ni le tarif ne sont contractuellement fixés. Nous ne disposons, à la date de rédaction, d'aucune garantie sur l'un ni sur l'autre. C'est la raison pour laquelle le chapitre 5 traite la maîtrise du coût comme une discipline de pilotage, et non comme une hypothèse budgétaire.

1.4 Ce dossier n'est pas neutre

Nous assumons trois partis pris. Premièrement, nous parlons du **développement logiciel**, pas de « l'IA en général ». Cette restriction délibérée nous permet d'être précis là où le débat général est bruyant. Nous privilégions la **grammaire CMMI** parce que c'est un standard de fait de bonnes pratiques, et qu'elle permet de localiser précisément où la valeur se déplace. Les 13 domaines que nous retenons sont une sélection REDSEN, assumée et justifiée au §2.2 : le modèle en compte davantage. Troisièmement, nous refusons **l'opposition binaire** entre pro-IA et anti-IA : nous adoptons une position d'un praticien qui livre sous contrat, et qui doit répondre de ce qu'il signe.

Chapitre 2 — La valeur ne disparaît pas, elle se déplace

Sept bénéfices, dont un conditionnel, une cartographie CMMI, et les zones où l'IA n'apporte rien

2.1 CMMI comme grammaire de lecture

Nous refusons deux discours symétriques : « l'IA change tout » et « l'IA ne change rien ». Ces deux positions sont paresseuses parce qu'elles n'obligent pas à préciser où, dans le cycle de développement, le changement opère.

Nous avons choisi CMMI comme grille d'analyse parce le modèle est reconnu comme un standard de facto, et qu'il est suffisamment fin pour localiser le déplacement de valeur domaine par domaine.

2.2 Cartographie des 13 domaines × IA générative

La vue *Development* du modèle CMMI v3.0 compte davantage de *practice areas* que les 13 retenues ici, notamment DAR, Planning, Monitor & Control et Process Asset Development. Nous opérons une **sélection REDSEN de 13 domaines** considérés comme pertinents pour cadrer la discussion IA-développement en 2026 : ils représentent les zones où la question de l'usage de l'IA se pose de manière opérationnelle immédiate. Les domaines exclus (DAR, Planning, Monitor & Control, Process Asset Development) sont soit peu discriminants à ce stade de maturité IA, soit traités indirectement via les 13 retenus. Cette sélection est assumée comme un choix pédagogique et non comme un jugement sur l'exhaustivité du modèle CMMI.

Parmi les 13 domaines examinés, sept subissent une transformation notable grâce à l'IA générative. Les six autres ne sont que légèrement ou pas du tout affectés. La classification est **argumentée** ; chaque cas dispose d'une justification explicite :

Domaine CMMI	Impact IA	Justification
Technical Solution (TS)	Fort	Cœur de la génération de code : boilerplate, migrations, scripts. Ratio effort/valeur démontré par Cui et al. [14] et ANZ [19].

Domaine CMMI	Impact IA	Justification
Verification and Validation (VV)	Fort	La génération massive de cas de test est aujourd'hui opérationnelle sur tâches répétitives (tests unitaires, fuzz élémentaire).
Peer Reviews (PR)	Fort	Second regard IA sur code humain : détection de patterns anti-idiomatiques et vulnérabilités connues, avec revue humaine finale.
Requirements Development and Management (RDM)	Fort	Analyse sémantique d'exigences et reconstitution de matrices de traçabilité : cas d'usage à gain élevé sur des documents volumineux.
Estimation (EST)	Fort	Assistance par comparaison à historique : l'IA accélère la première passe. Le jugement final reste humain — l'impact IA porte sur la préparation, pas sur la décision.
Risk Management (RSK)	Fort	Identification et classification de risques à partir d'un catalogue documentaire : l'IA est efficace pour élargir la couverture initiale. La priorisation reste humaine.
Configuration Management (CM)	Fort	Documentation de commits, analyse de dépendances, détection de dérives de configuration : gains structurels déjà industrialisés.
Governance (GOV)	Faible	Les décisions de direction engagent la responsabilité juridique de l'organisation. L'IA peut préparer un dossier, elle ne peut pas décider ni signer.
Product Integration (PI)	Faible	Assemblage et validation d'interfaces physiques et applicatives : l'IA aide à générer du code de test d'intégration, mais l'arbitrage d'architecture d'assemblage reste humain.
Managing Performance and Measurement (MPM)	Faible	Le choix des indicateurs de performance est un acte de gouvernance : l'IA peut proposer, elle ne peut pas décider ce qui sera mesuré.

Domaine CMMI	Impact IA	Justification
Process Management (PCM)	Faible	La définition et l'amélioration des processus organisationnels relèvent de la culture d'entreprise, pas de la génération textuelle.
Causal Analysis and Resolution (CAR)	Faible – jugement expert	L'analyse de causes racines exige de tenir ensemble des signaux techniques, humains et organisationnels. L'IA peut suggérer des pistes ; l'expertise humaine reste indispensable pour éviter les fausses causes plausibles.
Supplier Agreement Management (SAM)	Faible	La négociation contractuelle est une activité de lecture de signaux faibles et de mise en jeu de la relation – elle relève du jugement humain intégral (voir §2.4).

Le partage Fort/Faible ne porte aucun jugement de qualité sur les domaines eux-mêmes : il traduit notre appréciation, en 2026, de l'endroit où la génération textuelle probabiliste apporte un levier significatif. Un domaine classé « Faible » reste critique pour la réussite d'un projet ; il signifie simplement que, pour nous l'investissement IA n'y produira pas de gains de productivité significatifs à court terme.

La cartographie n'est pas figée. Un domaine « peu impacté » aujourd'hui peut basculer si un nouvel outil apparait. C'est une photographie d'août 2026, à réévaluer annuellement. Elle a une conséquence directe sur le pilotage : concentrer l'investissement IA sur les sept domaines à fort impact, et concentrer l'investissement humain sur les six domaines résistants.

Cette lecture évite deux erreurs symétriques. La première consiste à déployer l'IA partout, en pensant qu'elle apportera partout de la valeur : elle produit alors du bruit sur les six domaines résistants, sans gain mesurable. La seconde consiste à cantonner l'IA à un seul domaine (souvent la génération de code), en manquant les gains structurels sur la revue par les pairs, la traçabilité des exigences ou la documentation de configuration.

2.3 Sept bénéfices, dont un conditionnel

Bénéfice 1 — Accélération. L'IA générative produit du code fonctionnel plus vite qu'un humain sur les tâches standardisées : boilerplate, tests unitaires, scripts d'intégration, migrations de format. Le gain se mesure sur les tâches où la spécification est claire et le résultat vérifiable rapidement. Il s'effondre sur les tâches complexes, ambiguës ou sur les bases de code que le développeur maîtrise déjà mieux que l'IA.

Bénéfice 2 — Le contrôle qualité augmenté (bénéfice conditionnel). Le bénéfice existe lorsque l'IA sert de second regard en revue de code systématique (détection de patterns anti-idiomatiques, tests manquants, vulnérabilités connues), pas comme moteur de génération autonome. La position prudente est de considérer que l'IA ne dispense d'aucun contrôle qualité classique et **en impose souvent d'en ajouter** (linting renforcé, revue humaine obligatoire sur code généré, seuils de duplication surveillés).

Bénéfice 3 — Validation. L'IA permet de générer massivement des cas de test qu'un ingénieur seul n'aurait pas le temps de produire. Le bénéfice est réel sur la couverture. Le risque est que le développeur cesse de réfléchir aux cas limites qui n'apparaîtront pas dans la génération. La validation demeure une compétence d'ingénieur, pas une fonctionnalité d'outil. Une équipe qui délègue intégralement la conception des cas de tests à l'IA perd la capacité de détecter les régressions non typiques.

Bénéfice 4 — Idéation. L'IA élargit l'espace des solutions envisagées en début de conception. C'est une partenaire de brainstorming plus disponible qu'un pair, plus rapide qu'une revue de littérature. Ce bénéfice se dégrade si le développeur choisit la première proposition au lieu d'utiliser la génération comme un miroir. Une bonne pratique observée en atelier consiste à générer au moins trois options et à arbitrer, plutôt qu'à accepter la première.

Les trois bénéfices qui suivent portent moins sur le geste de production que sur ce qui l'entoure : qui peut faire le travail, ce que l'on garde de sa trace, et à qui on sait le raconter.

Bénéfice 5 — Accessibilité. L'IA rend accessibles aux profils juniors des tâches qui exigeaient auparavant une expertise. C'est un vrai bénéfice pour l'organisation, à condition de préserver la construction d'expertise sur les zones critiques. Un junior qui court-circuite systématiquement l'apprentissage construit une compétence artificielle qui s'effondre lorsque l'IA est indisponible ou incorrecte.

Bénéfice 6 — Traçabilité. L'IA aide à documenter le code, les commits, les décisions techniques. La traçabilité produite par l'IA n'a de valeur que si elle est vérifiée : sinon elle documente ce que l'IA imagine avoir été fait, pas ce qui a été fait.

Bénéfice 7 — Communication. L'IA facilite la production de documents adressés à des non-techniques (métier, régulateur, auditeur). Il permet à une équipe technique de produire des livrables lisibles par les métiers, ce qui améliore le dialogue et raccourcit les cycles de décision.

Ces sept bénéfices, dont le second est explicitement conditionnel, ne s'additionnent pas mécaniquement. Une organisation qui capture trois bénéfices sur sept avec discipline produit davantage de valeur qu'une organisation qui tente les sept sans cadre. La question ne porte pas sur le nombre de bénéfices visés, mais sur ceux que l'organisation capture durablement.

2.4 Où la valeur ne se déplace pas

Trois zones résistent structurellement.

Le jugement architectural. Choisir une architecture pour 10 ans, décider d'un pattern d'intégration : ces décisions ne bénéficient pas de la génération. Elles bénéficient d'une conversation avec l'IA, utile pour explorer les options, mais la décision doit rester humaine, tracée, argumentée. Une architecture générée sans arbitrage humain constitue un actif orphelin dont personne ne peut défendre les choix six mois plus tard.

La négociation client-fournisseur. Comprendre ce que le client ne dit pas, arbitrer une clause contractuelle ou désamorcer un désaccord : l'IA peut aider à la préparation, mais elle ne peut pas remplacer. La négociation est une activité de lecture de signaux faibles, d'ajustement en temps réel, de mise en jeu de la relation. Elle relève du jugement humain intégral.

La responsabilité juridique et éthique. Un livrable signé par une entreprise engage cette entreprise. L'IA ne signe pas. Elle produit. Aucun outil n'externalise cette responsabilité, aucun contrat fournisseur ne la transfère intégralement, aucune clause de non-garantie n'exonère celui qui livre. Cette asymétrie est la contrainte structurelle de tout usage IA en contexte professionnel régulé.

2.5 Transition

La valeur se déplace, mais elle ne disparaît pas. Elle se concentre sur trois compétences : **savoir spécifier ce qu'on demande, savoir vérifier ce que l'on reçoit, savoir arbitrer entre plusieurs options générées**. Un ingénieur qui excelle sur ces trois compétences est plus productif avec l'IA. Un ingénieur qui les néglige devient dangereux.

Cette réalité ne se résout pas par un outil de plus. Elle appelle à une lecture lucide des risques que l'usage IA introduit dans une organisation, indépendamment de la qualité des outils choisis. C'est l'objet du chapitre suivant.

Chapitre 3 — Les nouvelles zones de risque

Six mécanismes, deux régimes de dette, un angle mort

3.1 Ce que ce chapitre soutient

Ce dossier défend l'usage de l'IA générative en atelier de développement. Il consacre un chapitre entier à ce qui se casse, à ce qui se paie tard, à ce qui ne se voit pas.

Nous soutenons trois choses, et ce chapitre les instruit.

La croyance en un gain est quasi universelle ; la mesure du gain est contextuelle. 9 professionnels sur dix utilisent l'IA au travail et plus de 8 sur dix estiment qu'elle a augmenté leur productivité [17]. Sur un protocole mesuré, des développeurs expérimentés travaillant sur des dépôts qu'ils maîtrisent ont été 19 % plus lents avec assistance, tout en croyant avoir été 20 % plus rapides [15]. Ces deux résultats ne se contredisent pas : ils décrivent le même phénomène à deux échelles.

L'assistance accélère la livraison et déstabilise la production dans le même mouvement. Les équipes qui l'adoptent le plus livrent plus vite, et connaissent davantage d'échecs de changement, de reprises et de délais de résolution [17]. L'idée que la vitesse compenserait l'instabilité, résumée en « échouer vite, corriger vite », n'est pas étayée par les données : l'instabilité continue de dégrader la performance produite et alimente l'épuisement. Nous en tirons la thèse qui traverse ce dossier, celle de l'**amplificateur** : l'assistance magnifie les forces des organisations dont le socle technique et culturel est solide, et magnifie les dysfonctionnements de celles qui n'en disposent pas.

Les risques ne se voient ni dans le code livré ni dans le planning. Ils apparaissent en aval : dette structurelle, instabilité de production, perte de maîtrise interne, dépendances coûteuses à défaire. Un projet qui a « bien fonctionné » sur 12 mois peut avoir accumulé une dette qui sera longue à résorber. C'est la raison d'être des six mécanismes du §3.4 : nommer ce qui ne se voit pas, lui associer un signal, et fixer le seuil à partir duquel il faut agir.

3.2 La redistribution du gaspillage : réduit, déplacé, créé

Le Lean identifie sept gaspillages, que Mary et Tom Poppendieck ont traduits en 2003 dans le vocabulaire du développement logiciel [20] : fonctionnalités superflues, attente, transferts, traitement superflu, changement de tâche, travail partiellement terminé, défauts. Le corpus Lean occidental a ajouté dans les années 1990 un huitième gaspillage : les talents inexploités. C'est le mécanisme le plus lent et difficile à inverser.

L'erreur d'analyse courante consiste à traiter l'IA comme un réducteur uniforme de gaspillage. L'erreur symétrique consiste à en faire un pur générateur de rebut, ce qui rendrait incompréhensible le fait, mesuré [17], que le débit augmente réellement.

La lecture juste est en trois temps. **Trois gaspillages sont réellement réduits. Plusieurs sont déplacés. Tous, sans exception, connaissent une forme de création ou d'aggravation.**

Le solde net n'est pas calculable en l'état, et nous ne prétendons pas le calculer. Les gains se mesurent en débit ; les coûts se paient en dette, en instabilité et en compétence. Ces unités ne s'additionnent pas. Le gain est immédiat, local et mesuré ; le coût est différé, systémique et non quantifié.

Gaspillage	Réduit	Déplacé (et où)	Créé ou aggravé
Fonctionnalités superflues	–	–	Volume de code sans plafond de coût marginal ; fonctionnalités et code défensif non demandés
Attente	Attente d'une compétence rare ; friction de démarrage sur page blanche	Vers la revue : le contrôle devient le goulot	Attente d'arbitrage sur du code que personne n'assume
Transferts	Passages de main vers un spécialiste pour tâches standard	Vers les bascules humain ↔ IA : re-contextualisation, vérification	Perte d'intention entre demande et sortie, non tracée
Traitement superflu	–	Du geste manuel vers la sollicitation : boilerplate, tests répétitifs, documentation de routine, migrations mécaniques	Effet bazooka – un modèle mobilisé pour une tâche triviale ; corvée de sélection d'outil (*tool sprawl*)
Changement de tâche	–	Vers la conduite de plusieurs sollicitations en parallèle : fragmentation de la bande passante	Charge de décision liée à la multiplication d'outils IA
Travail partiellement terminé	–	–	Code généré non relu, accumulé dans branches, brouillons, fils

Gaspillage	Réduit	Déplacé (et où)	Créé ou aggravé
Défauts	Erreurs de syntaxe, typages incorrects, oublis mécaniques	Du compilateur vers la production	Défauts sémantiques plausibles : cas limite manqué, exception métier absente. Rebut direct : code généré puis annulé après revue
Talents inexploités	–	–	Atrophie des compétences ; blocage de la constitution des juniors

Trois gaspillages présentent une réduction réelle (attente, transferts, défauts), et c'est ce qui explique la hausse de débit mesurée. Aucun n'est supprimé : chacun des trois est simultanément vecteur de création. Les cinq autres ne présentent aucune réduction. **Ce n'est donc pas une compensation, c'est une redistribution asymétrique** : le gain se concentre sur des postes visibles et mesurés, le coût sur des postes différés et non quantifiés.

Le rebut direct est le poste le plus facile à ignorer parce qu'il ne laisse pas de traces dans le produit livré. Sur nos projets pilotes menés en mode agentique, nous observons des volumes importants de code annulé après une seconde relecture [21]. Il s'agit de code incorrect, incohérent ou dangereux, qui n'aurait pu être détecté sans une relecture ligne par ligne. Aucun tableau de bord du marché ne compte les lignes annulées. On peut y voir une lacune d'outillage. On peut aussi remarquer qu'aucun éditeur n'a d'intérêt particulier à la combler.

Le débit gagné ne se convertit en valeur livrée que si le projet dispose d'une capacité d'absorption en aval : revue, tests automatisés, plateforme interne de qualité, petits lots. Un projet qui adopte l'IA sans cette capacité convertit son gain en instabilité : le débit monte, la date de livraison ne bouge pas.

3.3 Deux régimes de dette

Pourquoi deux projets qui ont accumulé la même dette technique se dégradent-ils, l'un lentement, l'autre d'un coup ? Parce qu'ils ne relèvent pas du même régime. La dette technique d'un projet IA-intensif n'obéit pas à une seule loi : deux régimes coexistent, et ils appellent des dispositifs opposés.

LE RÉGIME D'ACCUMULATION – L'EFFET ICEBERG

Ce qu'il recouvre. Duplication de code, tests fragiles, documentation périmée, dépendances non consolidées, couverture en baisse. Ce que les comités voient, stories fermées, commits et vélocité, constitue la partie émergée. La partie immergée croît proportionnellement à la vitesse de production.

Sa loi. Croissance continue, approximativement proportionnelle au débit. Résorption à coût à peu près linéaire.

Sa propriété décisive. Il est mesurable en continu, à condition d'installer l'instrument. Aux quatre métriques de livraison historiques s'en ajoute depuis 2025 une cinquième [17], le taux de reprise (*rework rate*), qui donne une visibilité directe sur le coût des défauts et des corrections non planifiées.

Ce que le régime appelle. Un indicateur de dette ajouté aux indicateurs de vélocité dans le tableau de bord. Sans lui, la vélocité ment par construction.

LE RÉGIME DE SEUIL – LE POINT DE BASCULE

Ce qu'il recouvre. Couplage architectural non voulu, hypothèses implicites jamais explicitées, dépendances dont plus personne ne connaît la raison, modules que l'équipe évite d'ouvrir. Rien de tout cela n'entrave le projet, jusqu'au jour où une refonte devient impossible.

Sa loi. Effet de seuil, pas de croissance exponentielle. Un système absorbe la dégradation sans effet visible, puis le coût marginal du changement franchit un seuil au-delà duquel toute modification devient disproportionnée.

Sa propriété décisive. Il n'est pas extrapolable. Les métriques du régime d'accumulation ne prédisent pas le franchissement du seuil.

Ce que le régime appelle. Une surveillance qualitative, distincte du tableau de bord. Signaux typiques : hausse du temps moyen de revue, hausse du taux de reprise, incidents « bizarres » plus longs à diagnostiquer qu'à corriger, inquiétudes récurrentes des développeurs seniors sur des zones identifiées, apparition de zones que l'équipe contourne. Aucun de ces signaux n'est décisif en soi, mais leur convergence l'est. Le dispositif utile est une revue trimestrielle d'architecture, contradictoire, avec un intervenant extérieur, plutôt qu'un métrique de plus.

3.4 Six mécanismes de risque

Nos retours d'expérience et notre veille nous ont conduits à structurer six mécanismes de risque.

