Malorean Corp.

Un roman pour s'identifier

Un panorama pour comprendre

Développeurs, IA et petite entreprise : une histoire (presque) vraie de transformation

Ce que vivent vraiment
les équipes d'un SaaS ERP
face à l'IA aujourd'hui



LAURENT DESÉCHALLIERS

Malorean Corp

> *Hier, il fallait parfois des semaines pour transformer une idée en prototype. Aujourd'hui, une personne aidée par l'intelligence artificielle peut y parvenir en quelques jours. Ce changement est spectaculaire. Mais le véritable bouleversement pourrait bien se jouer ailleurs : dans la façon dont nous travaillons ensemble.*

Laurent Deséchalliers

Introduction

Ce qui me frappe le plus avec l'IA, ce n'est pas qu'elle sache écrire du code. C'est la vitesse à laquelle elle rend possible quelque chose qui, peu de temps auparavant, aurait demandé plusieurs jours ou plusieurs semaines de travail.

Une idée peut devenir un prototype avant même que la discussion sur sa faisabilité soit terminée. Un développeur peut explorer une piste sans attendre qu'une équipe entière soit disponible. Quelqu'un qui ne sait pas programmer peut parfois construire une première version de ce qu'il imagine.

Tout cela ne signifie pas qu'un prototype soit un produit fiable, ni que la complexité ait disparu. Mais le simple fait de pouvoir essayer change déjà notre rapport à ce qui semblait possible.

Or nos entreprises n'ont pas été construites pour un monde où produire une première solution peut devenir aussi rapide. Elles se sont organisées autour des compétences, du temps disponible, des passages de relais et des arbitrages nécessaires pour faire avancer le travail. Ces mécanismes ont leurs raisons d'être. Pourtant, quand les capacités des personnes évoluent, ils commencent à susciter des questions nouvelles. Faut-il encore travailler dans le même ordre ?

Attendre les mêmes validations ? Réserver les mêmes activités aux mêmes métiers ?

Ces questions ne concernent pas seulement les développeurs. Les Product Managers, les designers, les commerciaux, les responsables d'équipe et bien d'autres découvrent eux aussi de nouvelles façons de travailler. Certains peuvent désormais accomplir directement une partie de ce qu'ils demandaient auparavant à leurs collègues. D'autres voient une partie de leur activité changer de nature. Ce qui paraissait être une question d'outillage commence à toucher à la manière dont nous coopérons et prenons des décisions.

Il serait tentant d'annoncer une révolution dont on connaîtrait déjà l'issue. Je ne crois pas que nous en soyons là.

Des entreprises expérimentent de nouvelles répartitions du travail et des responsabilités. D'autres avancent plus prudemment, parfois avec de bonnes raisons. Certaines pratiques semblent prometteuses, mais nous manquons encore de recul pour savoir lesquelles résisteront aux contraintes du quotidien. Et une solution adaptée à une petite équipe ne devient pas automatiquement pertinente dans une entreprise plus grande, plus réglementée ou plus ancienne.

Je n'ai donc pas voulu écrire un manuel de la bonne organisation à adopter avec l'IA. Ce livre est une tentative de comprendre ce qui bouge réellement : ce que les outils rendent possible, comment les entreprises réagissent, ce

qu'elles gagnent, ce qu'elles risquent de perdre et les questions qu'elles devront trancher.

Pour cela, j'ai choisi deux chemins qui se croisent.

Le premier est celui de l'observation. Nous regarderons des entreprises réelles, des pratiques rendues publiques, des études et des expériences. J'essaierai de distinguer ce qui est établi de ce qui reste une hypothèse, et de ne pas confondre une annonce séduisante avec un résultat démontré.

Le second est celui d'une histoire. Nous suivrons Malorean Corp., une entreprise fictive dont les salariés vont découvrir progressivement ce que ces nouvelles possibilités changent à leur travail. Ils ne disposent pas d'un plan pour réinventer leur organisation. Ils ont des clients, des habitudes, des urgences, des convictions et parfois des désaccords. La fiction ne prouve rien ; elle permet de regarder ces transformations à hauteur de personnes.

Ces deux regards sont nécessaires. Les exemples réels aident à ne pas prendre nos intuitions pour des faits. L'histoire, elle, nous rappelle que les changements d'organisation ne se jouent pas seulement dans les organigrammes : ils touchent à la confiance, aux compétences, aux responsabilités et à la place de chacun.

Au fond, une question va traverser les pages qui suivent : **que se passe-t-il dans une entreprise lorsque les personnes qui la composent découvrent qu'elles peuvent faire bien**

davantage que ce que leur métier leur permettait jusqu'ici ?

À propos de l'auteur

Je travaille dans l'édition de logiciels depuis plusieurs décennies et j'exerce aujourd'hui comme Product Manager. Mon parcours m'a conduit du développement et de l'architecture à l'organisation des équipes, puis au produit, à ses opérations et à son marketing.

Cette expérience me donne une place privilégiée pour observer l'évolution de l'intelligence artificielle et ses effets sur notre travail. Je la découvre au quotidien dans mon activité professionnelle, mais aussi à travers les échanges avec mes collègues Product Managers. Nous partageons nos expériences, nos découvertes et nos interrogations. Ces conversations m'amènent souvent à regarder au-delà des outils : qu'est-ce qui change vraiment dans nos métiers, dans nos décisions et dans notre façon de travailler ensemble ?

Au fil de ces observations et de ces discussions, les questions se sont multipliées. J'ai eu envie de prendre du recul, de confronter nos impressions à ce qui se passe dans d'autres entreprises et de partager cette réflexion. C'est ainsi qu'est né ce livre.

L'écriture accompagne depuis longtemps mon parcours. J'ai contribué à la presse spécialisée et travaillé comme relecteur technique pour M2 Édition et Eyrolles. J'écris aujourd'hui sur

malorean.net, autour du Product Management, de l'IA, des méthodes de travail et des systèmes d'information.
Je contribue également à *Proposition Marketing*, un magazine indépendant consacré au marketing stratégique et opérationnel. Écrire pour des lecteurs venus du marketing et de la direction d'entreprise m'encourage à rendre les questions Produit accessibles, sans supposer un vocabulaire commun.

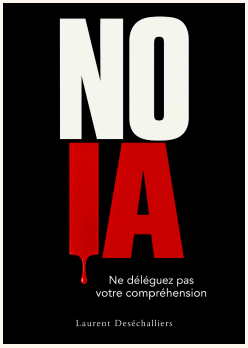
Laurent Deséchalliers

- laurent@malorean.net
- https://malorean.net

Du même auteur



Le Contexte — Comment construire un contexte qui permet à l'IA de comprendre ce qui compte, relier vos connaissances et capitaliser ce que vous apprenez au service de vos buts.
Sortie : octobre 2026.



No IA — *Ne déléguez pas votre compréhension.* Des méthodes sans IA pour entretenir notre mémoire, approfondir notre réflexion et exercer notre esprit critique. Des pratiques pour développer nos propres capacités et mieux tirer parti de l'IA sans lui déléguer notre compréhension.
Sortie prévue : fin 2026.

Plus d'informations sur https://malorean.net.

Remerciements

Je tiens d'abord à remercier les Product Managers avec qui j'échange régulièrement. Nous confrontons nos points de vue sur l'évolution du Product Management, des métiers du développement et, plus largement, sur les effets de l'intelligence artificielle sur le travail et l'organisation des entreprises. Ces conversations, parfois animées, sont une source constante de questions, de remises en cause et d'idées nouvelles. Certaines de ces réflexions se retrouvent dans les pages de ce livre. Merci pour votre curiosité, vos expériences partagées et votre envie de débattre.

Et puis, je veux remercier Christine, ma femme. À la fin de mon premier livre, *Le Contexte*, je lui avais promis une pause. J'ai finalement enchaîné presque aussitôt avec celui-ci. Heureusement, cette deuxième aventure a été moins gourmande de nos soirées et de nos week-ends que la précédente. Merci, Christine, pour ta patience et pour ton soutien, même lorsque mes réflexions finissent par devenir un nouveau livre.

Note sur les exemples

Les réflexions et les exemples de cet ouvrage se nourrissent de plus de trente ans de pratique dans les métiers de l'informatique. Ils viennent de situations rencontrées dans des projets et des environnements professionnels variés, et de nombreux échanges avec des product managers, CTO, engineering managers… d'autres entreprises : plusieurs organisations, plusieurs périodes de mon parcours.
Pour servir le propos du livre, j'ai modifié, généralisé ou recomposé ces situations. Cas d'usage, données chiffrées et métriques illustrent des problèmes récurrents et les enseignements que j'en ai tirés. Ce sont des constructions : aucune ne décrit un client, un produit ou une organisation réels, ni ne reflète les données, les pratiques ou les positions de mon employeur.
Les analyses, méthodes et opinions exprimées dans cet ouvrage sont personnelles et n'engagent que moi.

Ce livre a-t-il été écrit avec l'IA ?

Oui. Entièrement en collaboration avec l'intelligence artificielle.

Je sais que la question prend aujourd'hui une place particulière. On se demande si un article, un roman ou un livre a été écrit avec l'IA. Certains passent même les textes dans des détecteurs, comme s'il fallait absolument départager ce qui viendrait d'un humain et ce qui viendrait d'une machine.

Je préfère être transparent : l'IA a participé à toutes les étapes de cet ouvrage, de la recherche documentaire à la rédaction, en passant par l'organisation des idées, la vérification des références et la recherche d'angles morts.

Mais cette collaboration mérite une explication, car elle ne dit pas, à elle seule, qui a pensé le livre.

Pendant longtemps, pour mener un travail de cette ampleur, on pouvait s'entourer de personnes chargées de rechercher des documents, de confronter les idées, de relire, de structurer ou de mettre en forme. L'IA me donne aujourd'hui accès à une partie de ces capacités. Elle exécute vite, propose des formulations, signale des incohérences, pose parfois les questions auxquelles je n'avais pas pensé. Elle me permet d'explorer bien davantage que je ne pourrais le faire seul, avec le temps dont je dispose.

Pour écrire ce livre, j'ai utilisé un moteur de Contexte fondé sur les technologies de ChatGPT, avec Notion pour assurer la persistance des informations. Sources, idées, décisions et versions de travail peuvent ainsi être conservées et retrouvées d'une conversation à l'autre. J'ai déjà présenté cette approche dans mon premier livre, *Le Contexte*. J'y explique comment un tel système permet à l'IA de devenir une sorte de super secrétaire particulier : elle ne se contente plus de répondre à une question, elle garde le fil des projets et m'aide à les faire avancer dans la durée.

Il y a cependant une distinction essentielle : **la capacité d'exécuter n'est pas la responsabilité de comprendre et de choisir.** Les questions qui guident cet ouvrage, l'expérience professionnelle qui les nourrit, les interprétations, les arbitrages et ce que je décide finalement de défendre ou d'écarter sont les miens. L'IA peut me contredire, me challenger et m'aider à mieux formuler une pensée. Elle ne remplace ni mon jugement ni ma responsabilité d'auteur.