Chaque élément est présenté sous forme concise : les événements, l'horizon où ils deviennent visibles, l'unique signal à installer si l'on ne peut en installer qu'un, la contre-mesure principale et son revers. En effet, aucune contre-mesure n'est gratuite et celles qui le paraissent n'ont pas encore été évaluées. **L'annexe C porte les fiches complètes** : l'ensemble des signaux et de leurs seuils, les tests provoqués, les contre-mesures détaillées.

Les chiffres figurant sur les fiches comportent trois composantes : des échelles de grandeur tirées de la littérature consultée, des tendances observées sur le terrain et documentées par la FinOps Foundation et les fournisseurs de cloud, ainsi que des estimations basées sur nos propres expériences de missions. Il ne faut pas considérer ces données comme des vérités absolues. Chaque entreprise doit établir sa référence à une époque précédant l'adoption, puis déterminer sa propre limite d'alerte en fonction d'un écart significatif par rapport à cette référence. Les fourchettes proposées sont des points de départ défendables, non des valeurs opposables.

. . .

MÉCANISME 1 – DETTE STRUCTURELLE ET ÉCART DE PRODUCTION

Le coût marginal de production du code s'effondre ; celui de sa maintenance ne bouge pas. Cet écart se manifeste par la quantité de code inutilement conservé, l'accumulation de branches et surtout un écart de flux de travail [22] : l'assistance accélère le prototypage, mais ralentit l'industrialisation. Un projet atteint 80 % de son résultat visible en trois semaines et consomme six mois sur les 20 % restants. C'est la dérive de planning la plus courante, et elle est systématiquement lue comme un échec d'estimation alors qu'elle est un effet d'outil. L'accumulation est mesurée, pas seulement déduite [16].

Horizon de visibilité	6 à 18 mois – 2 à 4 mois pour l'écart de flux.
Probabilité · Gravité	très élevée · moyenne.
Signal de tête	Ratio lignes produites/lignes livrées en production, par sprint.
Seuil d'alerte	Alerte au delà de 1,5; à recaler sur une ligne de base établie avant adoption [23].

Ce qu'on fait. Agir sur le rythme d'entrée plutôt que sur le stock : plafond de demandes de fusion ouvertes par développeur, discipline de petits lots [17], et estimation en deux phases distinctes, prototype et industrialisation, aux taux de productivité assumés comme différents. Le ratio produit/livré n'a de valeur que rapporté à une ligne de base établie sur

au moins trois sprints antérieurs à l'adoption ; sa lecture en valeur absolue, sans référentiel interne, n'établit rien.

Ce qu'elle coûte. Le plafond incite à fusionner pour libérer le quota, donc à relire moins bien : il ne s'installe jamais seul, toujours couplé au mécanisme 2. L'estimation en deux phases affiche un coût d'industrialisation.

Fiche complète : annexe C.1.

. . .

MÉCANISME 2 – DÉFAUTS SÉMANTIQUES ET DÉPLACEMENT DU GOULOT

Le code généré compile presque toujours, mais peut manquer un élément : un cas limite, une exception métier, une condition d'arrêt. Le défaut n'est plus une erreur arrêtée par le compilateur, c'est une erreur de sens qui ressemble à du bon code. Dans le même temps, le volume à relire augmente pendant que la capacité de relecture reste constante. Le temps économisé à l'écriture est réaffecté à la vérification, également appelée « **taxe de vérification** » [22], et la charge est transférée entre les personnes : un auteur peut créer en quelques minutes ce qui occupera un réviseur pendant des heures. Cette asymétrie n'apparaît dans aucune métrique individuelle.

Horizon de visibilité	3 à 9 mois.
Probabilité · Gravité	très élevée · élevée.
Signal de tête	Taux de reprise [17].
Seuil d'alerte	Se lit en tendance, avec le délai médian en revue – alerte si doublement sur un trimestre.

Ce qu'on fait. Déplacer la vérification vers l'auteur [17] : délivrer le retour automatisé au moment de l'écriture plutôt qu'au relecteur en aval.

Ce qu'elle coûte. L'automatisation de la revue déplace la confiance sans la fonder : une équipe qui voit passer un agent relit moins attentivement. La contrainte de taille est contournable par découpage : elle doit porter sur l'unité de sens plutôt que sur le nombre de lignes.

Fiche complète : annexe C.2.

. . .

MÉCANISME 3 – PERTE DE MAÎTRISE ET ASYMÉTRIE DE RESPONSABILITÉ

Une part croissante du système est produite par génération assistée, sans que l'équipe qui en porte la responsabilité contractuelle puisse encore expliquer pourquoi il fonctionne. Le point à tenir n'est pas la part de code générée, qui mesure la production et non la compréhension. Le point est **l'asymétrie** : le contrat engage l'équipe dans les mêmes termes, tandis que la capacité à corriger, à expliquer une décision à un auditeur ou à absorber une contrainte réglementaire nouvelle a diminué sans que rien ne le signale [22].

Horizon de visibilité	12 à 36 mois – se révèle à l'incident, à l'audit ou au départ d'un expert.
Probabilité · Gravité	Moyenne · élevée.
Signal de tête	Nombre de personnes capables d'intervenir sur chaque module critique, en tendance.
Seuil d'alerte	Une décroissance non expliquée par les départs – le signal le plus fiable dont on dispose.

Ce qu'on fait. Définir et défendre une **part maîtrisée en propre**, incluant nécessairement les modules critiques, et la vérifier par un exercice d'explicitation trimestriel plutôt que par déclaration.

Ce qu'elle coûte. Une part maîtrisée définie trop largement devient une fiction documentaire, plus dangereuse que son absence, puisqu'elle rassure. Il vaut mieux respecter 15 % plutôt que de déclarer 60 %.

Ce mécanisme ne se détecte pas par un indicateur qui monte. Aucune métrique ne croît régulièrement à mesure que la maîtrise se perd : la détection repose sur trois tests provoqués, décrits en annexe C.3, dont l'explicitation, à froid par un développeur tiré au sort.

Fiche complète : annexe C.3.

. . .

MÉCANISME 4 – ÉROSION DES COMPÉTENCES

Deux phénomènes distincts. Le déconditionnement des gestes d'abord : l'atelier mobilise un modèle pour des tâches qui ne le méritent pas, par continuité d'usage et sans décision, avec un coût unitaire négligeable qui devient un coût cumulé. Le paradoxe de l'expertise ensuite, plus grave [22] : abaisser la barrière d'entrée offre un filet de sécurité réel, mais

court-circuite la lutte productive nécessaire à la constitution d'une expertise. L'atrophie n'est pas d'abord une perte chez les seniors, c'est une non-constitution chez les juniors : à cinq ou dix ans, la compétence non assistée moyenne baisse par renouvellement, sans que personne n'ait rien désappris.

Horizon de visibilité	24 mois et plus – le mécanisme le plus lent et le plus difficile à inverser.
Probabilité · Gravité	élevée · élevée.
Signal de tête	Écart de performance entre seniors et juniors sur des tâches non assistées comparables.
Seuil d'alerte	En tendance annuelle – tout élargissement soutenu.

Ce qu'on fait. Du compagnonnage plutôt que de l'abstinence [22] : apparier les juniors à des seniors sur les décisions d'architecture assistées, réserver l'écriture manuelle aux composants complexes, protéger une part du temps sans assistance, évaluer la compétence non assistée au recrutement.

Ce qu'elle coûte. La journée sans assistance devient un rituel symbolique dès lors que son résultat n'est pas examiné. Une politique d'usage trop stricte produit du contournement par comptes personnels.

Fiche complète : annexe C.4.

. . .

MÉCANISME 5 – INTÉGRITÉ DE LA CHAÎNE LOGICIELLE

Seul mécanisme du chapitre susceptible de causer un incident majeur en quelques heures, et dont la probabilité augmente directement avec l'autonomie accordée aux agents, un agent considère comme une donnée tout ce qu'il lit, que ce soit un ticket de documentation, une dépendance ou une page Web. Un contenu rédigé pour ressembler à une instruction peut être suivi. La surface d'attaque n'est plus l'incitation de l'utilisateur, mais plutôt tout ce que l'agent est autorisé à lire, en fonction des droits dont il dispose.

Horizon de visibilité	Des jours, parfois des heures.
Probabilité · Gravité	Moyenne · très élevée.
Signal de tête	Taux d'approbation des demandes de validation.
Seuil d'alerte	Un taux proche de 100 % signale un contrôle devenu formel, pas un dispositif sain.

Ce qu'on fait. Moindre privilège, validation humaine sur toute action irréversible, séparation stricte entre contenu de confiance et contenu observé. Aucun de ces contrôles n'est vérifiable sans inventaire des droits ni journalisation reconstituable.

Ce qu'elle coûte. La validation systématique produit de la fatigue d'approbation : au bout de quelques centaines de validations, l'opérateur approuve par réflexe. Personne n'a jamais été mis en cause pour avoir cliqué trop vite sur « approuver ». C'est précisément le problème. Restreindre le périmètre des agents vaut mieux que multiplier les points de contrôle.

L'absence d'incident ne prouve rien ici. Un dispositif défaillant et un dispositif jamais attaqué produisent la même mesure : la détection repose sur quatre tests provoqués décrits en annexe C.5, dont l'injection contrôlée, à conduire hors production et sous protocole validé par la filière sécurité.

Fiche complète : annexe C.5.

. . .

MÉCANISME 6 – COÛT ET DÉPENDANCE ÉCONOMIQUE

Deux dynamiques liées. Le coût est facturé dans une unité, le jeton, qui ne correspond à aucune ligne budgétaire préexistante et dont le volume n'est pas spécifié par l'acheteur : à sollicitation identique, sur une même tâche et un même modèle, l'exécution la plus coûteuse consomme le double de la moins coûteuse [24], et nos propres relevés vont dans le même sens [21]. Les usages agentiques aggravent la question d'un ordre de grandeur : une tâche agentique consomme environ 3 500 fois plus de jetons qu'une sollicitation de raisonnement à un tour [24], en déclenchant une chaîne d'appels dont la longueur n'est pas déterminée à l'avance. En revanche, la captivité émerge de décisions rationnelles : des requêtes adaptées à un modèle, des intégrations conçues sur une interface, des données stockées chez un fournisseur. Cumulées sur douze à dix-huit mois, elles produisent un système dont la portabilité n'est plus réelle.

Horizon de visibilité	3 à 6 mois pour la dérive de coût – 12 à 18 mois pour la captivité.
Probabilité · Gravité	Élevée · moyenne.
Signal de tête	Écart entre consommation prévue et réelle, par sprint.
Seuil d'alerte	Alerte au-delà de 30 % [25], à 80 % du plafond pour l'enveloppe [26]

Ce qu'on fait. Enveloppe par équipe avec alerte à 80 % du plafond. Évaluation de différents scénarios sur la charge réelle de l'organisation et surtout exécution du **test de bascule annuel**.

Ce qu'elle coûte. Le plafond pousse l'usage vers les comptes personnels, et l'organisation perd simultanément la donnée et la maîtrise. La couche d'abstraction anti-captivité remplace une dépendance fournisseur par une dépendance de framework, et fuit précisément là où elle compterait.

Fiche complète, avec l'encart sur le fine-tuning propriétaire : annexe C.6.

. . .

3.5 L'angle mort : l'usage non mesuré

Prenons une direction des systèmes d'information qui resserre sa politique d'usage. Le trimestre suivant, sa consommation déclarée de jetons baisse de 20 % et sa production ne bouge pas. Les deux chiffres sont exacts, mais pris ensemble, ils ne veulent rien dire.

Tout le dispositif décrit dans ce chapitre repose sur des signaux mesurés à l'intérieur du périmètre de l'entreprise, et cette hypothèse cesse d'être vraie dès que le dispositif devient trop contraignant.

Un plafond de consommation, une politique d'usage gradué, un suivi par développeur, une restriction d'outils : chacune de ces mesures crée une incitation à basculer une partie de l'usage vers des comptes personnels, des abonnements individuels, des outils non déclarés. L'organisation perd alors la donnée et la maîtrise en même temps, et elle les perd précisément sur les usages que la mesure visait à surveiller. C'est l'inversion la plus coûteuse : le tableau de bord s'améliore pendant que la situation se dégrade.

Plusieurs conséquences de méthode en découlent.

Aucun signal de ce chapitre ne doit être lu en valeur absolue. Une baisse de consommation déclarée est ambiguë : elle peut signaler une discipline installée ou un basculement hors périmètre. Elle doit toujours être croisée avec un indicateur de production.

L'effet est le plus net sur les signaux les plus utiles. Le mécanisme 3 propose de mesurer la part des sollicitations portant sur le code de l'équipe elle-même : c'est le meilleur indicateur de perte de maîtrise dont dispose ce chapitre, parce qu'il est direct et non substitutif. Il est aussi le premier détruit par le contournement : une équipe qui interroge un assistant personnel sur son propre code produit exactement zéro dans le journal de l'entreprise. La qualité d'un signal et sa fragilité au contournement vont ici de pair, et le tableau de bord se dégrade d'autant plus qu'il était bon.

Des mesures correctives restrictives ne devraient être mises en place que si une option de remplacement est proposée. Une politique qui restreint sans fournir de solution de rechange accessible produit du contournement, mécaniquement. Deux des sept facultés qui accroissent les avantages de l'aide [27] sont la définition et la diffusion d'une position claire

sur l'IA, car l'incertitude entraîne des risques et une politique claire procure la sécurité nécessaire pour expérimenter. De plus, la qualité de la plateforme interne est cruciale, car elle fournit des voies optimisées pour que les avantages soient étendus, plutôt que de se perdre dans les goulets d'étranglement.

L'usage hors périmètre est le premier risque de sécurité. Il combine le mécanisme 5, fuite de données et absence de contrôle des dépendances, avec l'absence totale de traçabilité. Une entreprise qui ne sait pas quels outils sont réellement utilisés sur ses données ne peut répondre à aucun audit.

L'indicateur pertinent est donc l'écart entre la production observée et la consommation déclarée, et non la consommation prise seule. Un projet dont le débit augmente pendant que sa consommation déclarée stagne ne fait pas d'économies.

3.6 Priorisation

Un catalogue de risques ne dit pas par où commencer. C'est pourtant la seule question que pose un directeur des systèmes d'information. Les six mécanismes se hiérarchisent selon quatre axes : probabilité, gravité, délai de détection, coût de la contre-mesure.

Mécanisme	Probabilité	Gravité	Délai de détection	Coût de la contre-mesure	Horizon d'engagement
5. Intégrité de la chaîne logicielle	Moyenne	Très élevée	Jours	Moyen	**Premier mois**
6. Coût et dépendance	Élevée	Moyenne	3 à 6 mois	Faible	**Premier mois**
2. Défauts et goulot de revue	Très élevée	Élevée	3 à 9 mois	Élevé	**Premier trimestre**
1. Dette et écart de production	Très élevée	Moyenne	6 à18 mois	Moyen	**Premier trimestre**
3. Perte de maîtrise	Moyenne	Élevée	12 à 36 mois	Élevé	**Première année**
4. Érosion des compétences	Élevée	Élevée	24 mois et plus	Très élevé	**Première année**

L'ordre n'est pas celui de la gravité, c'est celui du délai de détection combiné au coût d'inaction. Le mécanisme 4 est probablement le plus grave à 10 ans ; il figure en dernière ligne parce que rien ne se dégrade irrémédiablement dans les 6 premiers mois et que sa contre-mesure coûte cher. Le mécanisme 6 figure en tête non par gravité, mais parce que sa contre-mesure est peu coûteuse et son délai de détection court.

3.7 Deux objections

« C'EST L'ARGUMENT DU COMPILATEUR »

L'objection est ancienne. La même argumentation, qui consiste à dévaloriser un outil, a été utilisée pour contester le compilateur, l'environnement de développement intégré, la collecte automatique de la mémoire, les forums de questions-réponses. À chaque fois, la compétence ne s'est pas perdue : elle s'est déplacée vers le haut. Pourquoi cette fois serait-elle différente ?

Trois différences, qui n'invalident pas l'objection, mais en limitent la portée.

L'abstraction est non déterministe. Un compilateur produit, pour une même entrée, une même sortie, dont les propriétés sont spécifiées. On peut cesser de comprendre l'assembleur parce que le compilateur, lui, ne se trompe pas de manière imprévisible. Un modèle génératif n'offre pas cette garantie : il faut donc conserver la compétence de vérification que les abstractions précédentes permettaient d'abandonner.

L'interface n'est pas stable, elle ne s'apprend donc pas. On apprend un langage, une bibliothèque, une norme, objets stables sur des années. La couche d'assistance change de comportement à chaque version de modèle. Il n'y a pas de compétence durable à constituer sur l'interface elle-même, seulement sur ce qu'elle produit.

Ce qui est délégué est le jugement, pas le geste. Les abstractions précédentes ont automatisé des gestes mécaniques et libéré du jugement. L'assistance générative propose des décisions de conception, donc du jugement, et la compétence déléguée est exactement celle qui permettrait d'évaluer la proposition.

Ces différences plaident pour une vigilance, pas pour un refus. Elles indiquent surtout où placer l'effort de formation : sur la vérification et l'arbitrage, non sur la production manuelle pour elle-même.

« VOS CONTRE-MESURES COÛTENT DE LA VÉLOCITÉ »

Elles en coûtent, et c'est l'objection la plus sérieuse qui soit opposée à ce chapitre. Nous ne la traitons pas ici, parce qu'elle n'appelle pas une réponse d'analyse, mais une réponse de conduite : la question n'est pas de savoir si ces dispositifs ralentissent, mais si le creux qu'ils produisent est transitoire, et à quelles conditions. Cette dynamique prend la forme d'une courbe en J [28], dont la remontée est conditionnée. Le chapitre 5 le détaille.

3.8 En synthèse

1. **L'IA est un amplificateur, pas un accélérateur.** Elle magnifie l'état existant du projet, forces comme dysfonctionnements. La question n'est pas « faut-il adopter », mais « quel est l'état du socle sur lequel on adopte ». C'est ce qui rend le diagnostic préalable non négociable.
2. **Le goulot se déplace vers la vérification, et la vérification est une charge transférée à d'autres personnes** que celles qui gagnent du temps. C'est, avec le mécanisme 1, l'un des deux mécanismes les plus probables, et le mieux documenté des deux. Sa contre-mesure est aussi la plus coûteuse à installer, ce qui explique qu'il figure au premier trimestre du tableau de priorisation et non au premier mois : il est prioritaire par la gravité, pas par la séquence.
3. **Le creux de performance initial est attendu.** La courbe en J décrit une contrainte de conduite bien plus qu'un argument de vente : tenir la trajectoire au-delà des premières déconvenues, et modifier le système de travail, sans quoi la remontée n'a pas lieu.

Chapitre 4 – Le référentiel de maturité REDSEN

Cinq paliers, cinq dimensions

La cartographie des domaines de pratique répondait à une question : où l'IA déplace-t-elle la valeur dans l'activité de développement ? Cette question en amène une autre : à quelles conditions cette valeur peut-elle être captée sans exposer l'organisation ?

Ce chapitre y répond par un instrument, et nous disons d'emblée où il s'arrête. À sa lecture, une organisation doit pouvoir dire où elle se situe sur cinq dimensions, quel palier elle vise sur chacune, et par quel artefact daté elle le prouverait à un tiers.

4.1 Adresser les risques (dimensions)

Le référentiel s'organise sur cinq dimensions. Chacune capture une question qu'une organisation utilisant l'IA en projet de développement doit pouvoir trancher, indépendamment de son secteur.

	Dimension	Question structurante
A	Responsabilité	Qui assume nommément les livrables produits avec assistance IA, et sous quelle politique ?
B	Maîtrise	Que contrôlons-nous des flux qui traversent le modèle, et des actions qu'il déclenche ?
C	Autonomie	Que savons-nous encore faire sans assistance ?
D	Portabilité	De qui dépendons-nous, et à quel coût de sortie ?
E	Mesure	Que savons-nous factuellement de notre usage et des bénéfices ?

Chacune de ces dimensions se rattache à un mécanisme du chapitre 3, à l'exception de la Responsabilité, qui procède du principe fondateur : **l'IA produit, un humain signe**.

Dimension	Mécanisme du chapitre 3	Ce que la dimension met sous contrôle
A Responsabilité	Principe fondateur, et mécanisme 3 pour sa face contractuelle	Qui signe, sous quelle politique, avec quelle voie de refus
B Maîtrise	Mécanisme 5 – intégrité de la chaîne logicielle	Ce qui entre dans le modèle, ce que les agents peuvent atteindre

Dimension	Mécanisme du chapitre 3	Ce que la dimension met sous contrôle
C Autonomie	Mécanisme 4 – érosion des compétences	Ce que l'organisation sait encore produire sans assistance
D Portabilité	Mécanisme 6 – coût et dépendance économique	De qui l'organisation dépend, et à quel coût de sortie
E Mesure	Instrumentation transverse, et mécanisme 3 pour la cartographie	Ce que l'organisation sait factuellement de son usage

Ce que ce référentiel ne couvre pas, et pourquoi. Les mécanismes 1 (dette structurelle) et 2 (défauts sémantiques et goulot d'étranglement) sont les plus susceptibles de se produire, mais ils ne possèdent pas de dimension spécifique. C'est un choix de périmètre et non un oubli, ils relèvent de la qualité d'ingénierie logicielle, pour laquelle il existe déjà des référentiels que nous n'avons pas à refaire : la grammaire CMMI présentée au chapitre 2, et les sept capacités DORA. Ce référentiel-ci porte la **gouvernance de l'assistance**, c'est-à-dire les questions qu'aucun référentiel d'ingénierie antérieur à l'IA générative ne pose. Une organisation qui progresse ici sans traiter par ailleurs sa dette et sa capacité de revue aura mis sous contrôle sa gouvernance et laissé filer sa qualité. Les deux chantiers se conduisent en parallèle, avec deux outils distincts.

4.2 Évaluer la maturité (paliers)

Une organisation a écrit sa charte d'usage. Elle est publiée, elle est claire, et personne ne saurait dire si elle est appliquée. Quel palier peut-elle revendiquer ? Les cinq paliers ci-dessous répondent à cette question et à elle seule. Ils ne mesurent pas la qualité des règles : ils mesurent ce qui, dans l'organisation, tient sans que personne ait à y penser.

	Palier	Ce qui caractérise l'organisation
1	Exploratoire	Chacun décide dans son coin. Rien de ce qui s'y pratique ne peut être présenté comme une politique.
2	Encadré	Les règles sont écrites. Leur application repose sur la bonne volonté.
3	Outillé	Les règles sont devenues des dispositifs qui fonctionnent sans intervention. Premier palier qui résiste à une vérification par un tiers.
4	Piloté	Les dispositifs produisent des mesures, les mesures alimentent des arbitrages tracés. L'organisation sait si ce qu'elle fait produit l'effet attendu.
5	Résilient	L'organisation a démontré en conditions réelles qu'elle absorbe la disparition d'un fournisseur ou de l'assistance elle-même.

Règle d'or. Aucun palier n'est meilleur en soi. Le palier cible se choisit selon la mission : le mauvais choix est le palier mal aligné, jamais le palier le plus bas. Une cible à 5 sur les cinq dimensions se rencontre d'ailleurs plus souvent dans les documents que dans les organisations.

4.3 Situer l'organisation (attendus)

Les paliers sont cumulatifs : un attendu satisfait au palier N reste exigible à tous les paliers supérieurs. La matrice n'énonce à chaque palier que ce qu'il ajoute au précédent.

Palier	A – Responsabilité	B – Maîtrise	C – Autonomie	D – Portabilité	E – Mesure
1 Exploratoire	Usage individuel, aucun engagement formalisé	Aucun contrôle systématique sur les flux ni sur les actions	Aucune zone critique identifiée	Fournisseur unique, aucune alternative évaluée	Aucun indicateur d'usage
2 Encadré	Politique écrite de revue humaine couvrant les livrables engageants, liste d'outils autorisés	Règles d'entrée écrites, journalisation des sorties, validation humaine des actions agentiques	Zones à maîtrise interne cartographiées et arbitrées	Alternative fournisseur identifiée, plan de sortie ébauché	Incidents tracés, coûts suivis mensuellement
3 Outillé	Registre nominatif des signataires, taux de revue mesuré sur le code et sur les livrables documentaires	Contrôles automatiques en entrée, tests systématiques sur sorties IA, périmètre d'action des agents borné techniquement	Exercice annuel de production sans assistance sur le périmètre critique, résultat consigné	Deux fournisseurs qualifiés techniquement, prompts portables sur le périmètre critique	Tableau de bord daté, coûts par équipe, alerte de dépassement
4 Piloté	Revue périodique des incidents de responsabilité au niveau exécutif, plan d'action tracé	Audit qualité annuel des sorties IA et des actions agentiques, plan d'action documenté	Écart de performance avec et sans assistance mesuré sur le périmètre critique, seuil d'alerte défini	Bascule fournisseur testée sur un périmètre représentatif, résultat consigné	Indicateurs de qualité et de dette suivis dans le temps, arbitrages tracés

Palier	A – Responsabilité	B – Maîtrise	C – Autonomie	D – Portabilité	E – Mesure
5 Résilient	Chaîne de responsabilité éprouvée sous incident réel ou simulé	Contrôles tenus lors d'une évolution brutale des usages : montée en charge, extension du périmètre agentique	Capacité démontrée à livrer le périmètre critique sans assistance	Bascule fournisseur exécutée en production sur le périmètre critique, sans interruption de service	Séries de mesures maintenues comparables à travers un changement d'outillage ou de fournisseur

Une organisation peut être palier 4 en Responsabilité et palier 3 en Portabilité. C'est un état acceptable pour autant qu'il réponde à la stratégie de maîtrise des risques de l'organisation.

LA PREUVE ATTENDUE

Un attendu satisfait se démontre par un artefact daté, jamais par une déclaration. C'est la règle que l'annexe A applique ligne à ligne dans sa colonne *preuve produite*, et elle vaut ici sans exception : **une case cochée sans artefact opposable est une case vide**.

Dimension	Artefacts qui font preuve	Ce qui n'en fait pas
A Responsabilité	Registre nominatif des signataires, politique de revue publiée et datée, taux de revue mesuré, liste des refus opposables	Une charte non diffusée, une intention de revue
B Maîtrise	Inventaire daté des droits détenus par chaque agent, journal reconstituable des sources lues et des actions engagées, configuration écrite du bornage agentique, comptes rendus de tests provoqués	Une politique de moindre privilège non vérifiée
C Autonomie	Cartographie de la part maîtrisée en propre, comptes rendus datés des exercices de production sans assistance, écart mesuré avec et sans assistance	Une part maîtrisée déclarée, mais jamais éprouvée
D Portabilité	Plan de sortie par fournisseur, compte rendu du dernier test de bascule exécuté avec sa date, clause de réversibilité signée	Un second fournisseur identifié, mais jamais qualifié

Dimension	Artefacts qui font preuve	Ce qui n'en fait pas
E Mesure	Tableau de bord daté incluant un indicateur de dette, registre des incidents, traces d'arbitrage	Une enquête de satisfaction interne sur l'usage de l'IA

La colonne de droite n'est pas une précaution rhétorique : ce sont les cinq substituts que nous rencontrons le plus souvent en évaluation. Ils ont tous en commun d'être sincères, ce qui les rend difficiles à écarter en réunion.

4.4 Choisir des paliers cibles (archétypes)

Le palier cible se choisit selon la mission. Pour illustrer le mécanisme, nous nous sommes prêtés à l'exercice de proposer des exemples appropriés aux enjeux des organisations que nous rencontrons en mission.

Archétype	Risque dominant	Dimension la plus exposée	Cible d'ensemble	Cible sur périmètre critique
DSI secteur public	Légalité, souveraineté, continuité, explicabilité	A – Responsabilité	Palier 3 – Outillé	Palier 4 – Piloté
Éditeur de solutions	Qualité produit, réversibilité, responsabilité envers les clients	B – Maîtrise	Palier 4 – Piloté	Palier 4, palier 5 sélectif
ESN	Risque contractuel, hétérogénéité des missions, engagement de livraison	A – Responsabilité + D – Portabilité	Palier 3 au niveau socle	Palier 4 sur les missions engageantes
DSI banque privée	Confidentialité, traçabilité, dépendance tiers, non-délégation	B – Maîtrise + D – Portabilité	Palier 4 – Piloté	Palier 4 renforcé, palier 5 sélectif
DSI < 30 personnes	Dispersion, dépendance à quelques personnes, gouvernance disproportionnée	D – Portabilité	Palier 2 robuste	Palier 3 ciblé

Trois lectures ressortent de cette grille.

La cible d'ensemble n'est pas la cible réelle. Une organisation vise un palier d'ensemble pour tenir un socle collectif ; elle relève ce palier sur les activités où l'exposition le justifie. Une DSI publique au palier 3 doit être au palier 4 sur les services aux citoyens, les données sensibles et les applications décisionnelles. Une ESN au palier 3 doit être au palier 4 sur les missions à engagement de résultat, données sensibles ou contraintes réglementaires. Une petite DSI au palier 2 doit être au palier 3 sur les 20 à 30 % du portefeuille où l'exposition est réellement élevée.

Le palier 5 n'est jamais une cible générale. Il a du sens pour les organisations qui font de leur maîtrise de l'assistance un facteur durable de positionnement : un éditeur qui industrialise sa confiance client, une banque privée qui structure une capacité IA différenciante, une ESN qui capitalise une doctrine opposable. Il n'est ni nécessaire ni rentable ailleurs.

La règle du minimum sur les activités critiques prévaut sur la moyenne. Une bonne maturité d'ensemble ne compense pas une faiblesse sur un système sensible, réglementé ou contractuellement exposé. C'est le principe qui donne sa valeur au profil de maturité présenté au §4.5 : cinq scores lisibles séparément, pas une note globale flatteuse.

Ces archétypes ne sont pas des cases. Une organisation peut relever de plusieurs (un éditeur avec une DSI interne, une ESN avec une filiale éditrice), auquel cas la cible se compose plutôt qu'elle ne s'unifie.

4.5 Profil de maturité

A4 · B3 · C2+ · D3 · E4. Une évaluation tient dans cette ligne : cinq scores, un par dimension, aucune note globale. L'absence de moyenne est un choix, parce qu'une moyenne masquerait exactement ce que le profil doit faire apparaître.

La règle de notation. Le niveau le plus élevé atteint sur une dimension est le **palier maximal où tous les critères sont remplis, avec des preuves à l'appui**, les paliers étant additionnés. La règle est volontairement sévère : un attendu manquant au palier 3 fixe la dimension au palier 2, quels que soient les attendus du palier 4 déjà satisfaits. Le score s'écrit alors sous la forme `2+`, la notation indiquant qu'une progression est engagée sans être acquise, et l'écart se documente en une ligne : quel attendu manque, quel artefact l'établirait.

Cette sévérité a une raison. Un référentiel destiné à être opposé à un client ou à un auditeur ne peut pas admettre qu'une organisation revendique un palier qu'elle ne tient pas sur l'un de ses attendus : c'est précisément là que l'auditeur regardera.

Un profil se lit en une ligne. Reprenons celui qui ouvre cette section. Cette organisation signe et mesure correctement, contrôle ses flux et ses agents à un niveau outillé, dispose de deux fournisseurs qualifiés, et ne sait pas démontrer ce qu'elle produirait sans assistance. Le `C2+` indique que la cartographie existe, mais que l'exercice annuel n'a pas encore été conduit. Le diagnostic tient dans la lecture : c'est l'Autonomie qui décide de sa trajectoire de l'année, et aucune autre dimension ne compense.

Le profil montre où investir en priorité, ce qui est acquis, ce qui progresse et ce qui stagne d'un exercice à l'autre. À la question « vous êtes à quel niveau ? », il répond en cinq valeurs.

Ce que le profil ne dit pas. Il porte sur l'organisation, pas sur un projet. Une organisation dont le profil d'ensemble est satisfaisant peut porter une mission critique très en deçà de sa cible, et c'est la raison de la cible par périmètre décrite au §4.4.

4.6 Conduire une évaluation (démarche)

La matrice énonce les attendus. La méthode d'évaluation détermine comment on constate qu'une organisation y satisfait, et c'est elle qui décide de la valeur du résultat.

Notre démarche repose sur trois principes.

L'unité évaluée est l'organisation. La maturité désigne ce qu'elle sait faire de façon répétable. La criticité d'un projet détermine le niveau d'exigence qu'on lui applique.

L'évaluation procède par échantillon. On sélectionne des activités représentatives des variantes de pratique, puis on consolide. Les règles d'échantillonnage décident de la validité du résultat.

Deux natures de preuve sont requises. Les artefacts établissent qu'un dispositif existe ; les entretiens établissent qu'il est appliqué.

4.7 À propos de ce référentiel

Un référentiel qui ne mesure pas la qualité d'ingénierie, dans un dossier dont le centre de gravité est ce que l'assistance dégrade : la contradiction est apparente, et elle mérite d'être levée avant qu'un lecteur ne la formule à notre place.

Ce référentiel est un outil de gouvernance interne et de dialogue externe. Il permet de situer une organisation dimension par dimension, de choisir les paliers cibles alignés sur sa mission, et de rendre compte de sa trajectoire.

Son objet se limite à l'atelier de développement logiciel, ce qui s'y produit, sous quelle responsabilité, avec quelles dépendances. Il ne porte pas la qualité d'ingénierie elle-même, pour la raison exposée au §4.1. Les modèles d'adoption généralistes couvrent quant à eux l'entreprise entière : l'*AI Adoption Maturity Model* publié le 8 juin 2026 par le SEI de Carnegie Mellon avec Accenture [29] évalue huit dimensions : stratégie d'organisation, personnel et culture, refonte des processus, risque et gouvernance, données, ingénierie, exploitation, écosystème. Celui-ci s'articule avec eux sur le seul périmètre des équipes de développement, et n'a pas vocation à s'y substituer.

Trois choix nous sont propres :

- L'ancrage **européen** : les attendus s'empilent avec les obligations qui s'imposent déjà à une organisation soumise au droit de l'Union. Un artefact produit au titre de l'un de ces textes satisfait souvent aussi un attendu de ce référentiel.

- La dimension **Autonomie** : nous mesurons ce que l'organisation sait encore faire sans assistance. C'est aussi celle dont on nous demande le plus souvent si elle est vraiment nécessaire.
- Le **profil de maturité** enfin, qui restitue cinq scores là où la plupart des référentiels consolident vers un niveau unique.

CORRESPONDANCE AVEC LES OBLIGATIONS EXISTANTES

Le tableau ci-dessous indique, dimension par dimension, quels artefacts déjà produits pour un texte européen ou une norme peuvent être réemployés. Il ne dispense d'aucune obligation : il évite de produire deux fois le même document.

Dimension	Texte ou norme applicable	Artefact réemployable
A Responsabilité	Règlement (UE) 2024/1689 (IA) [30] ; ISO/IEC 42001 [31]	Politique d'usage, registre des responsabilités, documentation du système
B Maîtrise	Règlement (UE) 2016/679 (RGPD) [32] ; directive (UE) 2022/2555 (NIS 2) [33]	Analyse d'impact, registre des traitements, mesures techniques et journalisation
C Autonomie	Aucun	–
D Portabilité	Règlement (UE) 2022/2554 (DORA, résilience opérationnelle du secteur financier) [34]	Stratégie de sortie, test de résilience, clauses contractuelles avec les prestataires TIC
E Mesure	ISO/IEC 42001 [31] ; règlement (UE) 2022/2554 [34]	Indicateurs de système de management, registre des incidents

La ligne vide de la colonne centrale est le point à retenir. Aucun texte n'oblige une organisation à préserver sa capacité de production sans assistance. La dimension Autonomie est celle qu'aucun régulateur ne viendra vérifier, ce qui en fait la première abandonnée sous contrainte de calendrier, et c'est la raison pour laquelle elle figure dans ce référentiel. Le chapitre 5 revient sur ce point au §5.4.

4.8 Pour conclure

Une organisation qui mesure ses cinq dimensions sait ce qu'elle tient. Une organisation qui n'en mesure que quatre finira par découvrir la cinquième au moment où elle en aura besoin.

Chapitre 5 — Conduire et porter la trajectoire

Par où commencer, à travers quel creux, sous quelle responsabilité

. . .

Deux questions décident du sort d'une trajectoire d'adoption. Comment tient-on une trajectoire dont les premiers mois consomment plus que les rendements qu'ils génèrent ? Et qui, dans l'organisation, porte les décisions que le référentiel a nommées ?

Elles se répondent mal séparément, et ce chapitre l'explique. **Une trajectoire ne tient pas par la volonté : elle tient par des porteurs nommés et par des artefacts datés.** Le creux se traverse parce que quelqu'un l'a annoncé avant qu'il ne survienne. L'artefact transforme le refus en position argumentée. La décision qu'aucun texte n'impose se prend parce qu'une personne physique en répond. Retirez le porteur, la trajectoire dérive ; retirez l'artefact, le porteur n'a qu'une intention à opposer.

Ce chapitre ne prescrit pas une organisation type. Il dit dans quel ordre s'y prendre, ce qui arrivera pendant, qui répond de quoi, et ce que rien n'oblige à décider.

5.1 Par où commencer

Cette section ne propose pas un calendrier. Elle propose un ordonnancement. Une organisation qui démarre sur ces sujets rencontre toujours les mêmes questions dans le même ordre, et se trompe rarement en respectant cet ordre. Le principe qui le gouverne tient en une phrase. **Ce qui est invisible ne peut être encadré.**

Cinq blocs se succèdent, chacun rendu possible par le précédent. Chaque bloc produit un artefact opposable, sans lequel le bloc suivant repose sur une hypothèse.

#	Bloc	Ce qu'on installe	Artefact opposable	Ce qui bloque le bloc suivant si on l'ignore
1	Voir	Inventaire des usages, inventaire des agents en production, journalisation reconstituable	Registre unique daté et signé	On charte des usages qu'on n'a pas cartographiés
2	Écrire	Liste d'outils autorisés, charte d'usage IA, politique de revue humaine, politique d'estimation en deux phases	Quatre documents courts publiés au même moment	Les règles restent des principes, aucun refus ne s'oppose
3	Outiller	Canal d'accès contrôlé, enveloppes budgétaires par équipe avec alertes, bornage technique du périmètre agentique	Canal en service, tableau de consommation daté, configuration d'agent écrite	La charte s'érode à la première échéance serrée
4	Rendre opposable	Clause d'usage IA type, accord de traitement des données, engagement de traçabilité	Trois clauses signées par la direction juridique	Ce qui vaut en interne ne vaut pas devant un client, un auditeur, un régulateur
5	Éprouver	Exercice de production sans assistance, test provoqué, bascule fournisseur documentaire	Trois comptes rendus consignés, dont au moins un mauvais	L'ensemble tient sur une hypothèse plutôt que sur une preuve

Trois lectures du tableau méritent d'être signalées.

L'ordre n'est pas négociable, le rythme l'est. Une organisation à socle solide engagera les trois premiers blocs en parallèle. Une organisation en constitution, composée majoritairement de profils juniors, ajoutera au bloc *Écrire* un cinquième document, l'appariement junior-senior sur les décisions d'architecture.

Rendre opposable vient après outiller, pas avant. Une clause qui promet ce que l'organisation ne contrôle pas en interne se retourne contre celui qui l'a signée le jour d'un litige. C'est la raison pour laquelle le bloc juridique succède aux blocs opérationnels, plutôt que d'ouvrir la marche comme la logique du contrat le suggérerait.

Le bloc *Éprouver* se conduit à cadence étalée. Les sept tests provoqués décrits à l'annexe C n'ont pas le même rythme : trimestriel pour l'explicitation, annuel pour la refonte à blanc, ponctuel et sous protocole validé pour l'injection contrôlée. Les lancer ensemble produit un volume d'exercices que l'organisation abandonnera au deuxième cycle ; le premier trimestre en retient un par mécanisme prioritaire, pas davantage.