Je fais aussi une différence entre ce livre et des œuvres dont l'écriture est elle-même une part fondamentale de la création. Quand on lit Shakespeare dans une traduction de François-Victor Hugo, Cervantès, ou Cizia Zykë dans le texte, la voix, le rythme et le style comptent autant que ce qui est raconté. Pour une œuvre traduite, les choix du traducteur font eux aussi partie de l'expérience. La place que l'on souhaite

donner à l'IA dans ce type de littérature soulève d'autres questions, que je ne prétends pas trancher ici.
Ce livre comporte une part de fiction, mais son ambition première reste de comprendre une transformation technique et organisationnelle. La forme doit être agréable et claire ; elle est surtout au service des idées, des faits et des interrogations que je souhaite partager.
L'IA a rendu ce projet possible à une échelle que je n'aurais probablement pas pu atteindre avec mes seuls moyens, en dehors de mon activité professionnelle. Elle me permet de développer des projets personnels, comme ce livre, sans renoncer à l'exigence de réflexion qu'ils demandent.
Et il y a quelque chose d'assez cohérent, finalement, à avoir écrit un livre sur les transformations provoquées par l'IA en travaillant précisément de cette manière.
Laurent Deséchalliers

Sommaire

Introduction 3

À propos de l'auteur 7

Du même auteur 9

Remerciements 10

Note sur les exemples 11

Ce livre a-t-il été écrit avec l'IA ? 12

Malorean Corp. 17

1 Malorean Corp. — une journée pas comme les autres 18
2 L'IA a démultiplié la capacité de production des développeurs 26
3 Malorean Corp. — Le jour d'après 37
4 De nouveaux modèles d'organisation apparaissent 51
5 Malorean Corp. — Ce qu'on peut faire, ce qu'on doit faire 62
6 Les rôles se brouillent, les responsabilités restent 77
7 Malorean Corp. — Ce qu'Alice savait sans l'avoir écrit 89
8 Plus d'autonomie exige plus de cadre, pas moins 109
9 Malorean Corp. — Deux décisions raisonnables 127
10 L'IA va-t-elle faire disparaître les couches intermédiaires de l'entreprise ? 144
11 Malorean Corp. — Le temps retrouvé 164
12 Conclusion 184

Malorean Corp.

1 — Malorean Corp. — une journée pas comme les autres

Le siège de Malorean Corp. se trouvait au Havre. Mais depuis le Covid, cette adresse ne disait plus grand-chose de l'endroit où se faisait le travail. Les salariés étaient répartis dans toute la France. On se retrouvait devant une caméra, on discutait sur un écran partagé et, la plupart du temps, personne n'avait besoin d'être dans la même pièce pour faire avancer un dossier.

Malorean Corp. développait un ERP accessible en ligne : un de ces logiciels qui finissent par se glisser dans presque toutes les opérations d'une entreprise. Un client pouvait être une TPE, avec quelques salariés et des besoins très concrets. Un autre pouvait être un grand groupe, avec des processus complexes, des exigences particulières et peu de patience pour les approximations.

Les clients étaient surtout français. Il y avait bien quelques utilisateurs en Allemagne et en Espagne, mais principalement dans des filiales de groupes déjà clients en France. Hubert Félix Duca, le dirigeant, regardait plus loin. Il aurait aimé voir Malorean Corp. se développer réellement en Europe.

La difficulté, c'est qu'un ERP ne franchit pas les frontières aussi facilement qu'un site vitrine. Les règles changent d'un pays à l'autre. Les pratiques de paie, les obligations

administratives et la façon de travailler des entreprises aussi. Hubert connaissait bien ces contraintes. Il venait du métier, de la gestion, de la comptabilité et de la finance, pas du développement logiciel. Pour lui, une fonctionnalité n'avait de valeur que si les clients pouvaient s'en servir sans se retrouver en difficulté.

À l'intérieur de l'entreprise, les rôles étaient assez lisibles. Deux équipes de cinq développeurs faisaient évoluer le logiciel. Alice et Bob, les deux Product Managers, travaillaient avec elles pour comprendre les besoins, choisir les évolutions utiles et donner une direction au produit. Alice avait été développeuse : elle pouvait entrer dans les discussions techniques sans perdre le fil. Bob venait d'une école de commerce et abordait les sujets autrement. Ils n'avaient pas le même bagage, mais travaillaient très bien ensemble.

Une équipe commerciale allait chercher de nouveaux clients et entretenait les relations existantes, épaulée par une personne au marketing. Cinq spécialistes intervenaient en avant-vente puis pendant les déploiements, sur site ou à distance. Ils ne se contentaient pas d'installer l'ERP : ils aidaient les entreprises à préciser leurs besoins, à adapter leurs processus et continuaient souvent à les accompagner pendant les six mois suivant la vente.

Le support, lui, comptait douze personnes. Sept suivaient les petits comptes ; cinq, les grands. Ce sont souvent elles qui entendaient les premières qu'une fonctionnalité obligeait un

client à contourner le logiciel plutôt qu'à s'en servir. Deux juristes veillaient aux affaires de Malorean Corp., mais apportaient également leur expertise à la conception du produit, notamment lorsqu'une évolution touchait à des règles sensibles comme celles de la paie.

Tout cela faisait fonctionner l'entreprise. Pas parfaitement, bien sûr. Un logiciel installé chez autant de clients accumule les demandes, les exceptions, les habitudes difficiles à changer et les fonctions qu'on garde faute d'avoir le temps de les refaire. Mais chacun savait à peu près ce qu'il avait à faire. Puis ChatGPT avait commencé à apparaître dans les conversations.

Au départ, il ne s'agissait pas d'un grand programme de transformation. Deux personnes, parmi les plus curieuses, l'avaient essayé quelques mois plus tôt : Julie Morel, au commerce, et Lucas Martin, au développement. Ils avaient commencé avec leurs propres usages, puis raconté ce qu'ils arrivaient à faire.

Chez Julie, le bénéfice sautait aux yeux. Elle préparait un email difficile, demandait une formulation plus claire, réorganisait des notes avant un rendez-vous, transformait une série d'idées en plan de présentation. Parfois, elle préparait aussi le contenu de quelques slides. Ce n'était pas ChatGPT qui décidait ce qu'il fallait dire au client. Elle lui donnait la matière, reprenait les propositions, corrigeait les formulations et envoyait seulement ce qu'elle assumait.

Elle ne prétendait pas que l'outil connaissait ses clients mieux qu'elle. Elle était simplement heureuse de ne plus passer une demi-heure à mettre en forme ce qu'elle savait déjà.

Philippe Garnier, son collègue, regardait cela de loin. Commercial expérimenté, habitué aux grands comptes, il connaissait les personnes derrière les contrats, les désaccords qui ne figuraient dans aucun compte rendu, les petites phrases qui changeaient le sens d'une réunion.

— Tu devrais essayer pour tes mails, lui avait dit Julie un matin. Rien que pour la mise en forme.

— Mes mails, je sais les écrire.

Il n'avait pas dit cela méchamment. Simplement, il ne voyait pas pourquoi confier à un outil une partie de son travail alors que sa valeur, pensait-il, tenait justement à sa connaissance des clients et à sa manière de leur parler.

Julie n'avait pas insisté. Elle avait un rendez-vous à préparer.

Lucas, lui, utilisait ChatGPT d'une tout autre façon. Développeur confirmé, installé à Lyon, il ne cherchait pas seulement à gagner quelques minutes sur des tâches répétitives. Il passait beaucoup de temps à expérimenter, à poser des questions de plus en plus exigeantes, à recommencer quand une réponse ne lui convenait pas. Là où Julie voulait surtout obtenir un résultat utile, Lucas cherchait jusqu'où il était possible d'aller.

Les retours des premiers utilisateurs avaient fini par remonter jusqu'à Hubert. Pour une vingtaine d'euros par personne et

par mois, le calcul semblait simple : si chacun récupérait ne serait-ce qu'un peu de temps sur les tâches fastidieuses, l'investissement se justifiait largement. Malorean Corp. avait donc décidé de mettre ChatGPT à disposition de tout le monde.

Ce n'était pas une révolution annoncée à grand renfort de présentations. Chacun pouvait s'en servir lorsqu'il y trouvait un intérêt. Certains l'ouvraient tous les jours, d'autres rarement, et Philippe pas du tout. Pour la plupart, l'IA restait un assistant : on lui expliquait ce qu'on voulait, elle proposait, on vérifiait, puis on décidait.

La vie de l'entreprise continuait.

Le mercredi, au cours d'une discussion entre les équipes Produit et Développement, un vieux sujet était revenu sur la table : la gestion des congés.

Personne n'aimait vraiment ce module. Il était ancien, rigide et proposait peu d'options. Il fonctionnait suffisamment pour qu'on continue à l'utiliser, mais pas assez bien pour qu'on cesse de recevoir des remarques à son sujet.

Un salarié qui voulait poser ses congés devait passer par son ordinateur de travail. Pas moyen de le faire simplement depuis son téléphone, un dimanche soir, en pensant à ses vacances.

Et le calcul des absences avait un défaut plus embarrassant encore : le module ne tenait pas correctement compte des jours fériés. Pour une demande du lundi au vendredi, si le

mercredi était férié, il pouvait décompter cinq jours au lieu de quatre.

Le support connaissait le problème. Les commerciaux aussi. Chez certains clients, le module était devenu un sujet d'agacement récurrent. On le gardait parce qu'il existait, pas parce qu'il répondait vraiment aux attentes.

— Il faudrait le reprendre correctement, avait résumé quelqu'un.

Personne n'était en désaccord. Mais « reprendre correctement » voulait dire revoir l'interface, traiter les règles de calcul, rendre le parcours utilisable sur mobile et remplacer une partie de ce qui avait vieilli avec le temps.

On parlait d'un chantier de l'ordre de deux mois. Il y avait d'autres priorités. Le sujet avait rejoint, une fois encore, la liste de ce qu'il faudrait faire.

Le vendredi après-midi, à la fin d'une visioconférence, Lucas demanda quelques minutes.

— J'ai quelque chose à vous montrer.

Il n'avait pas annoncé le sujet. Personne ne savait qu'il avait travaillé sur le module de congés depuis la discussion du mercredi.

Son écran apparut dans la réunion. Une interface s'ouvrit, claire, moderne, avec les informations essentielles là où on les attendait.

Lucas montra comment déposer une demande de congés. Puis il réduisit la fenêtre à la largeur d'un téléphone. L'interface

s'adapta. On pouvait continuer, choisir ses dates et confirmer sa demande sans revenir devant un ordinateur de bureau.
Il passa à un deuxième exemple : une semaine du lundi au vendredi, avec un jour férié le mercredi.
Le décompte affichait quatre jours.
Il modifia les dates. Le résultat se recalcula. Il revint à la vue mobile, fit défiler quelques écrans et enchaîna les manipulations. Ce n'était pas une succession de maquettes figées : les formulaires réagissaient, les calculs se faisaient, le parcours fonctionnait.
Au début, quelques collègues avaient posé des questions ordinaires : où se trouvait telle option, que se passait-il si l'on changeait une date ? Puis les questions s'étaient arrêtées.
Sur les petites fenêtres de la visioconférence, les visages avaient changé d'expression.
Mercredi, ils avaient parlé de deux mois de travail. Vendredi, Lucas leur montrait une première version fonctionnelle. Il ne leur avait pas demandé de l'aide. Il n'avait averti personne. Et, pour l'instant, il n'expliquait pas comment il s'y était pris.
— Attends, dit enfin une voix. C'est bien le module dont on a parlé avant-hier ?
Lucas confirma.
Un silence s'installa.
Sur les écrans, chacun regardait la même chose : un travail qu'ils pensaient devoir planifier sur des semaines, apparu en deux jours.

Personne ne comprenait comment cela avait été possible.

2 — L'IA a démultiplié la capacité de production des développeurs

Pendant des décennies, une grande partie de l'organisation du développement logiciel s'est construite autour d'une contrainte simple : **produire du logiciel coûte cher**. Transformer une idée en produit demandait du temps, des compétences rares et beaucoup de coordination. Une fonctionnalité devait être cadrée, découpée, priorisée, développée, testée, relue, déployée. Le nombre de développeurs disponibles, leur niveau d'expertise et le temps qu'ils pouvaient consacrer à un sujet limitaient mécaniquement ce qu'une entreprise pouvait entreprendre. Cette rareté a façonné les organisations. On a appris à protéger le temps des développeurs, à préparer leur travail en amont, à réduire les interruptions, à organiser des backlogs et à arbitrer longtemps avant de lancer la construction. Une bonne partie du fonctionnement Produit moderne s'est développée dans ce contexte : il fallait éviter de consommer une ressource Engineering coûteuse sur une mauvaise idée. L'arrivée des assistants de code, puis des agents capables d'exécuter des tâches plus larges, commence à modifier cette équation.

Le changement n'est plus seulement une meilleure autocomplétion

Les premiers outils d'IA pour développeurs pouvaient être compris comme une nouvelle génération d'outils de productivité : compléter une fonction, expliquer un morceau de code, écrire un test, proposer un refactoring.

La génération suivante change drastiquement la nature du travail. Un agent peut recevoir une tâche, explorer une base de code, modifier plusieurs fichiers, lancer des tests, corriger certaines erreurs et rendre un changement prêt à être relu. Un développeur peut donc déléguer une partie majeure de l'exécution et, surtout, mettre en mouvement davantage de travail qu'auparavant.

La conséquence n'est pas seulement que « le développeur code plus vite ».

La quantité de travail logiciel qu'un individu peut initier et faire avancer augmente drastiquement.

Certaines entreprises commencent à publier des résultats spectaculaires. AMPECO, qui décrit son propre passage à un système d'ingénierie AI-native, rapporte une vélocité quatre fois supérieure et 50 % de bugs en moins par story. Ce chiffre ne doit pas être traité comme une moyenne du marché : il s'agit du résultat déclaré par une entreprise, dans un contexte particulier. Mais il illustre l'ordre du changement recherché : il ne s'agit plus seulement de gagner quelques minutes sur

l'écriture d'une fonction, mais d'augmenter fortement la capacité d'exécution d'une équipe. [1]

Ce résultat spectaculaire revendiqué par une entreprise n'est pas le seul type d'indice disponible. J'en ai fait moi-même l'expérience avec Claude Code. En quatre jours et plus de 180 commits, j'ai construit une première version de produit avec des écrans, des API et des tests. Avec mes anciennes méthodes, j'estime que ce travail m'aurait pris un à deux mois. J'ai raconté cette expérience sur Malorean.net en juillet 2026. Elle ne permet évidemment pas de généraliser un gain de productivité : elle montre ce qu'un PM ayant déjà développé peut désormais réaliser, mais aussi les difficultés de qualité, de maintenabilité et de maîtrise du code généré. [7]

Ce point change l'échelle de la question.

Si l'IA ne faisait gagner que quelques pourcents sur le code, l'organisation pourrait rester identique et absorber le gain. Lorsque la capacité de production augmente beaucoup plus vite, les autres contraintes deviennent relativement plus importantes.

Accélérer le code ne suffit pas à accélérer l'entreprise

Une équipe logicielle n'est pas une chaîne dont le seul maillon serait l'écriture du code.

Avant de construire, il faut comprendre un problème, décider qu'il mérite un investissement, choisir une direction et clarifier certaines contraintes. Après avoir construit, il faut

relire, tester, sécuriser, déployer, observer et maintenir. Entre les deux, il faut continuellement arbitrer.

Si un seul de ces maillons accélère fortement, il pousse simplement davantage de travail vers les autres.

Le code peut être produit plus vite sans que la compréhension client accélère au même rythme. Les prototypes peuvent se multiplier sans que la capacité à décider lequel mérite d'être poursuivi augmente. Davantage de changements peuvent arriver en review sans que l'attention humaine disponible pour les examiner soit multipliée dans les mêmes proportions.

Le problème n'est donc pas que l'IA rende soudainement inutiles les autres activités.

Il est qu'elle **déplace les goulots d'étranglement**.

Ce qui était auparavant masqué par la lenteur de l'exécution devient plus visible : qualité du problème choisi, clarté de l'intention, arbitrage, validation, cohérence et capacité à apprendre après livraison.

Produire plus n'est pas créer plus de valeur

Cette distinction devient centrale.

Lorsque produire était difficile, le volume de logiciel livré pouvait sembler proche de la capacité de l'organisation à créer de la valeur. Ce raccourci était déjà imparfait. Avec l'IA, il devient dangereux.

Une équipe peut désormais générer davantage de code, davantage de pull requests, davantage de prototypes et davantage de fonctionnalités sans améliorer le produit. Elle

peut même détériorer sa performance globale si ce volume supplémentaire augmente les défauts, le rework, les coûts ou la complexité à maintenir.

Autrement dit :

la capacité à fabriquer n'est pas la capacité à choisir ce qui mérite d'être fabriqué.

C'est pourquoi les métriques d'activité deviennent encore moins suffisantes. Le nombre de changements, la quantité de code ou le simple usage d'un outil IA ne disent pas si une équipe a créé davantage de valeur.

Canva fournit un exemple intéressant de ce déplacement vers l'impact : l'entreprise a développé un système permettant aux équipes Product, Engineering et Data d'estimer l'impact commercial de leurs expérimentations et de relier plus directement les décisions locales à leurs conséquences business. [2]

La question change alors de nature.

Avant :

combien pouvons-nous construire ?

Après :

parmi tout ce que nous pouvons désormais construire, qu'est-ce qui mérite réellement de l'être ?

> ***Le problème de la productivité logicielle a plus de quarante ans***

> *En 1986, Fred Brooks présente* No Silver Bullet*, publié dans* IEEE Computer *en 1987. Il distingue les difficultés* ***accidentelles*** *de la fabrication du logiciel — langages, outils, traduction d'une conception en code — de sa complexité* ***essentielle*** *: comprendre, concevoir, articuler et faire évoluer un système. Aucun progrès technique ou managérial isolé ne promet, selon lui, de multiplier par dix en une décennie la productivité, la fiabilité et la simplicité du développement. [3]*
> *Cet essai sera repris en 1995 dans l'édition anniversaire de* The Mythical Man-Month*, paru initialement en 1975. [4] L'IA générative change potentiellement l'échelle de l'exécution, mais pas automatiquement la justesse du problème choisi. C'est une mise en perspective actuelle de Brooks,* ***pas une prédiction de sa part sur les agents d'IA****.*

Le nouveau goulot est le jugement

À mesure que l'exécution devient moins coûteuse, plusieurs ressources prennent relativement plus de valeur : la connaissance du client, la compréhension du problème, le contexte stratégique, la capacité à formuler une intention claire, l'évaluation du risque, le goût produit, la capacité à dire non et la responsabilité de la décision.

Dans un article publié sur Malorean.net en mars 2026, je défendais cette idée : à mesure que les outils techniques deviennent plus accessibles, l'avantage concurrentiel se déplace vers la compréhension du contexte — métiers,

utilisateurs, clients, contraintes techniques, concurrence et réglementations. C'est une analyse stratégique, pas un résultat expérimental. Elle précise ce que signifie exercer son jugement : comprendre assez bien une situation pour choisir la bonne chose à construire, et pas seulement savoir la produire. [8]

L'IA rend le « comment » plus accessible.

Elle ne rend pas automatiquement le « pourquoi » évident. Elle permet de produire une solution avant que l'organisation ait terminé de comprendre le problème. Elle permet de réaliser rapidement une demande client dont l'entreprise aurait dû décider qu'elle ne correspondait pas à son marché. Elle permet de multiplier les variantes d'une fonctionnalité sans savoir laquelle produit réellement un meilleur outcome. Plus la construction devient facile, plus la discipline consistant à choisir ce que l'on ne construit pas devient importante.

Ramp illustre bien ce principe : son CPO insiste sur la nécessité de clarifier les ICP prioritaires, les problèmes que l'entreprise veut résoudre particulièrement bien et d'accepter de refuser certaines demandes — y compris lucratives — lorsqu'elles tirent le produit dans une autre direction. [5]

En juillet 2026, Marty Cagan a nommé ce décalage le *paradoxe de productivité de l'IA*. Son diagnostic est que construire plus vite ne corrige pas le problème des idées qui ne méritaient pas d'être construites. Il distingue ainsi ce que

l'on fabrique pour apprendre de ce que l'on livre pour produire un résultat. C'est l'analyse d'un praticien du Produit, non une mesure démontrant que toutes les entreprises sont touchées de la même manière. [6]

Le paradoxe de l'IA est donc le suivant :

elle augmente fortement notre capacité d'action et rend, par conséquent, la sélection de nos actions plus importante qu'avant.

Une transformation de système, pas seulement d'outil

On pourrait répondre à cette évolution en distribuant simplement un assistant de code à chaque développeur et en demandant aux équipes de continuer comme avant.

Cela produit des gains locaux.

Mais à mesure que ces gains augmentent, l'organisation rencontre ses autres limites : capacité à comprendre, à décider, à coordonner, à vérifier et à absorber ce qui est produit.

À ce stade, le sujet n'est plus seulement l'adoption d'un outil. Il devient organisationnel.

Une organisation construite lorsque l'Engineering était la ressource la plus rare n'a aucune raison de rester optimale lorsque cette rareté diminue.

Une question devient alors inévitable :

que devient une organisation conçue pour protéger une ressource rare — la capacité Engineering — lorsque cette capacité devient beaucoup plus abondante ?

Bibliographie

[1] How we built an AI-native engineering system — AMPECO

- URL : https://www.ampeco.com/blog/how-we-built-an-ai-native-engineering-system/
- Retour d'expérience d'une organisation Engineering reconstruite autour d'agents, avec une forte hausse de capacité revendiquée par l'entreprise.

[2] Measuring Commercial Impact at Scale — Canva

- URL : https://www.canva.dev/blog/engineering/measuring-commerical-impact-at-scale/
- Décrit comment Canva relie les expérimentations des équipes Produit et Engineering à une estimation accessible de leur impact commercial.