CE QUE LA SÉQUENCE COÛTE

Un ordonnancement qui n'indique pas son coût n'est pas opposable à un comité qui engage la dépense. Nous donnons donc des ordres de grandeur, en distinguant deux natures d'effort que la plupart des dossiers confondent : **ce qu'il faut pour installer, et ce qu'il faut pour tenir**. C'est la seconde qui décide si le dispositif survit au deuxième trimestre.

#	Bloc	Mise en place	Charge récurrente
1	Voir	10 à 20 jours-homme, dont l'essentiel sur la journalisation	Environ 1 jour par trimestre, pour l'actualisation des inventaires
2	Écrire	5 à 10 jours-homme de rédaction	1 à 2 jours par an, à la revue annuelle
3	Outiller	15 à 30 jours-homme, très variable selon l'existant	Environ 2 heures par semaine, pour le suivi de consommation
4	Rendre opposable	5 à 10 jours-homme juridiques, mais 2 à 4 mois de circuit de validation	1 jour par an
5	Éprouver	5 jours-homme pour l'écriture des protocoles	10 à 15 jours par an, répartis sur l'année

Ces ordres de grandeur valent pour une organisation de développement de 50 à 200 personnes, et ils sont constatés en mission plutôt que mesurés sur un protocole : ils se calibrent, ils ne se transposent pas. Deux règles d'échelle les rendent lisibles ailleurs. La mise en place des blocs 2 et 4 est indépendante de la taille : quatre documents et trois clauses coûtent la même chose à 30 personnes qu'à 800. La charge récurrente des blocs 1, 3 et 5 croît avec le nombre d'équipes, pas avec le nombre de développeurs.

Le bloc 4 est le seul dont le délai excède largement la charge. Trois clauses représentent une dizaine de jours de travail juridique, étalés sur deux à quatre mois de circuit de validation. Il faut donc l'engager en parallèle du bloc 3 plutôt qu'à sa suite, faute de quoi il retardera le bloc 5 de plusieurs mois sans que personne n'ait travaillé davantage.

Le total répond à une objection que ce chapitre recevra. Sur cette référence, l'ensemble représente 30 à 75 jours-homme d'installation, puis une trentaine de jours par an, soit au moins deux mois-homme pour installer, et environ un huitième d'un temps plein pour tenir. C'est à comparer non pas à zéro, mais au coût d'un incident de production sur une chaîne dont personne ne sait reconstituer ce qu'un agent a lu, ou d'un audit auquel l'organisation ne sait pas répondre. Cette charge se compare aussi aux gains d'une chaîne de production plus efficiente.

Ce que la séquence installe, ce sont des artefacts. Chacun des cinq blocs se clôt sur une preuve datée plutôt que sur une intention, et c'est ce qui rend le suivant possible. Reste à savoir qui les produit, et ce que l'organisation traversera entre-temps.

Question ouverte. Lequel de ces cinq artefacts existe-t-il aujourd'hui chez vous, et lequel restez-vous incapable de produire à un tiers qui le demanderait ?

5.2 Le creux est attendu, la remontée ne l'est pas

Les contre-mesures décrites dans ce dossier coûtent de la vélocité. Elles en coûtent, et nous préférons le dire que le nier. Une revue systématique ralentit, un contrôle d'entrée ajoute une étape, un exercice de production sans assistance immobilise une journée d'équipe. La question n'est pas de savoir si ces dispositifs ralentissent, mais si le creux qu'ils produisent est transitoire.

Le programme DORA décrit cette dynamique sous la forme d'une **courbe en J** [28] : une phase de creux initial est attendue après déploiement, et la remontée est conditionnée à la redéfinition des processus ainsi qu'à la solidité des fondations d'ingénierie. Le cadre recommande d'évaluer plusieurs scénarios (prudent, réaliste, optimiste) plutôt que de retenir un chiffre unique de retour sur investissement.

Tenir la trajectoire pendant ce creux est un exercice politique autant que technique. L'écart de flux de travail documenté par DORA en mars 2026 [22] en donne la forme la plus courante : un projet atteint 80 % de son résultat visible en trois semaines, puis consomme six mois sur les 20 % restants. Le creux se manifeste donc au moment précis où la démonstration initiale a produit son meilleur effet, ce qui rend l'écart d'autant plus difficile à expliquer. **Notre recommandation est d'annoncer le creux avant qu'il ne survienne**, en le nommant dans le dossier d'investissement plutôt qu'en le découvrant au deuxième comité. Un creux annoncé est une trajectoire ; un creux constaté est un échec.

Deux conséquences en découlent, et elles se contredisent en apparence. La première est qu'un creux de performance après déploiement est un phénomène attendu et non un échec. Le renoncement au bout de trois mois est la manière la plus courante de perdre l'investissement, parce qu'il intervient exactement au point bas de la courbe, lorsque les coûts sont visibles et les bénéfices pas encore. La seconde est que la remontée n'a rien d'automatique. Elle est conditionnée à des changements du système de travail. Une organisation qui déploie les outils sans toucher à ses processus reste durablement dans le creux [17]. Le socle, une fois de plus, décide du résultat.

Reste à distinguer le creux de l'enlisement. Trois signaux le permettent, et ils se lisent au deuxième trimestre. Le taux de reprise a-t-il cessé de croître ? Le délai médian en revue s'est-il stabilisé ? A-t-on effectivement modifié un processus ? Lequel ? Lorsque trois signaux négatifs convergent, le sujet n'est plus l'outillage, mais l'absence de transformation opérationnelle. Sans correction des usages, de la gouvernance et des pratiques, poursuivre l'investissement détruit davantage de valeur qu'une suspension maîtrisée.

Scénario	Ce qui se passe	Ce qui le détermine
Prudent	Creux prolongé au-delà de 12 mois, remontée partielle	Outils déployés, processus inchangés, aval non renforcé
Réaliste	Creux sur 2 à 3 trimestres, remontée au-dessus du niveau initial	Revue et tests renforcés en parallèle du déploiement
Optimiste	Creux court, remontée nette	Socle déjà solide avant adoption, capacité d'absorption disponible

Notre recommandation est de présenter les trois au comité qui engage la dépense, plutôt qu'un chiffre unique que la première déconvenue rendra indéfendable.

Le creux ne se traverse pas par conviction, il se traverse par annonce. Un comité qui a reçu les trois scénarios avant l'engagement ne découvre pas un manquement au deuxième trimestre : il constate un point de la courbe qu'on lui avait décrite. Encore faut-il que quelqu'un ait signé cette annonce.

Question ouverte. À quel scénario votre socle vous condamne-t-il aujourd'hui, et qu'êtes-vous prêt à changer pour en sortir ?

5.3 Qui porte quoi

Un référentiel sans porteur reste un texte, et le principe tient en une phrase : **la responsabilité ne se délègue pas, les moyens d'y répondre si.** Une organisation où la direction des systèmes d'information décide seule est fragile, parce qu'elle fait porter à une fonction technique des arbitrages qui engagent l'entreprise. Une organisation où le dirigeant délègue sans comprendre l'est tout autant, parce qu'il signera des engagements dont il ne connaît ni les contrôles ni les recours.

La carte ci-dessous répartit les décisions sur 11 rôles.

Acteur	Ce qu'il porte	Ce qui ne se délègue pas	Artefact qui l'atteste
Dirigeant, personne physique	Signature des livrables engageants, posture publique en cas d'incident	La signature ; la parole	Registre des signataires, chaîne d'escalade écrite
Comité de direction DSI	Arbitrage vélocité/maîtrise, liste des refus, homologation du palier-cible	L'arbitrage annuel	Compte rendu daté, liste des refus, homologation

Acteur	Ce qu'il porte	Ce qui ne se délègue pas	Artefact qui l'atteste
Direction des systèmes d'information	Dispositifs socles, gouvernance de l'atelier, coordination juridique et achats	Rien : elle installe, elle n'arbitre pas le palier	Inventaires, journalisation, registre des incidents
Direction juridique	Clause d'usage IA type, accord de traitement, engagement de traçabilité	L'opposabilité	Trois clauses signées
Direction de projet	Refus outillés, semaine après semaine	L'opposition du plancher au cas d'espèce	Registre des refus et de leur escalade
Collaborateur	Signalement du doute, non-soumission, traçabilité substantive	La déclaration d'un usage substantiel	Trace d'usage sur les générations structurantes
Achats	Clause de réversibilité, arbitrage multi-fournisseurs	–	Clause négociée au contrat cadre
Ressources humaines	Exercice sans assistance, évaluation annuelle, accueil des juniors	–	Politique écrite, comptes rendus d'exercice
Comité de direction DSI, gestion de crise	Chaîne d'escalade	Son écriture avant l'incident	Procédure datée
Comité de direction DSI	Revue annuelle du profil de maturité	–	Trace de remontée régulière
Prestataire externe	Plancher de qualité, journalisation, compétence non assistée	–	Position écrite au contrat cadre

Encart — Huit questions clés

1. Quel est notre palier revendicable, et sur quelle dimension est-il le plus bas ?
2. Qui a signé le dernier livrable engageant produit avec assistance, et sous quelle politique ?
3. Que peut atteindre en production le plus privilégié de nos agents ?
4. Quand avons-nous livré pour la dernière fois un périmètre critique sans assistance, et qu'a donné l'exercice ?
5. Sur combien de fournisseurs savons-nous basculer, et quand l'avons-nous testé en conditions réelles ?
6. Quel écart existe-t-il entre notre consommation déclarée et notre production observée ?
7. Quel est notre taux de reprise, et depuis combien de temps le mesurons-nous ?
8. Quel refus a été opposé le mois dernier, par qui, et qu'est-il devenu ?

Chacune de ces questions appelle une réponse factuelle, datée, vérifiable. **Une réponse hésitante à l'une d'entre elles vaut diagnostic**, et il n'est pas nécessaire d'être ingénieur pour le constater.

LE JURIDIQUE REND OPPOSABLE CE QUE LE DSI APPLIQUE

Trois artefacts apparaissent dans les chapitres précédents comme des évidences disponibles : la clause d'usage IA type, l'accord de traitement des données articulé au règlement (UE) 2016/679 [32] et au règlement (UE) 2024/1689 [30], l'engagement de traçabilité. Ils ne le sont que si une fonction juridique les a négociés. **Sans le juridique, le DSI n'a que des dispositifs sans opposabilité.**

LE PRESTATAIRE EXTERNE APPELLE UNE POSITION ÉCRITE

Ce dossier étant écrit par une entreprise de conseil, la franchise s'impose. Quand une équipe extérieure livre du code assisté au sein d'une organisation cliente, quatre points doivent être tranchés au contrat cadre :

- qui porte le plancher de qualité : chef de projet du prestataire, DSI client, ou les deux, position la plus solide ;
- quel régime de journalisation s'applique ;
- qui décide si le prestataire livre le seul livrable final ou également le journal reconstituable de ses travaux ;
- quel régime de compétence non assistée s'applique dans l'équipe livrée, exigible sur les périmètres critiques et excessif sur des tâches d'intégration standard.

Encart — Trois positions à écrire au contrat cadre

Position exigeante. Le prestataire porte le même plancher que le client sur les cinq dimensions, journal reconstituable inclus, compétence non assistée démontrée sur périmètres critiques. Solide en litige, coûteuse en négociation, exigeante en équipe. Recommandée pour les prestations engageantes.

Position pragmatique. Le prestataire porte son propre plancher documenté, aligné aux exigences client par équivalence contractuelle plutôt que par identité de dispositif. Journal livré sur demande. Position des prestations standard, qui demande une réelle discipline documentaire côté prestataire.

Position défensive. Plancher négocié cas par cas, selon le périmètre et le régime de responsabilité. Position des prestations très courtes ou très encadrées, où le coût d'un cadre commun excède son apport.

Nous recommandons de retenir explicitement l'une des trois positions au contrat cadre, plutôt que de laisser l'ambiguïté nourrir un contentieux ultérieur. Ne rien écrire est le seul choix qui ne se défende pas.

CE QUE LA CARTE NE DIT PAS

Un acteur peut être désigné et la décision rester sans trace. **Le seul instrument que nous connaissions pour le vérifier est une matrice des décisions**, qui reprend une à une les décisions nommées dans ce dossier et demande, pour chacune, qui signe, qui arbitre, qui propose, et quel artefact daté l'établit. L'annexe A en fournit la trame, et dit ce que l'exercice fait apparaître. Nous ne la livrons pas remplie, parce que c'est l'exercice qui diagnostique et non le résultat.

Reste la question que la carte ne tranche pas : où arbitre-t-on, chez vous, ce que personne n'a mandaté ?

5.4 Les décisions que rien n'oblige à prendre

Six décisions engagent l'entreprise dans son ensemble. Elles se prennent en comité exécutif, se documentent et se revoient annuellement.

Décision	Ce qu'elle fonde	Ancrage réglementaire ou normatif
1 Souveraineté des données	L'éligibilité aux marchés régulés	Règlement (UE) 2016/679 [32]
2 Part maîtrisée en propre	La résilience opérationnelle	Règlement (UE) 2022/2554 (DORA) [34]
3 Stratégie fournisseur	La continuité de service	Règlement (UE) 2022/2554 (DORA) [34] ; directive (UE) 2022/2555 (NIS 2) [33]
4 Posture contractuelle client	Ce que l'entreprise garantit	Règlement (UE) 2024/1689 (IA) [30]
5 Plan de sortie et trajectoire de maturité	La capacité à changer d'avis	Règlement (UE) 2022/2554 (DORA) [34] ; ISO/IEC 42001 [31]
6 Compétences et autonomie	Ce que l'organisation saura faire dans dix ans	–

La colonne d'ancrage se lit sur sa case vide. Les cinq premières décisions s'adossent à un texte qui les rend, à des degrés divers, exigibles. La sixième n'en a aucun, et cette absence n'est pas fortuite.

Trois éléments se conjuguent pour rendre cette décision structurellement fragile. **Aucun texte européen ne l'oblige.** Aucun règlement, aucune directive, aucune norme technique ne demande à une organisation de préserver sa capacité de production sans assistance. **Son horizon d'effet dépasse le mandat.** Le mécanisme 4 du chapitre 3 est le plus lent des six : ses premiers effets ne sont pas visibles avant 24 mois, et son effet systémique, la baisse de la compétence non assistée, se compte en années. **Le mécanisme est documenté mais silencieux.** Contrairement à un incident de sécurité ou à une défaillance de conformité, la perte de compétence ne produit pas d'alerte instantanée ; elle se révèle au moment où l'organisation en a besoin, c'est-à-dire trop tard.

Question ouverte. Combien de vos décisions ne tiennent que parce que vous décidez de les prendre ?

Conclusion — Vers un usage raisonné de l'IA

. . .

6.1 Il y a des opportunités, mais il n'y a pas de magie

Ce dossier a consacré l'essentiel de ses pages à ce qui se casse, à ce qui se paie tard et à ce qui ne se voit pas. Il serait malhonnête de le refermer sans dire ce qu'il tient pour acquis du côté du gain.

Les opportunités sont réelles et elles sont mesurées. Le débit de livraison augmente, et cette hausse est établie par des essais contrôlés randomisés autant que par des enquêtes de terrain. Les tâches standardisées se produisent plus vite, la couverture de test s'élargit, l'espace des solutions envisagées en conception s'ouvre, et des livrables lisibles par des interlocuteurs non techniques deviennent accessibles à des équipes qui n'en produisaient pas. Une organisation qui refuse ce levier se prive de quelque chose de réel, et nous ne défendons à aucun moment l'abstention.

Il n'y a pas de magie pour autant, et c'est le second terme de la conviction. Un atelier de production qui relit, teste, découpe en petits lots et consigne ses décisions verra ces pratiques rendues plus productives. Un atelier mal structuré verra ses dysfonctionnements augmenter.

Cette formulation impose de relire tout ce qui précède dans un autre sens, parce que **ce que ce dossier appelle contre-mesure n'est pas une précaution défensive : c'est la condition du bénéfice.** La revue systématique, les tests automatisés, la plateforme interne, les petits lots, la consignation des décisions d'architecture et la compétence de vérification entretenue n'existent pas pour freiner l'assistance. Ils existent parce qu'ils sont ce qu'elle amplifie, et parce qu'un atelier qui ne les possède pas n'a rien à amplifier.

La conséquence de pilotage est simple et coûteuse : l'investissement rentable est le socle sur lequel l'outil opérera, avant l'outil lui-même. Ce socle a un contenu et un ordre, les cinq blocs du §5.1, et il commence par voir ce qui est déjà en place.

Plus d'industrialisation, plus d'expertise. Le volume produit dépasse ce qu'une revue à l'œil absorbe : c'est le mécanisme 2, l'un des deux plus probables du dossier (§3.4). Il appelle une chaîne dimensionnée pour ce volume — analyse statique, tests automatisés, vérification continue — où l'outil unique de mesure de qualité devient un étage parmi plusieurs.

Cette chaîne ne décide rien, elle rend visible ce qu'un humain doit trancher. Et ce que nous observons sur nos projets pilotes rend ce tri coûteux : le code généré a une facture d'apparence senior avec des défauts de débutant. Un défaut de sens qui ressemble à du bon code (annexe C.2) ne se repère pas en relecture rapide. Le repérer demande une expérience au moins équivalente à celle qu'exige l'écriture — c'est ce qui fait de la compétence de vérification un investissement, plutôt qu'une tâche qu'on délègue vers le bas.

6.2 L'opportunité la plus grande n'est pas la vitesse

Ceux d'entre nous qui exercent ce métier depuis 30 ans ont vu la chaîne de production logicielle se segmenter, méthodiquement, année après année.

Au départ, un ingénieur passait la matinée en atelier avec le métier et l'après-midi à réaliser ce qui venait d'être discuté. La compréhension du besoin et sa réalisation tenaient dans la même tête, et le délai entre les deux se comptait en heures. Ce que nous avons construit depuis ressemble à une chaîne taylorienne : un analyste métier rédige un référentiel d'exigences dont l'inventaire tient du poème de Prévert, un architecte définit un cadre, un chef de projet ordonnance. Et un développeur, souvent à plusieurs fuseaux horaires de là et recruté sur un différentiel de coût, implémente une fonctionnalité sans jamais avoir eu la compréhension générale du système auquel elle appartient.

Chacune de ces segmentations était rationnelle à l'époque où elle a été décidée. Le geste de production coûtait cher, et tout ce qui coûte cher se spécialise, s'industrialise, puis se déplace vers là où il coûte moins. La chaîne s'est allongée parce que chaque maillon supplémentaire se justifiait par un gain de coût unitaire.

Nous constatons chez nos clients ce que cette chaîne produit : des projets longs, plus chers que prévu, et difficiles à maintenir. La perte d'intention entre la demande et la réalisation ne figure sur aucune ligne budgétaire, mais elle se paie à chaque étape, et elle explique une part de la dette que le chapitre 3 décrit sans en nommer l'origine.

L'assistance générative effondre le coût du geste de production. Si produire coûte peu, la spécialisation extrême perd son fondement économique, et la chaîne qu'elle a rendue nécessaire devient un héritage plutôt qu'un choix.

Nous estimons que l'opportunité la plus grande n'est donc pas de faire la même chose plus vite, mais de retravailler l'organisation et les processus qui supportent les projets. Des équipes légères, proches du client, comprenant le système entier plutôt que le lot qui leur est confié. Des processus revus dans leur ensemble, et non améliorés maillon par maillon.

Nous formulons cette hypothèse comme telle, et aucune donnée de ce dossier ne l'établit. Elle est en revanche cohérente avec tout ce qui précède. Une équipe restreinte qui comprend son système est exactement celle qui peut revendiquer une part maîtrisée en propre, expliquer une décision d'architecture six mois plus tard, et absorber en aval ce

qu'elle produit en amont.

Le risque symétrique existe, et il est au moins aussi probable. L'assistance peut tout aussi bien approfondir la segmentation : un développeur encore plus éloigné du métier, produisant encore plus vite du code qu'il ne comprend pas, dans une chaîne qui s'allonge d'un maillon d'outillage. L'amplificateur joue dans les deux sens, et rien dans la technologie ne désigne la bonne direction. Le sens dans lequel elle jouera est une décision d'organisation, et non une conséquence de l'outil.