[3] No Silver Bullet: Essence and Accidents of Software Engineering — Fred Brooks

- URL : https://www.cgl.ucsf.edu/Outreach/pc204/NoSilverBullet
- Essai présenté en 1986 et publié en 1987, distinguant complexité essentielle et difficultés accidentelles de la construction logicielle.

[4] The Mythical Man-Month, édition anniversaire, chapitre 16 — Fred Brooks

- URL : https://www.oreilly.com/library/view/mythical-man-month-the/0201835959/ch16.xhtml

- Sommaire éditorial de l'édition 1995 attestant l'intégration ultérieure de No Silver Bullet ; aperçu de chapitre seulement.

[5] 5 insights from Ramp's CPO on building for hypergrowth — Ramp

- URL : https://ramp.com/blog/5-insights-from-ramps-cpo-reddit-ama
- Le CPO de Ramp insiste sur le focus, les ICP prioritaires et la nécessité de refuser les demandes qui dispersent.

[6] The AI Productivity Paradox — Marty Cagan, Silicon Valley Product Group (2026)

- URL : https://www.svpg.com/the-ai-productivity-paradox/
- Analyse de juillet 2026 distinguant les gains de vitesse permis par l'IA des résultats réellement créés ; réflexion de praticien, non démonstration causale.

[7] Quatre jours de vibe coding dans la peau d'un PM rouillé — Laurent Deséchalliers, Malorean.net (2026)

- URL : https://malorean.net/articles/2026-07-28-quatre-jours-de-vibe-coding-dans-la-peau-d-un-pm-rouille.html
- Retour d'expérience personnel avec Claude Code : vitesse de construction d'un produit, difficultés techniques rencontrées, tests et maîtrise de la qualité ; ni étude contrôlée ni preuve généralisable.

[8] Dans le logiciel, l'avantage ne sera plus la technologie. Ce sera la compréhension du contexte — Laurent Deséchalliers, Malorean.net (2026)

- URL : https://malorean.net/articles/2026-03-23-dans-le-logiciel-lavantage-ne-sera-plus-la-technologie-ce-sera-la-comprehension-du-contexte.html
- Essai stratégique de l'auteur sur le déplacement de la rareté vers la compréhension du contexte client, métier, réglementaire et technique, et non démonstration quantitative.

3 — Malorean Corp. — Le jour d'après

— J'ai utilisé l'IA.

Lucas avait prononcé cette phrase presque naturellement, comme si elle suffisait à expliquer ce que tout le monde venait de voir.

Il était toujours en partage d'écran. Le nouveau module de congés occupait la majeure partie de la fenêtre. Quelques minutes plus tôt, il avait montré comment demander une absence depuis un téléphone, puis comment le logiciel tenait compte des jours fériés dans le calcul.

— Attends, reprit Alice. Quand tu dis que tu as utilisé l'IA, tu veux dire pour t'aider à écrire le code ?

— Oui. Enfin, pas seulement. Pour concevoir le module, comprendre le code existant, développer les écrans, faire les calculs, écrire les tests… Un peu pour tout, en fait.

— Et tu as fait ça depuis mercredi ?

— Mercredi après-midi. J'ai fini les derniers trucs ce matin.

Alice regarda les visages sur son écran. Quelques personnes souriaient. D'autres semblaient chercher ce qui leur avait échappé.

— Mais tu n'avais pas de spécifications, remarqua-t-elle.

— Non. On en avait parlé mercredi. Je connaissais le module et ses problèmes. J'ai commencé par demander à l'IA

d'analyser ce qui existait. Ensuite, on a avancé morceau par morceau.

— Et tout fonctionne ?

Lucas eut un petit rire.

— Tout, je ne sais pas. Ce que je vous ai montré, oui. Il reste sûrement des choses à vérifier.

Thomas, qui travaillait dans la même équipe que lui, intervint pour la première fois.

— Et ça ressemble à quoi, le code derrière ?

— Tu pourras regarder. J'ai fait pas mal de tests.

— Je regarderai, répondit Thomas.

Il n'avait pas l'air particulièrement impressionné. Ou peut-être l'était-il sans avoir envie de le montrer.

Hubert Félix Duca était resté silencieux jusque-là.

Il avait assisté à suffisamment de démonstrations pour savoir qu'un logiciel qui fonctionne pendant dix minutes n'est pas nécessairement un logiciel prêt à être utilisé par des centaines d'entreprises. Il connaissait les exceptions, les besoins particuliers des grands comptes, les problèmes qui n'apparaissent qu'une fois chez le client.

Mais il savait aussi ce qu'il avait entendu mercredi.

Deux mois.

Et il avait sous les yeux quelque chose qui ressemblait déjà beaucoup à ce qu'ils avaient imaginé.

— Lucas, demanda-t-il, tu peux me redire combien de temps tu as passé là-dessus ?

— Deux jours, à peu près.

Hubert se pencha légèrement vers son écran.

— D'accord.

Il ne trouva rien à ajouter.

La réunion se termina quelques minutes plus tard. Chacun retourna à ses occupations, mais personne ne reparla vraiment de ses tâches de l'après-midi.

Le samedi matin, Alice repensa à la démonstration en préparant son café.

Ce qui la troublait n'était pas seulement la vitesse de Lucas. C'était le fait qu'il n'avait eu besoin de personne.

Elle connaissait bien son métier de Product Manager. Avant de lancer un développement, elle cherchait à comprendre le problème, confrontait les demandes, discutait avec les clients, travaillait avec les développeurs pour trouver une solution raisonnable.

Sur un sujet comme le module de congés, elle aurait probablement organisé plusieurs échanges. Elle aurait aidé à préciser les règles, à arbitrer les cas particuliers, puis à décider ce qui devait être développé en premier.

Lucas n'avait rien fait de tout cela.

Il avait entendu une discussion, pris une initiative et construit une solution.

Une partie d'elle était admirative. Une autre se demandait ce qui se passerait si tous les développeurs commençaient à travailler de cette manière.

Depuis plusieurs mois, pourtant, son propre quotidien avait considérablement changé.

Au début, Alice utilisait l'IA pour résumer des comptes rendus ou reformuler des documents. Puis elle avait commencé à lui confier des tâches plus importantes. Elle avait automatisé une partie du traitement des retours clients, la préparation de certaines synthèses, le regroupement de demandes similaires et plusieurs opérations administratives qui lui prenaient auparavant des heures.

Il lui arrivait désormais de terminer en quelques minutes un travail qui occupait autrefois une bonne partie de sa matinée.

Elle ne regrettait aucune de ces tâches.

Au contraire.

Elle avait enfin davantage de temps pour parler aux clients, comprendre les problèmes qui se cachaient derrière leurs demandes et réfléchir à ce qu'il fallait vraiment construire.

L'IA n'avait pas rendu son métier moins intéressant.

Elle l'avait débarrassé d'une partie de ce qui l'empêchait de bien le faire.

Alors pourquoi les développeurs devraient-ils, eux, continuer à travailler exactement comme avant ?

Alice reposa sa tasse.

La question lui paraissait évidente.

La réponse beaucoup moins.

Hubert avait lui aussi passé une bonne partie du week-end à réfléchir.
Son entreprise avait toujours eu plus d'idées que de capacité pour les réaliser. Des fonctionnalités attendaient depuis des mois. Certaines améliorations demandées par les clients étaient systématiquement repoussées. Et il y avait ce projet d'expansion européenne, qui impliquait encore beaucoup de travail sur le logiciel.
Il connaissait l'obstacle par cœur : ils n'avaient pas assez de temps pour tout faire.
Vendredi, Lucas avait peut-être montré que cet obstacle n'était plus aussi infranchissable.
Si un chantier évalué à deux mois pouvait produire une première version en deux jours, combien d'autres projets devenaient soudain envisageables ?
Dix développeurs travaillaient chez Malorean Corp. Hubert avait essayé d'imaginer ce qui se passerait si chacun pouvait avancer aussi vite.
Puis il avait essayé d'imaginer dix développeurs lançant chacun leurs propres fonctionnalités, sans coordination.
Cette deuxième image lui avait nettement moins plu.
Il avait envisagé d'appeler Alice dès le samedi. Il s'était retenu.
Le dimanche soir, il lui avait simplement envoyé un message.

Tu as un moment demain matin ? J'aimerais qu'on reparle de vendredi.
La réponse était arrivée quelques minutes plus tard.
Oui. Moi aussi.

Le lundi matin, à neuf heures, Alice et Hubert se retrouvèrent en visioconférence.
— Alors ? demanda Hubert.
— Alors quoi ?
— Tu as compris ce qui s'est passé vendredi ?
Alice sourit.
— Techniquement, à peu près. Lucas a utilisé l'IA pour développer beaucoup plus vite qu'on ne l'aurait imaginé.
— Et le reste ?
— C'est justement le problème.
Hubert hocha la tête.
— J'ai passé le week-end à y penser. On manque de capacité depuis des années. On refuse des demandes, on repousse des projets, on passe notre temps à choisir ce qu'on ne va pas faire. Et là, Lucas nous sort en deux jours quelque chose qu'on avait évalué à deux mois.
— Une première version, précisa Alice. On ne sait pas encore si elle est prête pour la production.
— Oui, je sais. Mais même s'il faut encore deux semaines pour la terminer, ça change beaucoup de choses.
Alice ne pouvait pas le contredire.

— Ce qui m'embête, reprit Hubert, c'est qu'il a fait tout ça sans passer par toi.

Alice ne répondit pas immédiatement.

— Ça m'a embêtée aussi, admit-elle.

— Tu penses qu'il faut lui rappeler comment on travaille ?

— J'y ai pensé.

— Et ?

— Et je crois que ce serait une erreur.

Hubert fronça les sourcils.

— Pourtant, s'il commence à décider tout seul de ce qu'il développe, on risque d'avoir quelques surprises.

— Bien sûr. Mais regarde ce qui s'est passé. Mercredi, on dit qu'il faudrait refaire le module de congés. Si on avait suivi notre fonctionnement habituel, j'aurais commencé à recueillir les besoins, on aurait organisé des réunions, préparé le sujet, découpé le travail… On aurait probablement passé plusieurs semaines avant de montrer quoi que ce soit.

— Ce n'est pas forcément du temps perdu.

— Non. Et je ne dis pas qu'il faut arrêter de comprendre les besoins. Mais Lucas nous a montré qu'il pouvait avancer sans attendre tout ça.

— Donc tu proposes qu'on laisse les développeurs décider eux-mêmes ?

— Non, justement.

Alice réfléchit quelques secondes.

— Je crois qu'on mélange deux choses. Décider ce qu'on doit faire et décider comment le faire.

— Les deux sont liées.

— Oui. Mais ça ne veut pas dire que je dois intervenir dans chaque décision.

Elle se redressa légèrement.

— Tu sais, mon métier a énormément changé ces derniers mois. J'utilise l'IA tous les jours. J'ai automatisé un tas de choses que je faisais à la main. Je passe beaucoup moins de temps à préparer des documents ou à organiser l'information.

— Je sais, tu m'en as parlé.