6.3 Nous n'avons traité qu'un maillon

Une dernière honnêteté est due au lecteur, et elle porte sur le périmètre même de ce dossier.

Le problème que nous cherchons à résoudre n'est pas de produire du code plus vite. Il est de livrer au métier des solutions qui lui servent, dans un délai qui garde la demande pertinente et à un coût que l'entreprise assume. Le développement n'en est qu'un moyen, et c'est sur ce moyen que ce dossier a concentré son attention.

Or, ce moyen est un maillon parmi d'autres. Entre le moment où un besoin apparaît et celui où la solution correspondante est utilisée, il y a la formulation de la demande et son arbitrage dans un portefeuille. Il y a ensuite sa spécification, sa réalisation, sa validation, sa mise en production et son exploitation. La réalisation est une station de cette chaîne, et elle n'en est ni la première, ni la dernière, ni nécessairement la plus lente.

La théorie des contraintes, formulée par Eliyahu Goldratt [35], énonce une règle que nous considérons comme exacte, mais inconfortable. Le débit d'un système est fixé par sa contrainte, et une amélioration obtenue ailleurs qu'à la contrainte n'améliore pas le débit du système. Une heure gagnée sur une station qui n'est pas la contrainte n'est donc pas une heure gagnée : c'est un gain local, visible sur son indicateur et absent du résultat d'ensemble.

Nous en tirons une conséquence directe pour tout ce qui précède. Une organisation qui met sous contrôle son développement assisté a sécurisé une station de sa chaîne. Sa contrainte réelle peut parfaitement se situer ailleurs, dans le délai d'arbitrage de son portefeuille, dans la disponibilité de ses experts métier, dans une chaîne de validation qui se compte en semaines ou dans une mise en production trimestrielle. Son délai de livraison au métier ne bougera pas pour autant, et le soin apporté à la production de code n'y changera rien.

Le cas est plus sévère encore lorsque la contrainte se situe en aval de la réalisation. Accélérer une station placée avant la contrainte ne produit pas du débit, mais de l'en-cours. En développement logiciel, cet en-cours porte un nom que ce dossier a déjà rencontré : du code écrit et non livré, qui vieillit, qui diverge de son environnement et dont le coût de

reprise croît avec le temps d'attente. **L'assistance générative est un moyen efficace d'alimenter une file d'attente**, et une organisation qui accélère sans savoir où se trouve son goulot fabrique de la dette en croyant produire de la valeur.

La première question n'est donc pas de savoir comment accélérer le développement, mais où se trouve la contrainte ? Une organisation qui l'ignore et qui investit malgré tout dans son atelier fait un pari, dont l'issue la plus probable est le financement d'un confort local. Cette question se traite avec les instruments de la théorie des contraintes et non avec ceux de ce dossier. Ces instruments tiennent en cinq gestes : identifier la contrainte, l'exploiter au maximum, y subordonner le reste de la chaîne, l'élever si cela ne suffit pas, puis recommencer parce qu'elle se sera déplacée.

Ces maillons ne seront pas épargnés, et le nôtre ne l'a pas été. L'assistance y apportera la même dualité qu'en développement, des gains réels et des mécanismes de risque qui leur sont attachés. Nous n'en instruisons aucun ici, mais l'un d'entre eux est déjà visible.

Le coût de production s'effondre pour les documents comme il s'effondre pour le code. Un dirigeant nous rapportait récemment être noyé sous les comptes rendus de réunion produits avec l'IA, dans un volume que plus personne dans son organisation ne lit. Ce témoignage est isolé et nous le donnons comme tel, sans lui prêter la valeur d'une mesure, mais la mécanique qu'il décrit est exactement celle du mécanisme 1 du chapitre 3.

La réponse qui vient spontanément est de demander à l'assistance de résumer ces comptes rendus, et cette réponse devrait nous arrêter. Elle produit un document que personne n'a écrit, résumé par personne et destiné à personne. Chaque maillon de la chaîne documentaire garde sa forme et perd sa fonction, sans qu'aucune étape puisse être désignée comme fautive.

Ce que le compte rendu coûtait autrefois n'était pas la frappe, mais l'arbitrage. Quelqu'un devait décider ce qui méritait d'être retenu, et cette sélection était la valeur du document bien plus que sa transcription. En supprimant le coût, l'assistance supprime aussi l'arbitrage, et elle laisse derrière elle un artefact complet et sans intention.

Ce cas illustre exactement la règle que nous venons de poser. Lorsque le coût de production s'effondre et que le coût de lecture reste ce qu'il est, la contrainte du système se déplace vers l'attention humaine. Produire davantage n'améliore alors plus rien, et résumer davantage ne desserre pas la contrainte : cela ajoute une station à une chaîne déjà saturée. La contre-mesure est une décision plutôt qu'un outil supplémentaire : combien de ces documents doivent exister, et qui en répond.

6.4 L'écart va se creuser

Nous prenons ici une position qui engage, et qui n'est pas confortable.

Les bons deviendront très bons. Les très bons deviendront excellents. Et les moyens, laissés sans aide, s'appauvriront.

Cette prédiction n'est pas une figure de style, elle découle des mesures citées dans ce dossier. L'enquête DORA 2025 [17] nomme l'assistance *the great amplifier* et établit ce que ce terme recouvre : l'IA magnifie les forces des organisations dont le socle technique et culturel est solide, et magnifie les dysfonctionnements de celles qui n'en disposent pas. Une organisation qui possède déjà une culture de revue installée capte donc le bénéfice là où une autre doit apprendre à relire en même temps qu'elle apprend à générer. L'écart de départ ne se comble pas, il se multiplie.

L'écart se creusera aussi entre les organisations qui saisiront l'occasion de simplifier et celles qui outilleront leur chaîne existante sans y toucher. Les premières réduiront le délai entre le besoin et sa réalisation ; les secondes ajouteront de la vitesse à un circuit qui en perdait déjà l'essentiel en chemin.

Le même mouvement opère entre les personnes, sous une forme plus difficile à voir. Les gains individuels immédiats sont les plus élevés chez les développeurs les moins expérimentés, et c'est là qu'ils sont les plus trompeurs : c'est ce qui masque l'appauvrissement. La non-constitution décrite au mécanisme 4 joue ici à l'échelle de notre domaine d'activité.

Deux lectures de cette position sont possibles, et une seule est la nôtre. La lecture fataliste conclut que le sort en est jeté, que les organisations solides prendront le large et que les autres décrocheront. Nous la rejetons, pour une raison qui tient à un seul mot de la phrase.

« **Sans aide** » **est le seul terme de cette prédiction sur lequel une direction peut agir.** Quatre variables décident de quel côté de l'écart une équipe se retrouvera. Le compagnonnage entre profils juniors et confirmés, d'abord, et le temps protégé de production sans assistance. La revue traitée comme une capacité à financer plutôt que comme un coût à comprimer, pour terminer, l'évaluation explicite de la vérification et de l'arbitrage. Elles ne coûtent pas cher au regard de ce qu'elles préservent, et aucune n'exige d'attendre une version suivante du modèle. Trois relèvent de la direction des ressources humaines et une de la direction de projet ; toutes se vérifient sur l'indicateur du mécanisme 4, l'écart de performance entre seniors et juniors sur des tâches non assistées comparables.

Ce que nous refusons, ce sont les postures binaires. L'organisation qui interdit se prive d'un levier établi, quand celle qui adopte sans cadre laisse l'amplification travailler contre elle. La voie que nous défendons est étroite et praticable : adopter par palier, tracer chaque décision, refuser ce qui compromet, mesurer ce qui compte, et tenir la trajectoire au-delà du creux.

Annexe A — Matrice des décisions

Trame à remplir

Cette annexe fournit la trame annoncée au §5.3. La bibliographie numérotée figure à l'annexe B, les fiches complètes des six mécanismes à l'annexe C, la méthode et la déclaration d'intérêts à l'annexe D, le glossaire à l'annexe E. Elle est livrée vide, et c'est délibéré : le diagnostic ne se lit pas dans son contenu, il se produit en la remplissant.

Comment la remplir

Un nom de fonction par case, pas un service. « La DSI » n'est pas une réponse ; « le directeur des systèmes d'information » en est une.

Signer désigne la personne physique qui engage l'organisation et qui répondra devant un client, un auditeur ou un régulateur. Une seule par ligne, jamais une instance collégiale.

Arbitrer désigne qui tranche lorsque deux exigences s'opposent, typiquement la vélocité contre la maîtrise. Peut être une instance.

Proposer désigne qui instruit le dossier et formule la recommandation. C'est presque toujours renseigné, et c'est ce qui rend les deux colonnes précédentes trompeuses lorsqu'on remplit vite.

Preuve produite désigne l'artefact daté qui établit que la décision a été prise et appliquée : un registre, un compte rendu consigné, une clause signée, un tableau de bord daté. Une intention n'est pas une preuve. Un document non daté non plus.

Une ligne sans nom en colonne *Signer*, ou sans artefact en colonne *Preuve produite*, se laisse vide. **Une case vide correctement laissée vide vaut mieux qu'un nom de complaisance**, et c'est précisément ce que l'exercice cherche à faire apparaître.

A — Responsabilité

Décision	Signer	Arbitrer	Proposer	Preuve produite
Signature d'un livrable engageant produit avec assistance				

Décision	Signer	Arbitrer	Proposer	Preuve produite
Politique de revue humaine : code et livrables documentaires				
Liste des refus opposables et de leur voie d'escalade				
Arbitrage annuel entre vélocité et maîtrise				
Homologation du palier cible, d'ensemble et par périmètre critique				
Posture publique et chaîne d'escalade en cas d'incident				

B – Maîtrise

Décision	Signer	Arbitrer	Proposer	Preuve produite
Liste des outils autorisés et canal d'accès contrôlé				
Périmètre d'action des agents en production, borné techniquement				
Inventaire des droits détenus par chaque agent				
Journalisation reconstituable des sources lues et des actions engagées				
Protocole des tests provoqués et validation par la filière sécurité				

Décision	Signer	Arbitrer	Proposer	Preuve produite
Contrôle des dépendances nouvellement introduites par génération				
Détection de secrets en amont de l'envoi				

C – Autonomie

Décision	Signer	Arbitrer	Proposer	Preuve produite
Périmètre de la part maîtrisée en propre, et sa cartographie				
Calendrier et périmètre de l'exercice de production sans assistance				
Appariement junior-senior sur les décisions d'architecture assistées				
Évaluation de la compétence non assistée au recrutement et en revue annuelle				
Part de temps protégée sans assistance				

D – Portabilité

Décision	Signer	Arbitrer	Proposer	Preuve produite
Stratégie fournisseur et qualification d'un second fournisseur				

Décision	Signer	Arbitrer	Proposer	Preuve produite
Plan de sortie documenté, par fournisseur				
Date, périmètre et exécution du test de bascule				
Souveraineté et localisation des données				
Décision de spécialisation d'un modèle, et versionnement du corpus				
Clause de réversibilité au contrat cadre				

E – Mesure

Décision	Signer	Arbitrer	Proposer	Preuve produite
Indicateurs retenus au tableau de bord, dette comprise				
Enveloppes budgétaires par équipe et seuils d'alerte				
Registre des incidents et cadence de revue				
Profil de maturité annuel sur les cinq dimensions				

Ce que la matrice révèle

Deux résultats se produisent presque toujours, et ils sont plus instructifs que le reste du tableau.

Une ou deux lignes se révèlent sans décideur identifié. Elles se concentrent en Autonomie et en Portabilité, les deux dimensions dont aucune fonction ne porte spontanément la charge, parce que leur horizon d'effet dépasse l'exercice budgétaire.

Plusieurs cases de la colonne « preuve produite » restent vides alors que les trois premières colonnes sont renseignées. C'est la configuration la plus fréquente et la plus coûteuse : la décision a un porteur, elle n'a pas de trace. Ces cases vides sont le diagnostic le plus rapide dont dispose ce dossier.

Une matrice remplie sans case vide, au premier passage, appelle un second passage.

Annexe B — Bibliographie

Sources numérotées par ordre de première apparition

Chaque entrée porte son niveau de preuve, selon la grille posée en annexe D : **A** étude publiée au protocole décrit et réplicable ; **B** enquête sectorielle large ou télémétrie d'éditeur, corrélationnelle ; **C** observation interne ou ordre de grandeur relayé. Les textes réglementaires et normatifs ne portent pas de niveau : ils ne sont pas des sources de mesure.

Les URL ont été vérifiées à la date d'édition.

. . .

Avant-propos

1. NVIDIA, *NVIDIA Blackwell Platform Arrives to Power a New Era of Computing*, communiqué du 18 mars 2024 et keynote GTC — revendications d'un facteur 25 sur le coût et la consommation d'énergie et de 30 sur la vitesse d'inférence, sur un scénario spécifique d'inférence GPT-MoE 1,8 T comparant FP4 sur GB200 NVL72 à FP8 sur H100. https://nvidianews.nvidia.com/news/nvidia-blackwell-platform-arrives-to-power-a-new-era-of-computing

2. Adrian Cockcroft, *Deep dive into NVIDIA Blackwell benchmarks - where does the 4x training and 30x inference speedup come from?* — détaille la dissymétrie FP4/FP8 de la démonstration GTC. L'auteur est ancien vice-président architecture cloud d'AWS. C'est l'élément qui porte la démonstration de l'avant-propos, indépendamment des amplitudes. https://adrianco.medium.com/deep-dive-into-nvidia-blackwell-benchmarks-where-does-the-4x-training-and-30x-inference-0209f1971e71

3. MLCommons, *MLPerf Inference*, versions v5.1 (septembre 2025) et v6.0 (avril 2026) — suite de benchmarks à protocole publié, résultats soumis par les constructeurs et revus par les pairs. DeepSeek-R1 est entré dans la suite en v5.1, gpt-oss 120B en v6.0. https://mlcommons.org/2025/04/mlperf-inference-v5-0-results/ https://mlcommons.org/2026/03/mlperf-inference-gpt-oss/ *Niveau A : protocole public, résultats vérifiables, soumissions revues. La fourchette citée dans l'avant-propos couvre deux versions successives de la suite, ce que le texte indique.*

4. SemiAnalysis, *InferenceMAX*, à partir d'octobre 2025 — banc d'essai d'inférence ouvert, exécuté en continu, couvrant les piles vLLM, SGLang et TensorRT-LLM. Renommé **InferenceX** depuis. Mesure notamment gpt-oss 120B et Llama 3.3 70B. Mesure directement le débit par mégawatt : sur gpt-oss 120B en FP4, environ 0,9 million de jetons/s/MW sur H100 contre 2,8 millions sur B200, soit un facteur voisin de 3 en interactivité courante et jusqu'à 7 à 180 jetons/s/utilisateur. Ces relevés portent sur des ser-

veurs à huit cartes, non sur la baie GB200 NVL72 dont l'annonce de l'entrée 1 revendique le facteur 25. C'est la source primaire du chiffre d'énergie par jeton cité à l'avant-propos. https://inferencex.semianalysis.com/ *Niveau B — vendeur-neutre et reproductible, mais les optimisations logicielles mesurées ont été conduites en collaboration avec le constructeur, ce qui rend les gains sensibles à la version de la pile testée.*

5. The Software Frontier, *The Blackwell Migration Question* - reprise et mise en perspective des mesures d'énergie par jeton de l'entrée 4, que l'article crédite explicitement. Citée pour la lecture qu'elle propose de la migration Hopper vers Blackwell, la mesure elle-même relevant de l'entrée 4. https://www.thesoftwarefrontier.com/p/the-blackwell-migration-question *Niveau B, source de second rang : l'article ne conduit pas de mesure propre.*

6. Yann LeCun, entretien au *Financial Times*, janvier 2026 — reconnaissance publique que les résultats de Llama 4 avaient été « fudged a little bit » et que des versions différentes du modèle avaient servi à des tests différents. LeCun a quitté Meta en novembre 2025. Source primaire sous péage : *Financial Times*, janvier 2026. Reprises en accès ouvert : https://finance.yahoo.com/news/yann-lecun-meta-fudged-little-100000402.html https://tech.slashdot.org/story/26/01/02/1449227/results-were-fudged-departing-meta-ai-chief-confirms-llama-4-benchmark-manipulation

7. TechCrunch, *Did xAI lie about Grok 3's benchmarks?*, 22 février 2025. https://techcrunch.com/2025/02/22/did-xai-lie-about-grok-3s-benchmarks/

8. Epoch AI, *OpenAI and FrontierMath*, 2025 — financement du benchmark par OpenAI, non communiqué aux mathématiciens contributeurs ni au public. https://epoch.ai/latest/openai-and-frontiermath

9. Bean, Mahdi et Rocher (Oxford Internet Institute, université d'Oxford), *Measuring what matters*, NeurIPS 2025 — revue systématique de 445 benchmarks, dont 16 % seulement emploient des tests statistiques rigoureux. Actes NeurIPS 2025 : https://papers.neurips.cc/paper_files/paper/2025/file/1967e0fc3aa6cbbace562f5cb8e3954e-Paper-Datasets_and_Benchmarks_Track.pdf Préprint : https://arxiv.org/html/2511.04703v1 Communiqué OII : https://www.oii.ox.ac.uk/study-identifies-weaknesses-in-how-ai-systems-are-evaluated/ *Attribution vérifiée : l'étude est bien conduite par l'Oxford Internet Institute. Le site oxrml.com est celui du Reasoning with Machines Lab, laboratoire hébergé par l'OII.*

10. Revelio Labs, données d'effectifs Anthropic — effectif passé d'environ 1 300 à environ 3 400 personnes en neuf mois. https://www.reveliolabs.com/companies/anthropic-pbc/employees *Page dynamique, consultée le 17 août 2026. À redater à chaque édition.*

11. Anthropic, page carrières, consultée le 5 août 2026. https://www.anthropic.com/careers/jobs

. . .

12. METR, *Uplift update*, 24 février 2026 — réexamen de l'étude de juillet 2025. Le suivi n'a permis ni de confirmer ni d'infirmer le −19 %: estimation ponctuelle de −18 % sur les développeurs revenus dans l'étude, intervalle de confiance [−38 %, +9 %], que les auteurs qualifient eux-mêmes de signal peu fiable. Le protocole d'origine est devenu difficilement reproductible, 30 à 50 % des développeurs refusant désormais de travailler sans assistance. https://metr.org/blog/2026-02-24-uplift-update/ *Niveau A. Citée conjointement à l'entrée 15 partout où celle-ci est mobilisée.*

13. Peng, Kalliamvakou, Cihon et Demirer (GitHub, Microsoft Research), *The Impact of AI on Developer Productivity: Evidence from GitHub Copilot*, 2022 - 95 développeurs recrutés, implémentation d'un serveur HTTP en JavaScript, 55,8 % plus rapides (71 minutes avec Copilot contre 161 sans), p = 0,001 7, intervalle de confiance à 95 % [21 %, 89 %]. https://arxiv.org/abs/2302.06590 https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/ *Niveau A, portée très étroite : tâche synthétique, hors base de code historique, sans mesure de qualité ni coût de revue aval.*

14. Cui, Demirer, Jaffe, Musolff, Peng et Salz, *The Effects of Generative AI on High-Skilled Work: Evidence from Three Field Experiments with Software Developers* — document de travail 2024, révisé 2025, paru dans *Management Science* en 2026. Trois essais contrôlés randomisés chez Microsoft, Accenture et une entreprise anonyme du Fortune 100, 4 867 développeurs ; +26,08 % de tâches complétées, erreur type 10,3 %, effet plus marqué chez les développeurs les moins expérimentés. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4945566 — DOI 10.2139/ssrn.4945566 https://economics.mit.edu/sites/default/files/inline-files/draft_copilot_experiments.pdf *Niveau A. Source d'origine partiellement fournisseur, recevable au critère posé en annexe D : protocole publié, limites documentées.*

15. METR, *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*, juillet 2025 - 16 développeurs, 246 tâches, ralentissement mesuré de 19 % contre une attente de +24 % et une perception ex post de +20 %. Outillage Cursor Pro (Claude 3.5 puis 3.7 Sonnet), dépôts matures de plus d'un million de lignes. https://metr.org/blog/2025-07-10-early-2025-ai-experienced-os-dev-study/ https://arxiv.org/abs/2507.09089 *Niveau A, portée étroite. Amplitude non reconfirmée par l'entrée 12.*

16. GitClear, *The Maintainability Gap: AI Code Quality in 2026* — 623 millions de changements de code analysés entre 2023 et le premier semestre 2026. Séries et bases au §1.2. Mesure sur commits réels, non déclarative. Seule mesure directe du régime d'accumulation mobilisée dans ce dossier, indépendante du corpus DORA. https://www.gitclear.com/the_ai_code_quality_maintainability_gap *Niveau B.* ***Divergence assumée****: GitClear publie −70 % sur le refactoring quand l'arithmétique directe sur 21 % → 3,8 % donne −82 %. La divergence figure telle quelle dans les publications de l'éditeur et n'y est pas réconciliée ; l'explication la plus probable est une normalisation différente. Le §1.2 signale l'écart plutôt que de trancher.*

17. DORA, *State of AI-assisted Software Development*, septembre 2025 (Google Cloud) — enquête auprès de près de 5 000 professionnels ; adoption associée à une hausse conjointe du débit et de l'instabilité ; thèse de l'amplificateur ; sept profils d'équipe ; introduction du taux de reprise comme cinquième métrique. https://dora.dev/research/2025/dora-report/ *Niveau B — déclaratif, corrélationnel, programme opéré par Google Cloud. Recevable au critère posé en annexe D : protocole publié, limites documentées, conclusions défavorables à un déploiement naïf.*

18. Microsoft Research, *AI and Productivity Report — First Edition*, décembre 2023 — écart entre perception (environ 36 minutes gagnées par jour) et mesure objective (environ 12 minutes) ; score de qualité de 12,40/15 sans Copilot à 11,06/15 avec Copilot. Périmètre : tâches d'écriture et de synthèse en environnement Microsoft 365, **non** développement logiciel. https://www.microsoft.com/en-us/research/wp-content/uploads/2023/12/ai-and-productivity-report-first-edition.pdf *Niveau A pour ce qu'il mesure, hors périmètre du développement. La transposition opérée au §1.2 est explicitement présentée comme une extrapolation de mécanisme, pas comme une mesure.*

. . .

Chapitre 2

19. Chatterjee, Liu, Rowland et Hogarth (ANZ Bank), *The Impact of AI Tool on Engineering at ANZ Bank*, février 2024, publié à SEC 2024 (ACM/IEEE Software Engineering in Practice) — phase expérimentale contrôlée sur 100 ingénieurs pendant 6 semaines, de mi-juin à fin juillet 2023, sur des défis de programmation Python ; gain de 42,36 % (17,86 minutes avec Copilot contre 30,98 sans), effet significativement supérieur pour la sous-population experte. Déploiement ultérieur sur environ 1 000 ingénieurs, population totale d'environ 5 000. https://arxiv.org/pdf/2402.05636 *Niveau A, portée étroite : défis standardisés hors base de code historique. C'est la source qui contredit l'entrée 14 sur le profil qui tire le plus de bénéfice, contradiction traitée au §6.4.*

. . .

Chapitre 3

20. Mary et Tom Poppendieck, *Lean Software Development: An Agile Toolkit*, Addison-Wesley, 2003 — traduction des sept gaspillages Lean dans le vocabulaire du développement logiciel. *Référentiel terminologique, pas une source de données.*

21. Observations internes REDSEN — variance de consommation entre exécutions répétées ; rebut après revue sur projets assistés ; retours de mission. Corroborent les ordres de grandeur de l'entrée 24 sans en constituer une réplication. *Niveau C.*

22. DORA, *Balancing AI tensions: Moving from AI adoption to effective SDLC use*, mars 2026 — analyse qualitative de 1 110 réponses d'ingénieurs Google recueillies au troisième trimestre 2025 ; taxe de vérification, paradoxe de l'expertise, écart de flux de travail. https://dora.dev/insights/balancing-ai-tensions/ *Niveau B, réserve de transposition : population Google, non transposable mécaniquement à une entreprise de services numériques française.*

23. GitHub Copilot Research et Bae Seokjae — mesures publiées du taux de survie du code accepté. GitHub rapporte une rétention de caractères de l'ordre de 88 % 7 jours après acceptation ; l'analyse méthodologique de Bae Seokjae (juin 2026) documente que la survie effective au niveau des lignes en commit tombe à environ 20 %, contre 35-40 % d'acceptation au niveau suggestion. La divergence recouvre deux métriques : rétention de caractères acceptés (numérateur : caractères conservés dans le fichier ouvert) contre survie en commit (numérateur : lignes conservées dans un commit publié). L'intervalle 20 à 88 % correspond à un ratio produit/livré de l'ordre de 1,1 à 5. Source mobilisée au §3.4 (mécanisme 1) pour borner le seuil d'alerte sur le ratio lignes produites/livrées. https://github.blog/ai-and-ml/github-copilot/the-road-to-better-completions-building-a-faster-smarter-github-copilot-with-a-new-custom-model/ https://baeseokjae.github.io/posts/ai-coding-accepted-code-quality-2026/ *Niveau B — télémétrie d'éditeur pour GitHub, analyse méthodologique tierce pour Bae Seokjae ; les deux mesures sont convergentes sur le mécanisme, divergentes sur l'amplitude en raison des définitions retenues.*

24. Longju Bai, Zhemin Huang, Xingyao Wang, Jiao Sun, Rada Mihalcea, Erik Brynjolfsson, Alex Pentland et Jiaxin Pei, *How Do AI Agents Spend Your Money? Analyzing and Predicting Token Consumption in Agentic Coding Tasks*, prépublication arXiv, 24 avril 2026, révisée le 29 avril 2026 — 500 instances SWE-bench Verified, quatre exécutions indépendantes, huit modèles de frontière sous OpenHands. Une tâche agentique consomme environ 3 500 fois plus de jetons qu'une sollicitation de raisonnement à un tour et 1 200 fois plus qu'un échange conversationnel ; les jetons d'entrée dominent le coût ; à tâche et modèle identiques, l'exécution la plus coûteuse consomme le double de la moins coûteuse. La difficulté estimée par des experts s'aligne mal sur le coût réel, et les modèles ne prédisent pas correctement leur propre consommation. Source mobilisée au §3.4 et à l'annexe C.6 (mécanisme 6). https://arxiv.org/abs/2604.22750 *Niveau A, trois réserves : prépublication non revue par les pairs ; banc SWE-bench Verified constitué de tickets GitHub publics, non transposable mécaniquement à un projet d'entreprise ; l'un des co-auteurs est employé par All Hands AI, éditeur du cadre agentique mesuré — au sens du §D.3, le protocole est publié, les limites sont documentées et les conclusions sont défavorables à l'émetteur.*

25. FinOps Foundation, *Forecasting Capability* et *Cloud Cost Forecasting Playbook* — échelle de maturité publique sur la variance entre consommation prévue et réelle. Cibles par palier : Crawl ≤ 20 %, Walk ≤ 15 %, Run ≤ 12 %. Le *Playbook* affine à cinq niveaux (Level 0 > 20 %, Level 1 < 20 %, Level 2 < 15 %, Level 3 < 12 %, Level 4 < 5 %). Source mobilisée au §3.4 (mécanisme 6) pour situer le seuil de 30 % retenu par ce dossier

comme délibérément plus permissif que la cible cloud mature, en tenant compte de la variance intrinsèque des sorties LLM à sollicitation constante documentée au §3.4 notes [21] et [24]. https://www.finops.org/framework/previous-capabilities/forecasting/ https://www.finops.org/wg/cloud-cost-forecasting/ *Référentiel de pratique, pas une source de mesure empirique.*

26. AWS, Google Cloud et Microsoft Azure — documentation officielle des seuils d'alerte budgétaire. AWS Budgets utilise 80 % comme exemple canonique dans sa documentation Budget Actions ; Google Cloud Billing applique une escalade par défaut à 50 %, 90 %, 100 % ; la pratique Azure documentée recommande une escalade 50 %, 75 % et 90 % en actuel doublée d'un seuil forecasted à 80 %. Source mobilisée au §3.4 (mécanisme 6) pour attester que le seuil de 80 % retenu par ce dossier correspond à la pratique par défaut des trois principaux fournisseurs cloud. https://docs.aws.amazon.com/cost-management/latest/userguide/budgets-action-configure.html https://docs.cloud.google.com/billing/docs/how-to/budgets https://zeeshan.net/build-alerts-for-azure-budget-thresholds-a-practical-azure-finops-guide/ *Documentation fournisseur pour AWS et GCP ; guide de pratique tiers pour Azure. Référentiel opérationnel, pas une source de mesure empirique.*

27. DORA, *AI Capabilities Model*, décembre 2025 (Google Cloud) — rapport compagnon de l'entrée 17 ; source des sept capacités listées au §3.5. https://services.google.com/fh/files/misc/2025_dora_ai_capabilities_model.pdf *Niveau B.*

28. DORA, *ROI of AI-assisted Software Development*, v.2026.1, 22 avril 2026 — modèle explicite de courbe en J ; trois coûts de mise en œuvre nommés (courbe d'apprentissage, taxe de vérification, adaptation de la chaîne) ; approche par scénarios prudent, réaliste et optimiste, le scénario par défaut donnant 39 % de retour la première année et 8 mois de seuil de rentabilité. https://dora.dev/ai/roi/report/ https://services.google.com/fh/files/misc/dora-roi-of-ai-assisted-software-development-2026.pdf *Niveau B.*

. . .

Chapitre 4

29. Software Engineering Institute (Carnegie Mellon University) et Accenture, *AI Adoption Maturity Model v1.0*, 8 juin 2026 — huit dimensions : stratégie d'organisation, personnel et culture, refonte des processus, risque et gouvernance, données, ingénierie, exploitation, écosystème. Périmètre entreprise entière, distinct de celui du référentiel présenté au chapitre 4. https://www.sei.cmu.edu/library/ai-adoption-maturity-model/

30. Règlement (UE) 2024/1689 du Parlement européen et du Conseil établissant des règles harmonisées concernant l'intelligence artificielle — obligations applicables aux fournisseurs de modèles à usage général entrées en application le 2 août 2025 ; orientations de la Commission sur le cas de la modification en aval. https://eur-lex.europa.eu/eli/reg/2024/1689/oj *Consulté le 17 août 2026. À redater à chaque révision : le régime est applicable mais son interprétation reste instable.*

31. ISO/IEC 42001:2023 — système de management de l'intelligence artificielle.

32. Règlement (UE) 2016/679 (RGPD). https://eur-lex.europa.eu/eli/reg/2016/679/oj

33. Directive (UE) 2022/2555 (NIS 2). https://eur-lex.europa.eu/eli/dir/2022/2555/oj

34. Règlement (UE) 2022/2554 sur la résilience opérationnelle numérique du secteur financier, dit **DORA**. https://eur-lex.europa.eu/eli/reg/2022/2554/oj *À ne pas confondre avec le programme de recherche DORA opéré par Google Cloud, entrées 17 à 28. La convention de dénomination du dossier est posée en annexe D.*

. . .

Clôture

35. Eliyahu M. Goldratt, théorie des contraintes — *The Goal*, 1984, et *Theory of Constraints*, 1990. Les cinq étapes de focalisation citées au §6.3 en sont issues. *Référentiel de gestion, pas une source de mesure empirique.*

. . .

Annexe C

36. Faros, *AI Engineering Report 2026: The Acceleration Whiplash*, avril 2026 — télémétrie sur environ 22 000 développeurs et plus de 4 000 équipes, comparant pour chaque organisation ses périodes de plus faible et de plus forte adoption. Chiffres mobilisés à l'annexe C.2 (mécanisme 2) : ratio incidents/demandes de fusion +242,7 %, délai médian en revue +441,5 %, demandes fusionnées sans revue +31,3 %. https://www.faros.ai/research/ai-acceleration-whiplash *Niveau B, éditeur intéressé. Les amplitudes sont citées comme variations intra-organisation mesurées sur ce jeu de données, jamais comme prévision transposable à une autre organisation. L'éditeur précise que ses jeux 2025 et 2026 sont des coupes indépendantes et que la comparaison entre les deux vaut pour la direction, non pour l'ampleur. Note de lecture : ce rapport conteste explicitement la thèse de l'amplificateur de l'entrée 17, en soutenant que la maturité d'ingénierie amont ne protège pas de la dégradation aval. Ce dossier mobilise Faros pour les amplitudes en aval et DORA pour la thèse de l'amplificateur, sans arbitrer entre les deux — un choix éditorial assumé, l'un reposant sur de la télémétrie et l'autre sur du déclaratif.*

. . .

Annexe C — Fiches complètes des six mécanismes

Le référentiel de détection

Le chapitre 3 présente les six mécanismes sous forme abrégée : ce qui se produit, l'horizon auquel cela devient visible, le signal à surveiller en premier, la contre-mesure principale et son revers. Cette annexe porte les fiches complètes : l'ensemble des signaux et de leurs seuils, les tests provoqués, les contre-mesures détaillées et les effets pervers de chacune. Chaque fiche se referme sur ses **porteurs**, distingués selon qu'ils installent le dispositif, l'appliquent au quotidien ou en arbitrent le périmètre : la carte complète des onze acteurs figure au §5.3.

Elle se consulte, elle ne se lit pas d'un trait. On y revient quand un signal du chapitre 3 s'allume, ou quand il faut instruire un mécanisme avant d'engager une contre-mesure.

Une précision sur la rubrique « signaux ». Les mécanismes 1, 2, 4 et 6 relèvent du régime d'accumulation : ils produisent des indicateurs continus, qui montent à mesure que le risque s'installe. Les mécanismes 3 et 5 relèvent du régime de seuil décrit plus haut : ils ne produisent pas d'indicateur qui croîtrait proportionnellement à l'exposition. Leur rubrique distingue donc les **signaux observés**, faibles et à lire par convergence, des **tests provoqués**, qui portent l'essentiel de la détection. Les dispositifs nécessaires à ces tests (inventaires, cartographies, journalisation) figurent en contre-mesure, à leur place : leur absence n'est pas un signal du risque, c'est une incapacité à le voir.

. . .

C.1 — Mécanisme 1 — Dette structurelle et écart de production

Mécanisme. Le coût marginal de production du code s'effondre ; celui de sa maintenance ne bouge pas. L'écart alimente trois accumulations. Les fonctionnalités superflues, d'abord : générer 3 000 lignes coûte le même prix marginal que 300, et la tentation est de tout conserver « puisque c'est gratuit ». Le travail partiellement terminé ensuite : code généré, non relu, non maîtrisé, qui s'accumule dans des branches et des brouillons. Et, le moins anticipé, **l'écart de flux de travail**.

Cet écart est documenté [22] : l'assistance accélère massivement le travail initial, typiquement le prototypage, mais la partie restante (précision, cas limites, intégration aux systèmes internes) peut demander davantage d'effort que si le projet avait été construit manuellement dès le départ. La raison est structurelle : l'outillage a d'abord ciblé la boucle

interne du développement, pas la boucle externe. Un projet atteint 80 % de son résultat visible en trois semaines et consomme six mois sur les 20 % restants. C'est la dérive de planning la plus courante, et elle est systématiquement lue comme un échec d'estimation alors qu'elle est un effet d'outil.

L'accumulation est mesurée, pas seulement déduite. GitClear [16] documente sur 623 millions de changements réels ce que ce mécanisme décrit : copié-collé en hausse, refactoring en chute, duplication de blocs en croissance. C'est la seule mesure directe du régime d'accumulation de ce dossier, et elle est indépendante du socle d'enquête sur lequel s'appuie le reste du chapitre.

Horizon de visibilité. 6 à 18 mois pour la dette d'accumulation. 2 à 4 mois pour l'écart de flux, qui se révèle au premier passage en production.

Coût réel. Coût de revue, test, documentation, maintenance de chaque ligne conservée sans avoir été demandée. Coût de replanification quand l'écart prototype/production se manifeste après engagement du calendrier, et c'est le poste le plus coûteux commercialement, parce qu'il se paie en crédibilité.

Signaux et seuil d'alerte.

- Ratio lignes produites / lignes livrées en production, par sprint. Alerte au-delà de 1,5.
- Nombre de branches sans activité depuis plus de 30 jours, en tendance.
- Écart entre estimation initiale et réel sur la phase d'industrialisation, distingué du prototypage. Alerte si l'écart dépasse 40 % deux sprints de suite.
- Taille moyenne des demandes de fusion, en tendance.

Contre-mesure. Agir sur le **rythme d'entrée** du travail partiellement terminé. Supprimer les branches accumulées ne traite que le symptôme : elles se reconstituent au débit qui les a produites. Réponse Lean orthodoxe, par flux tiré et limitation du travail en cours : plafond de demandes de fusion ouvertes par développeur, et discipline de **petits lots**, explicitement identifiée comme contre-mesure aux risques du développement assisté [17]. Sur l'écart de flux : ne pas réduire les estimations tant qu'il n'est pas fermé, et estimer en deux phases distinctes, prototype et industrialisation, avec des taux de productivité différents assumés.

Effet pervers. Le plafond de demandes ouvertes crée une incitation à fusionner pour libérer le quota, donc à relire moins bien. Il doit être couplé au mécanisme 2, jamais installé seul. L'estimation en deux phases est difficile à défendre commercialement : elle affiche un coût d'industrialisation que le concurrent, lui, n'affichera pas.

Porteur. *Installe* : direction des systèmes d'information pour le plafond de demandes de fusion et l'outillage de flux. *Applique* : direction de projet, semaine après semaine. *Arbitre* : comité de direction DSI pour la politique d'estimation en deux phases. Carte complète au §5.3.

. . .

C.2 – Mécanisme 2 – Défauts sémantiques et déplacement du goulot

Mécanisme. Les deux phénomènes n'en font qu'un et se renforcent circulairement. Le code généré compile presque toujours mais peut manquer une sémantique : un cas limite, une exception métier, une condition d'arrêt. Le défaut n'est plus une erreur visible arrêtée par le compilateur, c'est une erreur de sens qui ressemble à du bon code. Dans le même temps, le volume à relire augmente pendant que la capacité de relecture reste constante. Un code plausible en volume croissant sature la revue ; une revue saturée laisse passer davantage de code plausible.

Le premier versant porte un nom, la **taxe de vérification** [22] : le temps économisé à l'écriture est fréquemment réalloué à l'audit. 30 pour cent des développeurs déclarent peu ou pas de confiance dans le code généré : les outils ne signalent pas leur incertitude et produisent des sorties erronées avec un aplomb constant, ce qui contraint l'ingénieur à traiter chaque interaction comme potentiellement trompeuse. Or vérifier est cognitivement différent de créer, et généralement plus coûteux.

Le second versant est un **déplacement de charge entre personnes**. Un auteur produit en quelques minutes une modification qui occupera un relecteur pendant des heures. Cette asymétrie n'apparaît dans aucune métrique individuelle et reste socialement invisible : elle se manifeste d'abord comme une lenteur imputée au relecteur.

Horizon de visibilité. 3 à 9 mois.

Coût réel. Instabilité de livraison, le poste mesuré le plus directement [17] : davantage d'échecs de changement, de reprises, de délais de résolution. L'instabilité continue de dégrader la performance produit et d'alimenter l'épuisement, sans être compensée par la vitesse. À quoi s'ajoute un coût social peu discuté : le relecteur devient le point de friction perçu de l'équipe.

Signaux et seuil d'alerte.

- **Taux de reprise**, l'indicateur de tête, ajouté en 2025 aux quatre métriques de livraison classiques [17].
- Délai médian en revue, en tendance mensuelle. Un doublement sur un trimestre est une alerte.
- Part des demandes fusionnées sans revue effective. Toute valeur non nulle en croissance est une alerte.
- Taille médiane des demandes de fusion.