— Mais tu ne te rends peut-être pas compte à quel point ça a changé mon travail. Aujourd'hui, j'ai beaucoup plus de temps pour ce qui compte vraiment : comprendre les clients, identifier les problèmes et décider où investir notre énergie.

Hubert l'écoutait.

— Avant, une partie de mon travail consistait à faire avancer les sujets dans l'organisation. Préparer les tickets, préciser les demandes, répondre aux questions, coordonner les échanges. Et si demain les développeurs avaient beaucoup moins besoin de moi pour ça ?

— Tu aurais plus de temps.

— Exactement. Et eux aussi.

Elle marqua une pause.

— On accepte que l'IA transforme mon métier. Pourquoi est-ce qu'on voudrait que celui des développeurs reste identique ?

Hubert s'appuya contre le dossier de son fauteuil.

— D'accord. Mais il faut quand même que quelqu'un décide de ce qu'on construit.

— Évidemment. Et surtout pourquoi on le construit.

— Alors comment tu vois les choses ?

Alice hésita.

Elle n'avait pas préparé de présentation. Elle n'avait aucun modèle d'organisation à proposer, aucun nouveau processus à déployer.

Elle avait seulement une intuition.

— Je me demande si on ne devrait pas essayer quelque chose.

Hubert attendit qu'elle poursuive.

— On pourrait confier un sujet à Lucas. Un vrai problème client. Et lui laisser beaucoup plus de liberté pour le résoudre.

— Sans spécifications ?

— Sans lui donner une solution toute faite. On lui explique le problème, on lui donne accès aux personnes qui le connaissent, et on le laisse chercher.

— Et toi ?

— Je reste disponible. Je l'aide à comprendre les besoins, à challenger ses idées, à éviter de développer quelque chose

simplement parce qu'il est capable de le faire. Mais je ne lui dicte pas chaque étape.

Hubert réfléchit.

— Tu veux donc qu'il fasse une partie du travail de Product Manager ?

— Peut-être. Ou simplement qu'on arrête de découper systématiquement le travail entre nous.

— Et si ça ne marche pas ?

— On le verra.

Hubert sourit.

— C'est un peu léger comme méthode.

— Je sais.

Alice sourit à son tour.

— Mais je préfère qu'on essaie sur un sujet limité plutôt que de passer trois mois à discuter d'une nouvelle organisation.

Hubert ne répondit pas tout de suite.

Puis il sembla se souvenir de quelque chose.

— On a justement un problème qui remonte depuis plusieurs semaines chez certains clients de l'agroalimentaire.

Alice acquiesça. Elle voyait très bien lequel.

L'ERP de Malorean Corp. gérait les stocks et les opérations de fabrication, mais il ne permettait pas de suivre correctement les lots de matières premières utilisés pour fabriquer les produits finis.

Chez un fabricant alimentaire, lorsqu'un fournisseur signalait qu'un lot de matière première était défectueux, il fallait

pouvoir retrouver rapidement les produits fabriqués avec ce lot.

Aujourd'hui, les clients de Malorean Corp. devaient reconstituer ces informations par leurs propres moyens.

— J'ai encore eu un retour là-dessus la semaine dernière, expliqua Hubert. Un client m'a dit qu'il avait dû passer une journée entière avec son responsable qualité pour retrouver les produits concernés par une alerte fournisseur.

— Oui. Le support nous a remonté plusieurs cas du même genre.

— Tu penses que Lucas pourrait travailler là-dessus ?

Alice réfléchit.

Le sujet était plus délicat que le module de congés. Il ne suffisait pas de créer quelques écrans. Il fallait comprendre ce qui se passait réellement dans les ateliers, comment les matières premières étaient utilisées et comment les clients identifiaient leurs produits.

Mais c'était précisément ce qui l'intéressait.

— Je pense qu'on pourrait lui confier le problème, oui.

— Tu crois qu'il connaît suffisamment le métier ?

— Probablement pas.

Hubert leva les sourcils.

— C'est rassurant.

— Moi non plus, je ne connais pas tous les détails. Mais je sais à qui parler. On a des clients qui connaissent parfaitement leurs contraintes. On a des gens au support et au

déploiement qui voient les difficultés tous les jours. Lucas peut aller les voir.

— Tu veux qu'il parle directement aux clients ?

— Pourquoi pas ?

— D'habitude, c'est plutôt ton travail.

— Justement.

Hubert ne semblait pas entièrement convaincu.

— Et s'il construit quelque chose qui ne répond pas au besoin ?

— C'est à moi de l'aider à éviter ça. Mais pas en lui écrivant une spécification de cinquante pages.

— Et s'il veut mettre sa solution en production ?

— Il faudra qu'elle soit vérifiée et validée, comme n'importe quelle autre évolution du logiciel. Ce n'est pas parce qu'il travaille plus librement qu'on lui donne le droit de prendre tous les risques.

Hubert resta quelques instants silencieux.

— Tu sais ce qui m'inquiète ? demanda-t-il finalement. Ce n'est pas seulement qu'il puisse faire une erreur. C'est qu'on ne sache plus qui fait quoi.

— Moi aussi, ça m'inquiète.

— Et tu veux quand même essayer ?

— Oui.

Elle prit le temps de réfléchir avant d'ajouter :

— Parce que l'autre possibilité, c'est de continuer exactement comme avant, en demandant simplement aux développeurs de

coder plus vite. Je ne suis pas sûre que ce soit ce que Lucas nous a montré vendredi.

Ils discutèrent encore une vingtaine de minutes.
Hubert voulait éviter de perturber les projets en cours. Alice ne voulait pas que l'expérience devienne une initiative spectaculaire annoncée à toute l'entreprise avant même d'avoir commencé.
Ils convinrent donc de faire simple.
Lucas travaillerait sur le problème de traçabilité des lots. Il aurait la liberté d'explorer les solutions et de rencontrer directement les personnes capables de l'aider à comprendre le besoin.
Alice serait sa référente Produit. Elle ne lui préparerait pas un backlog détaillé, mais resterait impliquée dans les décisions importantes.
Ils commenceraient par un périmètre limité, sans promettre de date de livraison à un client.
Et surtout, ils prendraient le temps d'observer ce qui se passerait.
— On devrait quand même définir comment on saura si l'expérience est réussie, remarqua Hubert.
— Oui. Et je ne veux pas qu'on mesure seulement le nombre de fonctionnalités qu'il aura développées.
— Qu'est-ce que tu veux mesurer alors ?

— Déjà, voir s'il arrive à comprendre le problème et à proposer quelque chose d'utile. Ensuite, voir jusqu'où il peut avancer sans qu'on ait besoin de tout organiser pour lui.

— Et combien de temps ça nous prend à nous.

— Aussi.

Hubert regarda l'heure.

— Bon. On tente.

Alice sourit, sans se sentir complètement rassurée.

— On tente.

— Tu lui en parles aujourd'hui ?

— Oui.

La visioconférence se termina.

Alice resta quelques instants devant son écran. Elle ouvrit sa messagerie et chercha Lucas.

Elle commença à écrire :

Salut Lucas. Tu aurais un moment aujourd'hui ? J'aimerais te proposer un sujet un peu différent de ce qu'on fait d'habitude.

Elle relut son message avant de l'envoyer.

Pour la première fois depuis longtemps, elle allait confier un problème à un développeur sans avoir elle-même commencé à imaginer précisément la solution.

Elle ne savait pas encore si c'était une bonne idée.

Et elle n'était plus certaine que son rôle consistait à le savoir avant lui.

4 — De nouveaux modèles d’organisation apparaissent

L’IA rend davantage d’exécution possible. Mais comment les entreprises réorganisent-elles alors les équipes et les décisions ? Leurs réponses divergent.

Certaines renforcent le Product Management autour des développeurs. D’autres élargissent le rôle de ces derniers, donnent davantage de moyens aux métiers non techniques ou réorganisent le travail autour de petites équipes assistées par des agents.

La répartition historique du travail cesse d’être une évidence.

Le modèle traditionnel avait pourtant une logique : le Product Management concentrait une grande partie du contexte client et business, le Design matérialisait l’expérience et l’Engineering transformait ces intentions en logiciel. Tickets, spécifications, passages de relais et réunions permettaient de synchroniser ces métiers autour d’une capacité de développement limitée.

Cette organisation ne devient pas soudainement mauvaise. Mais lorsque la capacité d’exécution augmente, plusieurs entreprises commencent à déplacer différemment les personnes, le contexte et le travail.

Il ne s’agit donc pas encore de chercher « le bon modèle ».

Il s'agit d'observer les principales réponses qui apparaissent.

Renforcer le Produit autour des développeurs

La réponse la plus intuitive consiste à conserver une séparation relativement classique entre Product et Engineering, mais à augmenter la capacité Produit disponible autour des développeurs.

Gamma fournit un exemple intéressant. L'entreprise rapporte un ratio d'environ un Product Manager pour quatre engineers, là où le ratio historique souvent cité est plutôt proche d'un pour dix. [1]

Dans ce modèle, l'augmentation de la capacité Engineering s'accompagne donc d'une densification du Product Management.

Le choix est simple dans son principe : si davantage de choses peuvent être construites, davantage de capacité est consacrée à comprendre, prioriser et orienter ce travail.

Donner au développeur une plus grande part du sujet

Une autre stratégie consiste à élargir le périmètre de l'engineer.

Le développeur n'attend plus nécessairement qu'un problème soit entièrement traduit en solution puis en tickets. Il peut participer davantage à la compréhension du besoin, au prototypage, aux arbitrages locaux et au suivi du résultat.

Dans son offre d'emploi pour un Product Engineer, Linear décrit les responsabilités attendues : porter une fonctionnalité de l'idée au produit livré, en participant à la définition du

problème sans dépendre constamment du Product Management. [2]

Ce modèle cherche à réduire certains passages entre métiers en concentrant davantage de contexte et de capacité d'action sur la même personne.

Il ne signifie pas que chaque développeur devient PM.

Il décrit plutôt un élargissement du rôle Engineering sur certains sujets.

Laisser les métiers franchir leurs anciennes frontières

Le mouvement ne concerne pas seulement les développeurs. Figma rapporte dans son étude sur l'évolution des rôles que 64 % des répondants s'identifient à au moins deux rôles et que plus d'un tiers déclarent des responsabilités couvrant plusieurs métiers. Ce résultat ne peut pas être attribué uniquement à l'IA, mais il décrit bien des frontières de rôle moins rigides. [3]

Atlassian documente également des workflows où des designers travaillent directement dans le vrai code produit et ouvrent eux-mêmes une pull request avant review Engineering. [4]

Cursor décrit de son côté une organisation où Product, Design et Engineering se rapprochent davantage de la construction. [5]

Le constat est donc moins « tout le monde fait tout » que **davantage de personnes interviennent hors de leur périmètre historique**.

Cette porosité soulève ensuite une question distincte : si plusieurs métiers peuvent agir sur le même terrain, comment maintenir des responsabilités claires ?
Nous y reviendrons.

Confier davantage de travail à de petites équipes responsables

Une autre famille de réponses consiste à concentrer davantage de portée sur moins de personnes.
AMPECO décrit des « alignment syncs » où un engineer et un Product Owner définissent l'intention avant qu'une grande partie de l'analyse, du développement et des tests soit prise en charge par des agents. [6]
Taito décrit un fonctionnement où l'équipe valide un plan puis une ou deux personnes pilotent la construction avec des agents. [7]
Dans ces modèles, une petite unité humaine peut mettre en mouvement davantage de travail qu'auparavant.
Le changement organisationnel observable est donc une réduction possible des passages entre métiers et une concentration du sujet autour d'une personne ou d'un petit groupe.
Reste à définir ce que ces personnes peuvent décider seules et quand elles doivent solliciter un responsable de domaine.

Automatiser une partie des vérifications

Une production plus rapide exerce aussi une pression sur les mécanismes de validation.

Certaines entreprises répondent en automatisant davantage de contrôles ou en différenciant le traitement des changements selon leur risque.

Meta utilise un Diff Risk Score pour estimer le risque qu'un changement provoque un incident. [8]

Cloudflare formalise des standards Engineering qui peuvent être contrôlés automatiquement avant merge. [9]

Taito combine également règles déterministes, agents de review et déclencheurs de risque. [7]

Le constat important, à ce stade, est simplement que **la validation n'est plus nécessairement organisée de manière uniforme**.

La façon de combiner automatisation, responsabilité humaine et seuils d'escalade sera développée plus loin.

Donner du contexte plutôt que prescrire chaque étape

D'autres entreprises vont plus loin : elles remettent en cause le ticket détaillé comme principal moyen de coordonner le travail.

Linear critique explicitement le modèle dans lequel le PM scope un problème, l'écrit dans un système puis attend qu'Engineering l'exécute. L'entreprise insiste sur la nécessité de rendre accessibles aux humains et aux agents le feedback client, les idées, la direction stratégique, les décisions et le code. [10]

Cursor insiste également sur la nécessité de fournir des objectifs suffisamment clairs pour permettre davantage d'initiative locale. [5]

Le changement observé n'est donc pas la disparition de toute coordination.

C'est le déplacement d'une partie de cette coordination depuis la prescription détaillée vers **un contexte plus riche et plus directement accessible**.

Ce que doit contenir ce contexte, comment le maintenir et comment définir les limites de l'autonomie feront l'objet d'un chapitre dédié.

Permettre à celui qui connaît le besoin de construire

Certaines entreprises permettent même aux équipes qui connaissent le besoin de réaliser elles-mêmes de petits outils, sans toujours passer par la file d'attente des développeurs.

Shopify a créé Quick, une plateforme interne permettant à davantage de collaborateurs de publier rapidement des sites et petits outils. [11]

Vercel décrit un modèle où des collaborateurs non techniques construisent des outils internes dans un environnement sécurisé sans détourner les développeurs des priorités du produit principal. [12]

Atlassian montre également des designers intervenant dans le code avant relecture technique. [4]

Dans ces cas, l'IA ne rend pas seulement les développeurs plus rapides.

Elle permet parfois de déplacer l'exécution vers **la personne qui possède déjà le contexte du besoin**.

Un principe plus ancien que l'IA

En 2018, Marty Cagan défendait déjà ce qu'il appelait des *empowered product teams* : des équipes pluridisciplinaires auxquelles la direction confie des problèmes à résoudre, plutôt qu'une liste de fonctionnalités à livrer. Il leur demande de chercher la solution avec les compétences Produit, Design et Engineering, et de répondre du résultat plutôt que du volume produit. [13]

Avec Chris Jones, il développera cette approche dans le livre *EMPOWERED* en 2020. Ce modèle n'annonçait ni des métiers interchangeables ni la fin des managers : il supposait au contraire une vision partagée, des compétences solides et des responsables qui accompagnent les équipes. [14]

Cette idée n'a donc pas attendu les agents. Ce que l'IA change potentiellement, c'est la capacité d'exécuter plus vite et plus directement certaines solutions que l'équipe autonome a choisi d'explorer. **L'autonomie Produit est une ambition plus ancienne ; l'IA en déplace les moyens et les contraintes.**

Des choix qui ne convergent pas encore

Ces réponses ne sont pas les étapes d'une même transformation. Renforcer le Produit et confier davantage de responsabilités Produit aux développeurs peuvent même sembler contradictoires. Elles peuvent pourtant fonctionner

dans des entreprises différentes, voire coexister au sein d'une même organisation.

Leur point commun est ailleurs : **quand l'exécution devient plus accessible, la façon de répartir le contexte, les décisions et les responsabilités redevient un choix.**

Il reste à voir ce que ce déplacement provoque dans le quotidien des personnes concernées, au-delà des annonces et des organigrammes.

Nous retrouverons bientôt Malorean Corp. pour observer ce que ces modèles changent dans la vie d'une petite entreprise française. Nous examinerons ensuite les responsabilités et les règles nécessaires à cette autonomie.

Bibliographie

[1] The bottleneck keeps shifting: What AI is changing about how we build — Atlassian / Gamma

- URL : https://www.atlassian.com/blog/how-we-build/the-bottleneck-keeps-shifting-what-ai-is-changing-about-how-we-build
- Décrit la densification du Product Management chez Gamma lorsque la capacité de production Engineering augmente.

[2] Product Engineer — Linear Careers

- URL : https://linear.app/careers/0c7c2e26-0a98-42cf-a47c-9a3999fb513b

- Offre d'emploi présentant les responsabilités attendues d'un Product Engineer, de la définition du problème à la livraison.

[3] Are Roles and Responsibilities a Thing of the Past? — Figma

- URL : https://www.figma.com/blog/2025-shifting-roles-report/
- Documente la porosité croissante des rôles et l'élargissement des responsabilités au-delà des frontières métier traditionnelles.

[4] Designers' workflow for shipping code — Atlassian

- URL : https://www.atlassian.com/blog/ai-at-work/designers-workflow-for-shipping-code
- Décrit des designers intervenant directement dans le code produit avant une validation Engineering.

[5] How Cursor Is Navigating the New Shape of Product Craft — Atlassian / Cursor

- URL : https://www.atlassian.com/blog/how-we-build/how-cursor-is-navigating-the-new-shape-of-product-craft
- Montre un fonctionnement où Product, Design et Engineering se rapprochent davantage de la construction.

[6] How we built an AI-native engineering system — AMPECO

- URL : https://www.ampeco.com/blog/how-we-built-an-ai-native-engineering-system/
- Décrit une organisation où quelques humains cadrent l'intention avant de déléguer une grande partie de l'exécution à des agents.

[7] Inside an AI-native engineering organization — Swarmia / Taito.ai

- URL : https://www.swarmia.com/blog/inside-an-ai-native-engineering-organization/
- Décrit une petite équipe pilotant des agents, avec une grande part d'exécution concentrée sur une ou deux personnes.

[8] Diff Risk Score: AI-driven risk-aware software development — Meta

- URL : https://engineering.fb.com/2025/08/06/developer-tools/diff-risk-score-drs-ai-risk-aware-software-development-meta/
- Présente un score de risque utilisé pour différencier les workflows de validation selon la criticité probable des changements.

[9] How Cloudflare enforces engineering standards using AI — Cloudflare

- URL : https://blog.cloudflare.com/engineering-standards-enforcement/
- Montre comment des standards Engineering explicites deviennent des contrôles automatisés dans le workflow de développement.

[10] Issue tracking is dead — Linear

- URL : https://linear.app/next

- Défend un modèle où feedback, stratégie, décisions et code partagent davantage de contexte au-delà du ticket isolé.

[11] Quick: An internal hosting platform for the AI era — Shopify

- URL : https://shopify.engineering/quick
- Présente une plateforme interne permettant à davantage de collaborateurs de construire et publier directement.

[12] Stopping the slow death of internal tools — Vercel

- URL : https://vercel.com/blog/stopping-the-slow-death-of-internal-tools
- Montre comment des utilisateurs métier peuvent construire des outils internes sans détourner Engineering de sa roadmap principale.

[13] Empowered Product Teams — Marty Cagan, Silicon Valley Product Group (2018)

- URL : https://www.svpg.com/empowered-product-teams/
- Texte fondateur distinguant équipes responsables de résoudre des problèmes et équipes cantonnées à la livraison de fonctionnalités.

[14] EMPOWERED: Ordinary People, Extraordinary Products — Marty Cagan et Chris Jones (2020)

- URL : https://www.svpg.com/books/empowered-ordinary-people-extraordinary-products/
- Présentation du livre sur les équipes Produit autonomes, les objectifs confiés, le contexte stratégique et le rôle des responsables dans leur développement.

5 — Malorean Corp. — Ce qu'on peut faire, ce qu'on doit faire

Le mardi matin, Lucas avait commencé par poser une question qui avait surpris Alice.

— Est-ce que je peux parler au client ?

Il avait reçu la veille les quelques comptes rendus que le support conservait sur le problème de traçabilité des lots. Les messages parlaient de numéros de fabrication, de matières premières, de fichiers Excel et d'alertes fournisseurs. Il avait passé la soirée à les organiser avec l'aide de l'IA.

— Bien sûr, avait répondu Alice. Pourquoi ?

— Parce que je ne comprends pas exactement ce qui leur manque.

Alice avait souri. Une semaine plus tôt, elle se serait probablement chargée de ce rendez-vous avant de transmettre ses conclusions au développement. Elle avait maintenant promis de laisser Lucas mener le sujet.

— Je vais te présenter la personne du support qui suit ce compte. Elle pourra organiser l'échange.

Le mercredi, Lucas avait donc découvert une situation beaucoup moins simple que la phrase « gérer les lots ».

Le responsable qualité du client leur montra un courrier reçu d'un fournisseur. Une matière première devait être écartée. Le fournisseur connaissait son numéro de lot. Le fabricant savait

quelles commandes avaient été réceptionnées. Mais pour retrouver précisément les produits finis concernés, ses équipes devaient croiser plusieurs documents.

— La dernière fois, expliqua le responsable qualité, nous étions quatre à vérifier des fichiers pendant une bonne partie de la journée. Et même après, il a fallu retourner dans l'atelier pour être sûrs.

Lucas demanda à voir comment les informations étaient notées au moment de la fabrication. Le client lui montra un écran de l'ERP, puis une feuille de calcul.

— Mais pourquoi vous notez aussi ce numéro dans Excel ?

— Parce que votre logiciel ne nous permet pas de retrouver ce qu'on veut.

La réponse n'était pas agressive. Elle n'en était que plus difficile à entendre.

À la fin de l'appel, Lucas avait trois pages de notes et moins de certitudes qu'au début.

— Alors ? demanda Alice quand il la rappela.

— J'avais commencé à imaginer un écran pour enregistrer les numéros de lots.

— Et maintenant ?

— Maintenant je crois qu'il faut surtout comprendre comment ils enregistrent ce qu'ils ont réellement consommé. Sinon, on aura un bel écran et on ne saura toujours pas répondre à leur question.

Alice acquiesça. Elle aurait pu lui dire cela dès lundi. Mais elle n'était pas mécontente qu'il l'ait découvert lui-même.