- Incidents de production rapportés au nombre de changements livrés, et non en valeur absolue, qui masque l'effet volume.

La télémétrie *Acceleration Whiplash* de Faros [36] (2026, environ 22 000 développeurs et plus de 4 000 équipes ; **niveau B, éditeur intéressé**) documente une aggravation nette de ces indicateurs entre les périodes de plus faible et de plus forte adoption d'une même organisation : le ratio incidents rapportés aux demandes de fusion progresse de 242,7 %, le délai médian en revue de 441,5 %, la part des demandes fusionnées sans revue effective de 31,3 %. Ces amplitudes se lisent comme des ordres de grandeur, non comme des cibles : elles comparent deux périodes au sein d'une même organisation, sur une population d'éditeur, et l'entrée 36 signale que ce même rapport conteste par ailleurs la thèse de l'amplificateur retenue au §3.1.

Contre-mesure. Trois leviers, dans cet ordre.

Déplacer la vérification vers l'auteur. Le retour automatisé doit être délivré au moment de l'écriture, non au relecteur en aval [17] : il est plus efficace de faire corriger l'auteur que de faire détecter le relecteur. Cela suppose des agents de revue contextualisés, capables d'appliquer les standards de l'organisation avant intervention humaine.

Petits lots. Contrainte de taille maximale des modifications, opposable.

Réinterroger la revue asynchrone. Levier le plus inconfortable [17] : dès lors que le temps de génération s'effondre, la revue asynchrone n'est plus évidemment optimale, et automatiser les tests peut offrir un meilleur rendement qu'optimiser la relecture manuelle. La revue humaine est une porte de qualité parmi d'autres.

Effet pervers. L'automatisation de la revue déplace la confiance sans la fonder : une équipe qui voit passer un agent de revue relit moins attentivement, y compris quand l'agent n'a pas la contextualisation nécessaire. La contrainte de taille est contournable : on découpe une modification en cinq demandes mécaniquement liées, dont aucune n'est intelligible isolément. La contrainte doit porter sur l'unité de sens, pas sur le nombre de lignes.

Porteur. *Installe* : direction des systèmes d'information pour le retour automatisé porté vers l'auteur et la contrainte de taille. *Applique* : direction de projet. *Arbitre* : comité de direction DSI pour le régime de revue lui-même. Carte complète au §5.3.

. . .

C.3 – Mécanisme 3 – Perte de maîtrise et asymétrie de responsabilité

Mécanisme. Le bateau de Thésée est un vaisseau dont chaque planche est remplacée au fil des voyages ; au terme, aucune pièce d'origine ne subsiste. Le paradoxe devient concret quand une part croissante d'un système est produite par génération assistée, sans que l'équipe qui en porte la responsabilité contractuelle puisse encore expliquer pourquoi il fonctionne.

Le point à tenir n'est pas la part de code générée, qui mesure la production et non la compréhension. Le point est **l'asymétrie** : le contrat continue d'engager l'équipe humaine dans les mêmes termes, tandis que la capacité effective à répondre (corriger un défaut, expliquer une décision de conception à un auditeur, adapter le système à une contrainte réglementaire nouvelle) a diminué sans que rien ne le signale. Risque simultanément opérationnel, juridique et professionnel.

Le mécanisme sous-jacent est documenté [22] : le risque est aigu quand l'outil valide l'hypothèse initiale de l'utilisateur indépendamment de sa pertinence architecturale, et que le développeur n'a pas l'expertise pour en vérifier la justesse. Le système est alors accepté sur un critère de plausibilité, pas de compréhension.

Horizon de visibilité. 12 à 36 mois. Se révèle à l'occasion d'un incident majeur, d'un audit, ou du départ d'un développeur clé.

Coût réel. Allongement du délai de résolution sur les zones non maîtrisées. Impossibilité de répondre à un auditeur dans les délais. Coût de reconstitution de la compréhension, qui est en pratique un coût de réécriture. Exposition contractuelle non provisionnée.

Signaux et seuil d'alerte. Ce mécanisme relève du régime de seuil : aucun indicateur ne monte régulièrement à mesure que la maîtrise se perd. Les signaux observés sont faibles et ne valent que par leur convergence ; la détection repose principalement sur des tests provoqués.

Signaux observés

- Délai de résolution des incidents, ventilé entre zones à maîtrise revendiquée et zones assumées non maîtrisées. Écart supérieur à un facteur 3 = alerte.
- Ratio temps de diagnostic / temps de correction, par incident. Un diagnostic durablement plus long que la correction indique que le défaut est trouvé par tâtonnement, non par compréhension.
- Nombre de correctifs successifs avant résolution effective d'un même incident, en tendance.
- Part des correctifs produits par re-génération complète plutôt que par modification ciblée.

- Part des sollicitations de l'assistant portant sur le code de l'équipe elle-même, et non sur un langage ou une bibliothèque. Une équipe qui interroge un outil pour comprendre son propre code a franchi un seuil. Mesurable sur les journaux, sous réserve du périmètre décrit plus loin à propos de l'usage non mesuré.
- Nombre de personnes capables d'intervenir sur chaque module critique, en tendance. Une décroissance non expliquée par les départs est le signal le plus fiable de la liste.

Tests provoqués

- **Explicitation.** Un développeur tiré au sort dans l'équipe explique à un tiers, sans support et sans assistance, un module tiré au sort parmi ceux de la part maîtrisée revendiquée. Durée : 20 minutes. Le tirage au sort est constitutif du test : sans lui, l'exercice mesure le meilleur cas, qui n'est pas celui qui exposera l'équipe. Trimestriel. Échec = alerte.
- **Refonte à blanc.** Sur une journée, l'équipe reprend un composant restreint sans assistance. On mesure l'écart à l'estimation, pas le résultat produit.
- **Simulation d'audit.** Un tiers demande la justification de trois décisions de conception structurantes prises dans les six derniers mois. Absence de réponse documentée sur l'une des trois = alerte.

Contre-mesure. Définir et défendre une **part maîtrisée en propre** : modules dont l'équipe peut, à tout moment, expliquer le fonctionnement, effectuer une refonte sans assistance et assurer la transmission à un nouvel arrivant. Elle inclut nécessairement les modules critiques (cryptographie, authentification, gestion des accès, traitement de données sensibles) et se vérifie par l'exercice d'explicitation, pas par déclaration.

Deux dispositifs la rendent vérifiable, et sans eux les tests ci-dessus ne sont pas exécutables : une **cartographie explicite** des modules à maîtrise revendiquée, tenue à jour et opposable ; une **consignation des décisions d'architecture** au fil de l'eau, seul artefact opposable à un auditeur réel.

Effet pervers. Une part maîtrisée définie trop largement n'est jamais tenue et devient une fiction documentaire, plus dangereuse que son absence puisqu'elle rassure. Mieux vaut 15 % de périmètre réellement maîtrisé que 60 % déclarés. L'exercice d'explicitation, mal introduit, produit du contournement et de la défiance plutôt que de la maîtrise.

Porteur. *Installe* : direction des systèmes d'information pour la cartographie et la consignation des décisions d'architecture. *Applique* : direction de projet, et collaborateur pour la traçabilité substantive. *Arbitre* : comité de direction DSI pour le périmètre de la part maîtrisée. Carte complète au §5.3.

. . .

C.4 – Mécanisme 4 – Érosion des compétences

Mécanisme. Deux phénomènes distincts, souvent confondus.

Le premier est le **déconditionnement des gestes**, ou syndrome du bazooka : l'atelier mobilise un modèle pour des tâches qui ne le méritent pas : changer une règle de style, corriger une faute de frappe. Il s'installe par continuité d'usage, sans décision, et le coût unitaire négligeable devient un coût cumulé qui ne l'est pas. S'y ajoute le *tool sprawl* [22] : décider quel outil mobiliser pour quelle tâche est un travail à refaire à chaque fois, jamais capitalisé, qui détruit l'état de concentration que ces outils devaient préserver.

Le second est le **paradoxe de l'expertise**, plus grave [22] : abaisser la barrière d'entrée offre un filet de sécurité réel, mais court-circuite la lutte productive nécessaire à la constitution d'une expertise profonde. Ces usages délivrent des gains de productivité tout en **bloquant le développement des compétences**, puisqu'ils suppriment la résolution de problèmes par la pratique, qui est le mécanisme même du compagnonnage.

C'est le point à retenir : l'atrophie n'est pas d'abord une perte chez les seniors, c'est une **non-constitution chez les juniors**. À 5 ou 10 ans, la compétence non assistée moyenne du projet baisse par renouvellement, sans qu'aucun individu n'ait rien désappris.

Horizon de visibilité. 24 mois et plus. Le mécanisme le plus lent et le plus difficile à inverser.

Coût réel. Perte d'accès à un pan du métier en cas d'indisponibilité de l'assistance : coupure de service, changement de fournisseur, décision réglementaire, exigence d'un client refusant l'IA sur ses données. Perte de la capacité à arbitrer une proposition de l'outil, qui est la compétence dont dépendent tous les autres mécanismes de ce chapitre. Dégradation de l'attractivité pour les profils expérimentés.

Signaux et seuil d'alerte.

- Écart de performance entre seniors et juniors sur des tâches non assistées comparables, l'indicateur le plus direct.
- Performance de l'équipe lors des exercices périodiques sans assistance.
- Enquête anonyme sur la confiance à traiter une tâche courante sans assistance, en tendance annuelle.
- Volume de sollicitations par développeur en croissance sans corrélation avec le volume livré.
- Nombre d'outils IA distincts en usage sur un même projet. Au-delà de 3, alerte de *tool sprawl*.

Contre-mesure. Elle relève du compagnonnage plutôt que de l'abstinence [22] : apparier les juniors à des mentors seniors pour la revue des décisions d'architecture assistées, et réserver l'écriture manuelle aux composants systèmes complexes. Traduit en discipline de projet : une part définie du temps sur des tâches sans assistance, choisies pour leur valeur formatrice ; une politique de recrutement qui évalue la compétence non assistée ; une réduction délibérée du nombre d'outils.

Effet pervers. La « journée sans IA » se transforme en rituel symbolique dès lors qu'elle n'a pas d'enjeu : elle n'a de valeur que si son résultat est examiné. Symétriquement, une politique d'usage gradué trop stricte produit du contournement par l'usage de comptes personnels.

Porteur. *Installe* : ressources humaines pour la politique d'exercice, l'évaluation annuelle et le recrutement. *Applique* : direction de projet pour l'appariement et le temps protégé. *Arbitre* : comité de direction DSI. Carte complète au §5.3.

. . .

C.5 – Mécanisme 5 – Intégrité de la chaîne logicielle

Mécanisme. Seul mécanisme du chapitre capable de produire un incident majeur en quelques heures, et seul dont la probabilité augmente directement avec le degré d'autonomie accordé aux agents. Quatre vecteurs.

Injection de consigne par contenu tiers. Un agent qui lit un ticket, une documentation, un fichier de dépendance ou une page web traite ce contenu comme de la donnée, mais un contenu rédigé pour ressembler à une instruction peut être suivi. La surface d'attaque n'est plus la sollicitation de l'utilisateur, c'est tout ce que l'agent est autorisé à lire.

Agents porteurs de droits. Un agent disposant d'un jeton d'accès au dépôt, à la chaîne d'intégration ou à un environnement dispose de la capacité d'action associée. La question n'est pas « le modèle est-il fiable » mais « quel est le périmètre de dégâts atteignable avec ces droits ».

Fuite de données par les sollicitations. Secrets, identifiants, extraits de code client, données personnelles transmis dans le contexte d'une requête.

Dépendances hallucinées. Un modèle propose un paquet inexistant ; le nom est plausible et récurrent ; un tiers l'enregistre et y place une charge. La suggestion devient un vecteur d'exécution de code.

S'y ajoute la **licence du code produit**, qui n'est pas un risque de sécurité mais partage son horizon : elle se découvre à l'audit, pas à l'écriture.

Horizon de visibilité. Jours. Parfois heures.

Coût réel. Incident de sécurité, avec les conséquences contractuelles, réglementaires et réputationnelles associées. Le seul mécanisme du chapitre dont le coût peut excéder la valeur totale du projet.

Signaux et seuil d'alerte. Même régime que le mécanisme 3, pour une raison différente : ici, l'absence d'incident ne prouve rien. Un dispositif défaillant et un dispositif jamais attaqué produisent la même mesure. La détection repose donc sur des tests provoqués, et les signaux observés servent surtout à repérer l'érosion du contrôle lui-même.

Signaux observés

- Nombre d'actions d'agent à effet irréversible exécutées sans validation humaine, par semaine. Toute valeur non nulle non expliquée = alerte.
- **Taux d'approbation des demandes de validation.** Un taux proche de 100 % ne signale pas un dispositif sain : il signale que le contrôle est devenu formel. C'est la mesure directe de la fatigue d'approbation décrite plus bas.
- Délai médian entre présentation d'une demande de validation et approbation. En deçà de quelques secondes, l'opérateur n'a pas lu ce qu'il approuve.
- Nombre de jetons d'agent disposant d'un droit d'écriture en production, en tendance. La dérive se fait par ajout, jamais par retrait.
- Âge des paquets nouvellement introduits sur suggestion d'un outil, à la date d'introduction. Un paquet publié depuis moins de quelques semaines et faiblement adopté est le profil type de la dépendance hallucinée puis enregistrée par un tiers.
- Secrets détectés en amont de l'envoi, en volume et en tendance, à condition que la détection soit installée, faute de quoi ce signal vaut zéro par construction et non par sécurité.
- Part des sources lues par les agents provenant de l'extérieur du périmètre de confiance.

Tests provoqués

- **Injection contrôlée.** Placer une consigne inoffensive et traçante dans un contenu que les agents sont autorisés à lire (ticket, documentation interne, fichier de dépendance de test), puis vérifier si elle est suivie. C'est l'équivalent de l'exercice d'hameçonnage simulé, et c'est le seul test qui mesure réellement la séparation entre donnée et consigne. À conduire hors production et sous protocole validé par la filière sécurité.
- **Périmètre de dégâts.** À partir des droits effectivement détenus par un agent, établir jusqu'où une action non prévue pourrait aller. La question à documenter n'est pas « cela peut-il arriver » mais « si cela arrive, où cela s'arrête-t-il ».
- **Faux secret.** Introduire un identifiant factice dans une sollicitation et vérifier qu'il est bloqué avant l'envoi.

- **Reconstitution.** Rejouer a posteriori l'ensemble de ce qu'un agent a lu et fait sur 24 heures. L'impossibilité de le reconstituer est en soi le résultat du test.

Contre-mesure. Moindre privilège appliqué aux agents, avec validation humaine obligatoire sur toute action irréversible : écriture en production, suppression, publication, appel sortant vers un tiers. Séparation stricte entre contenu de confiance et contenu observé : ce qu'un agent lit dans un ticket, un dépôt tiers ou une page web est une donnée, jamais une consigne. Détection de secrets en amont de l'envoi. Contrôle des dépendances nouvellement introduites par génération, avant intégration.

Trois dispositifs conditionnent l'ensemble et doivent exister avant tout test : un **inventaire des droits** effectivement détenus par chaque agent en production ; une **journalisation** des sources lues et des actions engagées, suffisante pour permettre la reconstitution ; un **contrôle de résolution des dépendances**. Leur absence n'est pas un signal de risque élevé : elle est l'impossibilité de savoir quel est le risque, ce qui appelle une décision immédiate plutôt qu'une surveillance.

Effet pervers. La validation humaine systématique produit de la fatigue d'approbation : au bout de quelques centaines de validations, l'opérateur approuve par réflexe et le contrôle devient formel. Le dispositif ne tient que si le volume d'approbations reste compatible avec une attention réelle, ce qui impose de restreindre le périmètre d'action des agents plutôt que de multiplier les points de contrôle.

Porteur. *Installe* : direction des systèmes d'information pour l'inventaire des droits, la journalisation et le bornage agentique. *Applique* : direction des systèmes d'information, et prestataire externe sur les périmètres livrés. *Arbitre* : comité de direction DSI pour le degré d'autonomie accordé aux agents. Carte complète au §5.3.

. . .

C.6 — Mécanisme 6 — Coût et dépendance économique

Mécanisme. Deux dynamiques liées.

L'imprévisibilité du coût. Le coût unitaire est facturé dans une unité, le jeton, qui ne correspond à aucune ligne budgétaire préexistante et qui est opaque à la plupart des donneurs d'ordre. Les prix évoluent, les modèles changent, les usages s'étendent. À sollicitation identique, un même modèle peut produire des sorties de volumes sensiblement différents : sur 500 tâches rejouées quatre fois avec huit modèles de frontière, l'exécution la plus coûteuse d'une même tâche consomme le double de la moins coûteuse [24] ; nos tests internes relèvent des écarts entre exécutions répétées [21] (**niveau A pour la mesure publiée, niveau C pour nos relevés**). Le coût se loge dans les jetons d'entrée, c'est-à-dire dans le contexte réinjecté à chaque tour plutôt que dans le code produit ; ni la difficulté

estimée par des experts, ni la prédiction du modèle lui-même ne permettent d'anticiper la facture [24]. Le passage aux usages agentiques aggrave la question d'un ordre de grandeur, puisqu'une tâche déclenche une chaîne d'appels dont la longueur n'est pas déterminée à l'avance.

La captivité progressive. Sollicitations ajustées à un modèle, intégrations conçues sur une interface, développeurs formés à des spécificités, données hébergées chez un fournisseur ou son partenaire. Aucune décision n'est déraisonnable isolément ; cumulées sur 12 à 18 mois, elles produisent un système dont la portabilité n'a jamais été explicitement défendue.

Horizon de visibilité. 3 à 6 mois pour la dérive de coût. 12 à 18 mois pour la captivité.

Coût réel. Dépassement budgétaire non anticipé. Perte de pouvoir de négociation à l'échéance contractuelle. Coût de bascule, jamais provisionné parce que jamais estimé.

Signaux et seuil d'alerte.

- Écart entre consommation prévue et réelle par sprint. Alerte au-delà de 30 %.
- Coût par ticket fermé, en tendance sur six mois : la seule mesure qui rapporte la dépense à une valeur.
- Variance de consommation entre développeurs sur tâches équivalentes.
- Nombre d'appels à des fonctionnalités propriétaires sans équivalent chez un fournisseur alternatif.
- Écart entre la date théorique du prochain test de bascule et sa date réelle d'exécution, en tendance. Le report est le mode de défaillance normal de ce dispositif : il ne se supprime jamais, il se repousse. Au-delà de 6 mois de retard cumulé, considérer que le plan de sortie n'existe plus.
- Coût de bascule estimé, en tendance. Une estimation qui n'a pas été révisée depuis 12 mois n'est plus une estimation.

Contre-mesure. Suivi hebdomadaire de consommation avec analyse des écarts, alerte à 80 % d'un plafond défini par équipe. Stratégie de mixte de modèles, économiques sur les tâches simples, avancés sur les tâches complexes ; le gain doit être **mesuré sur la charge réelle de l'organisation** plutôt que repris d'un ordre de grandeur du marché. Sur la captivité, la dimension Portabilité du référentiel (chapitre 4) porte la contre-mesure structurelle : plan de sortie documenté par fournisseur, couche d'abstraction des appels, et **test de bascule annuel effectivement exécuté** : non planifié, exécuté. Ce test est le dispositif dont dépend la mesure : sans lui, le coût de bascule reste une hypothèse et les deux signaux ci-dessus n'ont pas de valeur d'entrée. Il suppose une date engagée au calendrier, un responsable nommé, et un périmètre restreint mais réel : basculer un usage secondaire de bout en bout vaut mieux que documenter la bascule de tous.

Effet pervers. Deux, importants.

Le plafond de jetons par équipe est la contre-mesure la plus risquée : il pousse mécaniquement l'usage vers les comptes personnels, et l'organisation perd simultanément la donnée et la maîtrise.