Le vendredi suivant, Lucas lui montra une première version sur une base de démonstration.
On pouvait associer un lot de matière première à une fabrication, puis retrouver les produits finis qui en étaient issus. Il avait ajouté un petit exemple pour suivre un lot jusqu'aux produits concernés par une alerte.
Alice cliqua plusieurs fois.
— Tu as fait ça cette semaine ?
— Oui. Il y a encore beaucoup de choses fausses ou manquantes, mais ça nous donne quelque chose à montrer au client.
— Et si on ajoutait la possibilité de remonter aussi dans l'autre sens ?
— C'est déjà là.
Il ouvrit une autre vue.
Alice éclata de rire.
— Évidemment.
Depuis la démonstration du module de congés, elle avait changé d'échelle dans sa manière d'imaginer les projets. Ce qu'elle aurait autrefois renvoyé au trimestre suivant lui semblait maintenant à portée de main.
— On pourrait aussi permettre de rechercher par fournisseur. Et faire une alerte directement sur les commandes

concernées. Et pourquoi pas préparer les documents de rappel ?

Lucas ne répondit pas tout de suite.

— On peut noter ces idées, dit-il enfin. Mais je voudrais d'abord retourner voir le client avec ce qu'on a.

— On ne va quand même pas attendre quinze jours pour lui montrer une version plus complète ?

— Ce n'est pas la démonstration qui m'inquiète.

— Qu'est-ce qui t'inquiète ?

Lucas partagea cette fois un écran moins agréable à regarder. Des tables anciennes, des champs dont les noms n'évoquaient rien, plusieurs liens entre la fabrication et le stock.

— On peut créer tous les écrans qu'on veut. Le problème, c'est que notre ERP n'a jamais été conçu pour garder toute cette histoire des lots. Il y a des mouvements de stock vieux de plusieurs années, parfois corrigés à la main. Certaines informations n'existent tout simplement pas.

— L'IA peut nous aider à remettre tout ça en ordre, non ?

— Elle peut m'aider à écrire une migration. Elle ne peut pas inventer ce qu'on n'a jamais enregistré.

Alice resta silencieuse.

— Si on veut faire ça correctement, poursuivit Lucas, il faudra modifier une partie du modèle de données et vérifier ce que ça change dans les autres modules. Et on ne peut pas se permettre de fausser les stocks d'un client pour gagner deux semaines.

— Je ne te demande pas de casser le logiciel.

— Je sais. Mais depuis vendredi, à chaque fois que je te montre quelque chose, tu m'en demandes trois de plus.

La remarque était sortie sans agressivité. Elle n'en piqua pas moins Alice.

— C'est vrai, reconnut-elle.

— Moi aussi, ça me donne envie d'aller vite. Tu sais bien. Mais ce n'est pas parce que je peux générer du code en une heure que quinze ans de fonctionnement de l'ERP deviennent simples.

Il ajouta, après une hésitation :

— Thomas a regardé une partie du code du module de congés. Il a trouvé deux cas qu'on avait mal traités. Il avait raison. On les a corrigés, mais ça m'a rappelé une chose : on peut être beaucoup plus rapides à construire et toujours avoir besoin de quelqu'un qui connaît les pièges.

Alice se souvint de Thomas, le vendredi de la démonstration, demandant simplement à voir le code.

— Alors tu proposes quoi ?

— On continue la démonstration avec des données fictives. On vérifie qu'on a bien compris le besoin. Mais pour toucher aux vrais stocks, je veux qu'on prenne le temps de comprendre l'existant et de faire les contrôles nécessaires.

Alice regarda la première version encore ouverte sur son autre écran.

La semaine précédente, elle avait défendu l'autonomie de Lucas devant Hubert. Il lui semblait soudain moins agréable de découvrir que cette autonomie comprenait le droit de lui répondre non.

— D'accord, dit-elle finalement. On ne touche pas aux données réelles pour le moment.

Le lundi suivant, Hubert commença leur réunion hebdomadaire par une question désormais familière.

— Alors, notre expérience ?

— Elle avance, répondit Alice. Plus vite que prévu sur certains points. Moins vite sur d'autres.

— C'est-à-dire ?

Alice lui montra la démonstration de Lucas, puis résuma la discussion du vendredi.

Hubert s'attarda sur les réserves techniques.

— Il a raison de faire attention, dit-il. Je n'ai pas envie qu'on règle un problème de traçabilité en créant des problèmes de stock chez tous nos clients.

— Oui. Et je crois que je l'ai un peu poussé.

— Toi ?

— Depuis deux semaines, j'ai l'impression que tout est devenu possible. Dès qu'une idée apparaît, je me dis qu'on peut essayer tout de suite.

Hubert sourit.

— C'était plutôt mon état d'esprit après sa première démonstration.
— Le problème, c'est qu'on confond facilement la vitesse à laquelle on obtient quelque chose à l'écran et celle à laquelle on obtient quelque chose de fiable.
— Au moins, dit-il, Lucas sait quand il faut ralentir.
— Oui. Et j'ai besoin de lui faire confiance là-dessus.

Deux jours plus tard, Julie Morel demanda à Alice si elle avait cinq minutes.
Alice avait l'habitude de cette formule. Avec Julie, cinq minutes pouvaient facilement devenir une demi-heure.
— J'ai rencontré un prospect assez important la semaine dernière, commença Julie. Ils ont plusieurs usines et ils cherchent un ERP capable de s'adapter à leurs façons de travailler. Regarde ce que j'ai préparé.
Elle partagea une série de maquettes très propres. On y voyait des étapes de validation différentes selon les catégories de commandes, des règles propres à chaque établissement et des écrans qui changeaient suivant les utilisateurs.
— C'est toi qui as fait ça ? demanda Alice.
— Avec ChatGPT. J'ai repris mes notes du rendez-vous, je lui ai demandé de proposer un parcours, puis on a corrigé ensemble. J'y ai passé une soirée.
Alice reconnut immédiatement ce que ces images pouvaient apporter. En quelques heures, Julie avait donné une forme

concrète à un besoin jusque-là enfermé dans un compte rendu commercial.

— Et le client a vu les maquettes ?

— Pas encore. Je voulais te les montrer d'abord. Mais il m'a clairement dit que s'on savait faire ce genre de choses, ça pourrait changer la discussion.

Alice fit défiler les écrans.

— L'idée est intéressante. Tu devrais en parler directement à Lucas.

Julie la regarda, étonnée.

— Directement ? Je pensais que tu allais préparer le sujet.

— Pourquoi ? Tu connais le client, tu as échangé avec lui, tu as même préparé les maquettes. Tu peux lui expliquer ce qu'il cherche mieux que moi.

— Et Lucas peut regarder si c'est faisable ?

— Oui. Mais Julie, attention : regarder si c'est faisable ne veut pas dire qu'on s'engage à le développer.

— Bien sûr.

Alice connaissait ce « bien sûr ». Ce n'était pas un mensonge. C'était l'enthousiasme d'une commerciale devant une vente possible.

Le lendemain, elles se retrouvèrent avec Lucas. Julie déroula ses maquettes, expliqua les demandes du prospect et répondit aux questions.

Lucas s'intéressa tout de suite à la manière dont les règles différaient entre les établissements.

— On a déjà une partie de ce mécanisme dans l'ERP, dit-il. Pas pour faire exactement ça, mais il existe. On pourrait sans doute préparer quelque chose pour tester.

— En combien de temps ? demanda Julie.

— Pour une démonstration ? Pas forcément très longtemps.

Julie regarda Alice. Son sourire était presque celui de quelqu'un qui avait déjà remporté un marché.

— Si on lui montre ça la semaine prochaine, ça peut vraiment faire la différence.

Alice sentit sa propre envie d'avancer lui répondre avant d'avoir réfléchi. Elle pensa au chantier de traçabilité, aux demandes en attente, aux règles particulières de ce prospect et à tous les autres clients qui n'en avaient peut-être aucun usage.

— Non, dit-elle.

Les deux autres se tournèrent vers elle.

— Non à quoi ? demanda Julie.

— Non à l'idée de commencer à développer cette fonctionnalité dans notre produit pour décrocher ce contrat.

— Pourtant tu viens de dire que c'était intéressant.

— Ça l'est. Mais on ne sait pas si ce client a besoin d'une fonction que tous nos clients devraient retrouver, ou d'un fonctionnement très particulier à ses usines.

— On risque de perdre la vente.

— Peut-être.

Julie ne s'attendait manifestement pas à cette réponse.

— On a enfin les moyens de répondre vite à ce genre de demande, insista-t-elle. Ce serait dommage de s'en priver.

— Je ne veux pas nous en priver. Je veux qu'on comprenne ce qu'on propose. Si on ajoute chaque demande particulière au cœur de l'ERP, on va finir avec un logiciel impossible à faire évoluer.

— Mais Lucas dit qu'on peut le faire.

— Lucas dit qu'il peut préparer une démonstration. Ce n'est pas la même chose.

Un silence suivit. Julie regarda une dernière fois ses maquettes.

— Alors je lui dis quoi ?

— Qu'on a compris son problème et qu'on veut lui montrer une piste. Pas qu'on a décidé d'ajouter cette fonctionnalité au produit.

Julie hocha la tête. Elle paraissait déçue, mais pas découragée.

— D'accord. Je peux organiser un rendez-vous pour qu'il nous explique plus précisément les différences entre ses usines.

— Parfait. Et garde les maquettes. Elles vont nous aider à poser les bonnes questions.

Quand Julie quitta l'appel, Alice resta avec Lucas quelques secondes.

— Tu aurais vraiment pu construire ça vite ?

— Une première version, oui.

— Et tu aurais eu envie de le faire ?

— Franchement ? Oui. Ça avait l'air amusant.

Alice rit.

— Moi aussi, j'avais envie de dire oui.

— Qu'est-ce qui t'a retenue ?

— Le fait qu'on était tous les deux en train de parler de ce qu'on pouvait faire, et presque plus de ce qu'on devait faire.

La semaine suivante, Lucas envoya à Alice un lien accompagné d'un message.

J'ai bricolé un truc hier. Tu peux regarder ?

Alice ouvrit la page. Elle reconnut l'interface de leur ERP, mais avec un petit panneau latéral permettant de modifier certaines règles. Elle pouvait choisir un établissement, changer un seuil de validation, afficher ou masquer une étape, puis relancer la démonstration.

Elle rappela Lucas.

— C'est quoi ?

— Un petit outil pour préparer des démonstrations avec des règles différentes. J'ai réutilisé notre ancien mécanisme de configuration. C'est juste sur une instance de test.

— Tu as mis combien de temps ?

— Deux heures, peut-être. J'ai eu l'idée en repensant aux maquettes de Julie.

— Mais ce n'est même pas le sujet sur lequel tu travailles !

— Non. Je voulais voir si on pouvait éviter de modifier du code chaque fois qu'on veut tester une variante.

Alice commença à changer les paramètres. L'écran réagissait. Elle essaya une autre configuration, puis une troisième.