La couche d'abstraction anti-captivité remplace une dépendance fournisseur par une dépendance de framework, et son abstraction fuit précisément là où elle compterait : usage d'outils, mise en cache, sorties structurées, gestion des sessions longues. Une couche maison, mince et sous contrôle, est souvent préférable à un framework générique. Ce qui protège réellement de la captivité n'est pas la couche mais le test de bascule exécuté.

Encart — Le fine-tuning propriétaire

Le fine-tuning est traité ici comme un cas particulier de ce mécanisme, et non comme un risque autonome.

L'argument distinctif habituellement avancé est l'irréversibilité : un investissement d'entraînement consenti chez un fournisseur ne se transfère pas. L'argument ne résiste pas à sa propre contre-mesure. Si l'on versionne le corpus d'entraînement, ce que tout le monde recommande, ce qui n'est pas transférable se réduit à l'exécution de l'entraînement, c'est-à-dire au poste le moins coûteux. Dans un budget de spécialisation, la préparation des données, les jeux d'évaluation et l'intégration représentent l'essentiel de la dépense, et ces trois postes sont portables.

Le verrou réel est ailleurs : dans les jeux d'évaluation qui définissent ce que « fonctionner » veut dire pour ce cas d'usage, dans l'échafaudage de sollicitations construit autour du modèle, et dans l'intégration au système d'information.

D'où trois recommandations. **Essayer d'abord une approche par récupération augmentée**, qui produit dans un nombre significatif de cas un résultat équivalent sur un modèle générique, sans engagement d'entraînement. **Versionner le corpus, les jeux d'évaluation et l'échafaudage** dans un dépôt maîtrisé : sans jeu d'évaluation reproductible, on ne sait même pas mesurer la dégradation d'un modèle de repli. **Ne pas spécialiser sur des données à valeur juridique critique** (contrats, exigences réglementaires, documents de tiers sous confidentialité forte) sans validation juridique explicite et clause fournisseur adaptée.

Sur le plan réglementaire, les obligations européennes applicables aux modèles à usage général [30] sont entrées en application le 2 août 2025, et la Commission a publié des orientations traitant du cas où une entité modifiant un modèle en aval devient elle-même fournisseur au sens du règlement. Le régime est **applicable mais encore instable dans son interprétation**.

Porteur. *Installe* : direction des systèmes d'information pour les enveloppes, le mixte de modèles et le test de bascule. *Applique* : direction de projet. *Arbitre* : comité de direction DSI pour la stratégie fournisseur ; achats pour la clause de réversibilité. Carte complète au §5.3.

Annexe D — Méthode, sources et intérêts

Comment ce dossier a été construit, et à qui il profite

Cette annexe porte ce qu'un lecteur exigeant est en droit de demander à un document chiffré : selon quelle règle les chiffres ont été retenus, comment les sources d'origine commerciale ont été traitées, ce que le document ne couvre pas, et quel intérêt sert celui qui l'a écrit.

. . .

D.1 La règle de sobriété évidentielle

Chaque chiffre cité est daté à l'année, attribué à sa source primaire, contextualisé (population, tâche, conditions) et renvoyé à une entrée numérotée de l'annexe B, où figurent l'URL de la source et son niveau de preuve.

Lorsqu'une source récente vient nuancer une source antérieure, les deux sont citées. C'est le cas de l'étude METR de juillet 2025 et de sa note d'actualisation de février 2026, qui apparaissent systématiquement ensemble.

Lorsqu'un chiffre relève d'une observation d'atelier REDSEN et non d'une source publiée, cela est indiqué explicitement. Cette édition ne mobilise que des sources publiées, un retour d'expérience externe nommé, et des observations d'atelier agrégées et anonymisées.

Les entrées de l'annexe B sont numérotées par ordre de première apparition dans le texte. Une entrée jamais appelée, ou un appel sans entrée, sont des défauts que nous traitons comme tels.

D.2 Les trois niveaux de preuve

Chaque affirmation chiffrée porte son niveau, et chaque entrée de l'annexe B le rappelle.

Niveau	Nature	Usage admis
A	Étude publiée, protocole décrit, réplicable	Peut fonder une décision d'investissement
B	Enquête sectorielle large ou télémétrie d'éditeur, corrélationnelle	Peut fonder une hypothèse de travail et une priorisation
C	Observation interne REDSEN ou ordre de grandeur relayé	Illustre, ne démontre pas. Ne fonde jamais seul une décision d'investissement, et porte toujours la mention de son échantillon

La distinction entre A et B n'est pas une distinction de qualité mais de nature. Une enquête déclarative auprès de cinq mille professionnels est une source précieuse et ne devient pas une mesure pour autant : elle établit des corrélations et des perceptions, pas des causalités.

D.3 Les sources d'origine fournisseur

Ce dossier s'ouvre sur une critique des études produites par ceux qui vendent les outils, et s'appuie ensuite sur plusieurs sources qui en émanent : le programme DORA est opéré par Google Cloud, l'étude Cui et al. est cosignée par des chercheurs Microsoft et conduite pour partie chez Microsoft, la télémétrie Faros émane d'un éditeur d'outils de mesure.

La contradiction n'est qu'apparente, et le critère que nous appliquons mérite d'être écrit.

> **Une source d'origine fournisseur est recevable lorsque son protocole est publié, ses limites documentées, et ses conclusions défavorables à son émetteur.**

Une étude qui remplit ces trois conditions vaut mieux qu'une étude prétendument indépendante dont le protocole reste opaque. Une étude qui n'en remplit aucune est un support de vente, quel qu'en soit l'auteur.

Deux conséquences pratiques. Les sources d'éditeur retenues portent la mention *éditeur intéressé* à leur entrée bibliographique, et ne sont jamais citées comme référence d'amplitude lorsqu'elles n'ont pas de protocole public. Et lorsque deux sources recevables se contredisent, comme le programme DORA et la télémétrie Faros sur la question de savoir si la maturité d'ingénierie protège de la dégradation aval, nous le signalons plutôt que de choisir silencieusement.

D.4 Une convention de dénomination

« DORA » désigne dans ce dossier **le programme de recherche opéré par Google Cloud**. Le règlement (UE) 2022/2554 sur la résilience opérationnelle numérique du secteur financier, également appelé DORA, est systématiquement cité ici sous son numéro. Les deux objets n'ont aucun rapport, et le lectorat visé connaît généralement le second avant le premier.

D.5 Périmètre et limites

Ce que ce dossier traite. La production de logiciel assistée par IA générative : ce qui s'y gagne, ce qui s'y dégrade, sous quelle responsabilité, avec quelles dépendances. La restriction au développement logiciel est délibérée et permet d'être précis là où le débat général est bruyant.

Ce qu'il ne traite pas, et qui compte. Le référentiel du chapitre 4 couvre la gouvernance de l'assistance, non la qualité d'ingénierie : les mécanismes 1 et 2 du chapitre 3, qui sont les deux plus probables, relèvent de référentiels existants et le §4.1 l'explicite. Le dossier porte par ailleurs sur un seul maillon de la chaîne qui va du besoin à la solution utilisée : la clôture y consacre le §6.3, et la conclusion qu'elle en tire est inconfortable : accélérer une station qui n'est pas la contrainte du système n'améliore pas le débit du système.

Ce qui datera le plus vite. Les références produits et fournisseurs, les amplitudes chiffrées, et l'état d'application des textes européens. La cartographie CMMI du chapitre 2 est explicitement présentée comme une photographie à réévaluer annuellement.

D.6 Déclaration d'intérêts

REDSEN est une entreprise de services numériques qui commercialise des prestations de développement logiciel, y compris assisté par IA, ainsi que des missions d'évaluation fondées sur le référentiel présenté au chapitre 4. **Ce document sert donc un intérêt commercial**, et il serait malvenu de l'écrire ailleurs qu'en toutes lettres.

Nous demandons au lecteur de lui appliquer le critère que nous appliquons nous-mêmes aux sources d'origine fournisseur.

Le protocole est-il publié ? Les sources sont numérotées, datées et accessibles ; les niveaux de preuve sont explicites ; les observations internes sont signalées comme telles et ne fondent seules aucune recommandation.

Les limites sont-elles documentées ? C'est l'objet du §D.5, et de plusieurs passages du corps du texte : la transposition assumée d'une étude hors développement au §1.2, la divergence non tranchée sur une série GitClear, la requalification d'une source commerciale au §5.2.

Les conclusions sont-elles défavorables à l'émetteur ? Sur ce point, nous soutenons que oui. Ce dossier affirme que l'assistance ne produit de valeur que dans un atelier capable de l'absorber, qu'une organisation sans socle a intérêt à renforcer son socle avant d'outiller, et qu'une accélération obtenue hors de la contrainte du système ne produit pas de débit. Ces trois affirmations restreignent le marché plutôt qu'elles ne l'élargissent.

Financement. Aucun financement externe n'a été reçu pour la rédaction de ce document. Aucun éditeur d'outil, aucun fournisseur de modèle n'a été consulté sur son contenu ni n'en a relu de version préalable.

Rédaction. Emmanuel Bouchet, associé gérant, et Xavier Perrin, consultant senior.

Contributions externes. L'avant-propos est signé par son auteur et engage celui-ci.

Usage de l'intelligence artificielle. Cet ouvrage a été conçu et rédigé sous la responsabilité de ses auteurs. Des outils d'intelligence artificielle générative ont été mobilisés à certaines étapes du travail : Perplexity pour l'exploration documentaire et l'identification de sources ; Claude pour l'assistance à la structuration, à la reformulation et à la relecture de certains passages. Les choix éditoriaux, l'analyse, la sélection des sources, la vérification des informations et la version finale relèvent de la responsabilité exclusive des contributeurs.

D.7 Ce que nous corrigerons

Une correction signalée est une correction due. Les erreurs de fait, d'attribution ou de calcul portées à notre connaissance seront corrigées dans la version suivante, et les corrections substantielles seront listées à l'ouverture de celle-ci.

Aucun point ne reste en suspens à la date d'arrêté au sens d'une action éditoriale due. Une limite mérite en revanche d'être nommée plutôt que tue : le facteur d'efficacité énergétique cité à l'avant-propos provient d'un banc d'essai ouvert unique, dont les optimisations logicielles ont été conduites avec le constructeur, et il porte sur des serveurs à huit cartes plutôt que sur la baie effectivement annoncée. L'entrée 4 le dit.

Annexe E — Glossaire

Les termes propres à ce dossier, et ceux dont l'usage courant prête à confusion

Chaque entrée indique où le terme est défini ou employé pour la première fois. Les termes anglais sont donnés lorsqu'ils sont d'usage courant en atelier.

. . .

A

Amplificateur. Thèse centrale du dossier : l'assistance générative ne corrige rien, elle multiplie ce qu'elle trouve. Elle magnifie les forces d'une organisation dont le socle technique et culturel est solide, et magnifie les dysfonctionnements de celle qui n'en dispose pas. *§3.1, §6.4.*

Artefact opposable. Trace datée qui établit qu'une décision a été prise et appliquée : registre, compte rendu consigné, clause signée, tableau de bord. Une intention n'est pas un artefact, un document non daté non plus. C'est l'unité de preuve du référentiel et de la matrice des décisions. *§4.3, annexe A.*

Attendu. Élément observable dont la satisfaction établit qu'une organisation a atteint un palier sur une dimension. Un attendu se démontre par un artefact, jamais par une déclaration. *§4.3.*

B

Boucle interne / boucle externe. La boucle interne désigne le travail de l'ingénieur à son poste : écrire, exécuter, corriger. La boucle externe désigne ce qui suit : revue, intégration, qualification, mise en production. L'outillage d'assistance a d'abord ciblé la première, ce qui explique l'écart de flux de travail. *Annexe C.1.*

C

CMMI. *Capability Maturity Model Integration*. Modèle de maturité organisationnelle structuré en domaines de pratique (*practice areas*). Employé ici comme grammaire de lecture, non comme référentiel de certification. *§2.1.*

Courbe en J. Forme que prend le retour sur investissement d'une adoption assistée : creux initial attendu, puis remontée conditionnée à la redéfinition des processus et à la solidité des fondations d'ingénierie. La remontée n'est pas automatique. *§5.2.*

D

Demande de fusion. Unité de proposition de modification du code, soumise à revue avant intégration (*pull request*, *merge request*). Sa taille et son délai en revue sont deux signaux du mécanisme 2. *§3.4.*

Dépendance hallucinée. Paquet logiciel inexistant, proposé par un modèle sous un nom plausible. Le risque naît lorsqu'un tiers enregistre ce nom et y place une charge : la suggestion devient un vecteur d'exécution de code. *Annexe C.5.*

DORA (programme). Programme de recherche sur la performance des organisations de développement, opéré par Google Cloud, publiant une enquête annuelle. **À ne pas confondre avec le règlement (UE) 2022/2554**, également appelé DORA dans le secteur financier, et cité dans ce dossier sous son seul numéro. *Annexe D.4.*

E

Écart de flux de travail. Décalage entre l'accélération du travail initial, typiquement le prototypage, et l'effort restant sur la précision, les cas limites et l'intégration aux systèmes internes. Un projet atteint 80 % de son résultat visible très vite et consomme l'essentiel de son délai sur les 20 % restants. Première cause de dérive de planning en développement assisté. *§3.4, mécanisme 1.*

En-cours. Travail engagé et non livré. En développement, du code écrit qui vieillit, diverge de son environnement et dont le coût de reprise croît avec le temps d'attente. Accélérer une station placée avant la contrainte du système produit de l'en-cours, pas du débit. *§6.3.*

F

Fatigue d'approbation. Érosion du contrôle par répétition : au bout de quelques centaines de validations, l'opérateur approuve par réflexe. Effet pervers de la validation humaine systématique, mesuré par le taux d'approbation et le délai de décision. *§3.4, mécanisme 5.*

Fine-tuning. Spécialisation d'un modèle sur un corpus propre à l'organisation. Traité ici comme un cas particulier du risque de dépendance, non comme un risque autonome : le verrou réel n'est pas l'entraînement mais les jeux d'évaluation, l'échafaudage de sollicitations et l'intégration. *Annexe C.6.*

I

IA prédictive. Modèles de classification, de détection d'anomalies et de scoring. Reproductibles à contexte constant, ils relèvent d'un contrat de performance mesurable. *§1.3.*

IA générative. Production probabiliste de texte, de code, d'images ou de son. Deux appels identiques peuvent produire deux résultats différents. Relève d'un contrat de moyens et de traçabilité, jamais de résultat reproductible. *§1.3.*

IA agentique. Composition de chaînes d'appels génératifs avec des outils (accès web, exécution de code, appels d'interface) pour accomplir des tâches en plusieurs étapes. Hérite des fragilités de la générative, avec la démultiplication des risques par la composition. *§1.3.*

Injection de consigne. Contenu rédigé pour ressembler à une instruction, placé dans une source qu'un agent est autorisé à lire : ticket, documentation, page web. La surface d'attaque n'est plus la sollicitation de l'utilisateur mais tout ce que l'agent peut lire. *Annexe C.5.*

J

Jeton. Unité de facturation de l'IA générative (*token*), comptée en entrée et en sortie. Sa particularité est que le volume facturé n'est pas spécifié par l'acheteur : la longueur de la réponse est décidée par le modèle. *§1.2.*

Journalisation reconstituable. Enregistrement des sources lues et des actions engagées par un agent, suffisant pour rejouer a posteriori ce qui s'est produit. Sans elle, l'absence d'incident constaté ne prouve rien. *§3.4, mécanisme 5.*

M

Mécanisme (de risque). Enchaînement causal nommé, doté d'un horizon de visibilité, d'un signal de tête, d'un seuil d'alerte, d'une contre-mesure et de l'effet pervers de celle-ci. Le dossier en instruit six. *§3.4, annexe C.*

Moindre privilège. Principe selon lequel un agent ne détient que les droits strictement nécessaires à sa tâche. La question utile n'est pas « le modèle est-il fiable » mais « quel est le périmètre de dégâts atteignable avec ces droits ». *Annexe C.5.*

N

Niveau de preuve (A, B, C). Qualification de chaque affirmation chiffrée. **A** : étude publiée, protocole décrit, réplicable. **B** : enquête sectorielle large ou télémétrie d'éditeur, corrélationnelle. **C** : observation interne ou ordre de grandeur relayé, qui illustre sans démontrer. *Annexe D.2.*

P

Palier. Niveau de maturité sur une dimension du référentiel, de 1 à 5. Les paliers sont cumulatifs et aucun n'est meilleur en soi. *§4.2.*

Palier-cible. Palier qu'une organisation décide d'atteindre, par dimension et par nature de mission. Il s'homologue, il ne se constate pas. *§4.4.*

Paradoxe de l'expertise. Abaisser la barrière d'entrée offre un filet de sécurité réel, mais court-circuite la lutte productive nécessaire à la constitution d'une expertise profonde. L'usage délivre alors un gain de productivité tout en bloquant le développement des compétences. *§3.4, mécanisme 4.*

Part maîtrisée en propre. Périmètre de modules dont l'équipe peut, à tout moment, expliquer le fonctionnement, effectuer une refonte sans assistance et assurer la transmission. Elle se vérifie par un exercice d'explicitation, jamais par déclaration. *§3.4, mécanisme 3.*

Petits lots. Discipline consistant à limiter la taille des modifications soumises à revue. Identifiée comme contre-mesure aux risques du développement assisté, elle porte sur l'unité de sens et non sur le nombre de lignes. *§3.4, mécanismes 1 et 2.*

Plan de sortie. Document décrivant, par fournisseur, ce qu'il faudrait faire pour cesser de l'utiliser. Il n'a de valeur qu'accompagné d'un test de bascule exécuté. *§4.3, annexe C.6.*

Profil de maturité. Restitution d'une évaluation sous forme de cinq scores, un par dimension, sans note globale. L'absence de moyenne est un choix : elle masquerait ce que le profil doit faire apparaître. *§4.5.*

R

Rebut direct. Code produit puis annulé après revue, parce qu'incorrect, incohérent ou dangereux. Le poste le plus facile à ignorer, puisqu'il ne laisse aucune trace dans le produit livré. *§3.2.*

Régime d'accumulation. Dette qui croît de façon continue et se mesure par indicateurs : duplication, tests fragiles, documentation obsolète. Effet iceberg : visible tard, résorbable progressivement. *§3.3.*

Régime de seuil. Dette qui ne se manifeste pas jusqu'au point de bascule, où elle se manifeste d'un coup. Les métriques du régime d'accumulation ne la prédisent pas ; sa détection repose sur des tests provoqués. *§3.3.*

S

Signal de tête. L'unique indicateur à installer sur un mécanisme si l'on n'en installe qu'un, avec son seuil d'alerte. Les autres signaux figurent en annexe C. *§3.4.*

T

Taux de reprise. Part du travail consacrée à corriger ce qui vient d'être livré (*rework rate*). Ajouté en 2025 aux quatre métriques de livraison classiques, c'est l'indicateur de tête du mécanisme 2. *§3.4.*

Taxe de vérification. Réallocation du temps économisé à l'écriture vers l'audit de ce qui a été produit. Vérifier est cognitivement différent de créer, et généralement plus coûteux. *§3.4, mécanisme 2.*

Test de bascule. Exercice consistant à faire fonctionner effectivement un périmètre sur un second fournisseur. Un plan de sortie non éprouvé est une intention ; le test exécuté est la preuve. *§4.3, annexe C.6.*

Test provoqué. Exercice délibérément déclenché pour révéler un risque qu'aucun indicateur ne signale : explicitation à froid, refonte à blanc, simulation d'audit, injection contrôlée, périmètre de dégâts, faux secret, reconstitution. Le dossier en décrit sept. *Annexe C.3 et C.5.*

Théorie des contraintes. Règle selon laquelle le débit d'un système est fixé par sa contrainte, et selon laquelle une amélioration obtenue ailleurs qu'à la contrainte n'améliore pas le débit d'ensemble. Elle limite explicitement la portée de ce dossier. *§6.3.*

***Tool sprawl*.** Prolifération d'outils d'assistance, dont le coût n'est pas la licence mais la décision : choisir quel outil mobiliser pour quelle tâche est un travail à refaire à chaque fois, jamais capitalisé. *§3.4, mécanisme 4.*