— Attends. Tu veux dire que je peux préparer moi-même plusieurs versions de démonstration ?

— Pour les règles qu'on a exposées, oui. Pas n'importe quoi. Et surtout pas sur les environnements des clients.

— Mais je peux montrer deux parcours différents à Julie et lui demander lequel correspond vraiment à son prospect.

— Exactement.

Alice resta quelques instants devant la page. Le premier module de congés avait rendu Lucas plus rapide. Cet outil lui donnait, à elle, une possibilité qu'elle n'avait pas la veille. Elle pouvait essayer une idée sans demander à un développeur de l'implémenter spécialement pour elle.

— Tu imagines ce qu'on pourrait faire avec ça ?

Lucas sourit.

— Je crois que oui. Mais avant que tu me demandes d'en faire un produit complet, je te rappelle que c'est un bricolage de deux heures.

— Promis.

Elle raccrocha, puis recommença à jouer avec les paramètres.

Le lundi suivant, Hubert remarqua qu'Alice n'avait pas ouvert son habituel document de suivi.

— Tu n'as rien préparé aujourd'hui ?

— Si. Mais je voudrais te montrer quelque chose.

Elle lui présenta l'outil de Lucas et fabriqua sous ses yeux deux variantes d'une même démonstration.

Hubert comprit assez vite.

— Tu n'as pas demandé aux développeurs de faire ces modifications ?

— Non. Lucas a créé un petit outil. Je peux tester certaines règles toute seule.

— On pourrait laisser les commerciaux s'en servir ?

— Peut-être plus tard. Pour le moment, ce n'est pas assez robuste. Mais Julie peut déjà travailler avec moi sur les cas de son prospect.

Hubert regarda l'écran avec intérêt.

— Et le projet des lots ?

— Lucas avance. Il est retourné voir le client avec sa première version. On comprend mieux ce qu'il faut faire. Mais il refuse de toucher aux données historiques avant d'avoir sécurisé la manière dont on les traite.

— Ça me paraît raisonnable.

— Oui. Vendredi, ça m'a agacée. Aujourd'hui, je suis contente qu'il ait insisté.

Alice fit tourner son stylo entre ses doigts.

— Il se passe quelque chose d'étrange, Hubert. Lucas discute avec les clients et prend des décisions sur la solution. Julie arrive avec des maquettes presque utilisables. Et moi, je peux maintenant tester certaines variantes de notre ERP sans passer par un développement.

— Tu trouves ça inquiétant ?

— Par moments. Je me demande ce que deviennent nos métiers si tout le monde commence à faire une partie du travail des autres.

— Et tu as trouvé une réponse ?

— Pas vraiment.

Elle réfléchit.

— Je sais seulement que je ne veux pas être la personne par laquelle chaque idée doit passer. Julie n'avait aucune raison d'attendre que je transforme ses maquettes en tickets. Lucas n'a pas besoin que je lui explique comment construire chaque écran.

— Mais tu as refusé qu'on développe la demande du prospect.

— Oui. Parce que c'était une décision sur notre produit. Je pouvais me tromper, je peux encore changer d'avis, mais c'était à moi d'assumer ce choix.

— Et Lucas a refusé de prendre un raccourci sur les stocks.

— Parce qu'il sait ce qu'on risque de casser. Là-dessus, il a un jugement que je dois écouter. Même si je suis pressée.

Hubert regarda un instant la petite fenêtre où s'affichait Alice.

— Donc on peut laisser les gens faire davantage de choses, mais on ne peut pas laisser toutes les décisions flotter entre eux.

— C'est ce que je commence à croire.

Elle hésita, puis ajouta :

— Avant, j'avais parfois l'impression que mon travail était important parce que beaucoup de sujets passaient par moi. Maintenant, je me demande si le plus important n'est pas ce qui arrive au bout. Est-ce qu'on résout le bon problème ? Est-ce qu'on construit quelque chose de fiable ? Est-ce que ça vaut la peine de le faire ?

Hubert acquiesça.

— Et si personne ne sait qui doit répondre à ces questions, on aura un problème.

— Oui.

Ils restèrent silencieux quelques secondes. Ni l'un ni l'autre n'avait envie de transformer cette conversation en annonce d'une nouvelle organisation.

Leur expérience venait à peine de commencer. Elle leur avait apporté plus de possibilités, mais aussi davantage de décisions à prendre.

Alice reçut une notification. Un message de Julie : le prospect acceptait de regarder les variantes de démonstration. Il voulait également savoir quand elles pourraient être disponibles pour ses équipes.

Alice ne répondit pas immédiatement.

Le logiciel permettait désormais de fabriquer des réponses beaucoup plus vite.

Mais quelqu'un devait encore décider lesquelles Malorean Corp. était prête à promettre — et à assumer.

6 — Les rôles se brouillent, les responsabilités restent

Pendant longtemps, les frontières entre Product, Design et Engineering ont servi de repères simples.
Le Product comprenait le problème et décidait quoi construire. Le Design concevait l'expérience. L'Engineering transformait cette intention en logiciel. Ce découpage n'a jamais été aussi propre dans la réalité, mais il structurait malgré tout les équipes, les recrutements, les processus et les responsabilités.
Cette répartition était déjà remise en question avant l'IA : les équipes Produit autonomes décrites au chapitre 1.4 faisaient travailler ensemble Produit, Design et Engineering sur un problème partagé. L'IA accélère désormais le franchissement d'une autre frontière : celle de l'exécution, lorsque chacun peut accomplir davantage de tâches autrefois réservées à un métier.
Le panorama des organisations a montré cette évolution, notamment à travers l'étude de Figma. [1] Sur Malorean, j'ai également décrit le mouvement croisé des développeurs vers la compréhension du problème et des PM vers sa réalisation. [2]
La tentation serait d'en déduire que les rôles vont disparaître. Je crois que le mouvement appelle plutôt à reformuler la

responsabilité de chacun : **plus les rôles deviennent poreux, plus les responsabilités doivent devenir explicites.**

La capacité d'agir n'appartient plus à un seul métier

Ce que l'IA redistribue en premier, ce n'est pas nécessairement le pouvoir de décision. C'est la capacité d'action.

Prenons le cas de designers qui modifient eux-mêmes le code, une pratique documentée par Atlassian : une équipe technique continue d'examiner la qualité et les conditions de mise en production. [3] Cette étape n'est pas un retour à l'ancien découpage ; elle montre que la capacité d'intervenir dans le code et la responsabilité de sa fiabilité ne se confondent pas.

Même distinction dans l'offre d'emploi de Linear pour un Product Engineer : l'entreprise attend de ce poste la capacité de porter un sujet, du problème initial au résultat livré. [4] Cela ne lui confère pas automatiquement l'autorité sur toutes les décisions de stratégie Produit.

On peut donc réduire des passages de relais sans supprimer l'expertise qui justifiait certains d'entre eux. La question n'est plus de compter les métiers qui interviennent, mais de savoir **qui répond de quoi lorsque plusieurs métiers interviennent sur le même terrain**.

Pouvoir contribuer ne signifie pas être responsable du domaine

Il faut distinguer deux choses que les organisations confondent souvent : la permission d'agir et la responsabilité.

Un Sales peut construire un outil interne. Cela ne fait pas de lui le responsable de la sécurité de la plateforme.

Un PM peut modifier une interface. Cela ne signifie pas qu'il devient responsable de la cohérence du Design System.

Un développeur peut décider d'un comportement Produit local. Cela ne lui donne pas automatiquement la responsabilité de la stratégie, du positionnement ou de la roadmap globale.

Cette distinction existe déjà dans beaucoup d'équipes, mais l'IA oblige à la formaliser.

Dans un fonctionnement très spécialisé, le métier et la responsabilité se superposaient souvent. Celui qui pouvait agir était aussi celui dont c'était le domaine. Dès que la capacité d'action se diffuse, cette superposition disparaît.

Le risque est alors de confondre autonomie et dilution de la responsabilité.

J'ai déjà défendu un principe proche à propos de la qualité : les développeurs ne peuvent pas livrer puis considérer que les défauts appartiennent à une équipe QA, à une rotation bugs ou au prochain développeur disponible. La qualité technique de ce qui est livré reste une responsabilité identifiable. [5]

La porosité des rôles ne change pas ce principe. Elle le rend plus important.

Le concept de Garant

C'est pour répondre à ce problème que j'utilise le concept de **Garant**.

Pour chaque domaine important, une personne ou une équipe reste garante de sa cohérence et de sa qualité.

Le Garant n'est pas nécessairement celui qui réalise tout le travail du domaine.

Il n'est pas non plus celui qui doit approuver chaque micro-décision.

Il est celui vers qui l'organisation se retourne lorsqu'un choix important touche ce domaine, lorsqu'une limite est franchie ou lorsque quelque chose se passe mal.

Dans une organisation simple, on peut imaginer au moins trois garanties explicites :

- **Garant Produit** : cohérence Produit, choix structurants, cible, proposition de valeur, arbitrages majeurs, relation avec la stratégie.
- **Garant Design** : Design System, cohérence d'expérience, accessibilité, nouveaux patterns et niveau d'exigence UX.
- **Garant Dev** : architecture, sécurité, performance, maintenabilité, qualité technique et conditions de production.

Le principe est simple :

tout le monde peut contribuer à plusieurs domaines, mais chaque domaine conserve quelqu'un qui en répond.
Ce modèle autorise la porosité sans faire disparaître les responsabilités.

Un Sales peut construire — sans devenir garant technique

Prenons un exemple volontairement simple.
Un Sales identifie un besoin, utilise un outil assisté par IA et produit une petite application ou une modification utile. Tant que l'entreprise dispose d'un environnement suffisamment sécurisé, il n'y a aucune raison de considérer que seul Engineering a le droit matériel de construire.
Mais cette liberté ne signifie pas que la qualité technique devient la responsabilité du Sales.
Si le changement touche la sécurité, les permissions, des données sensibles ou l'architecture, le Garant Dev doit pouvoir demander une relecture, imposer une correction ou bloquer le changement.
Le Sales reste responsable de son besoin.
Engineering reste garant de ce que l'entreprise accepte de faire vivre techniquement.
La même logique vaut pour le Design.
Un PM peut assembler un écran à partir de composants connus. Tant qu'il reste dans le Design System et dans des patterns maîtrisés, demander systématiquement une validation Design réduirait inutilement l'autonomie.

En revanche, si ce PM sort du Design System, invente un nouveau pattern ou conçoit une interaction structurante et complexe, le Garant Design doit pouvoir intervenir.
La frontière n'est donc pas : « qui a le droit de toucher à quoi ? »
Elle devient : **« à partir de quel niveau de risque ou de nouveauté la garantie d'un domaine doit-elle reprendre la main ? »**

L'owner du sujet et le Garant du domaine ne sont pas la même personne

Cette distinction est probablement l'une des plus importantes des nouvelles organisations.
L'**owner** porte un sujet.
Il doit comprendre le problème, faire avancer le travail, arbitrer les décisions locales, coordonner ce qui doit l'être, vérifier le résultat et assumer que le sujet aboutisse.
Le **Garant**, lui, porte une responsabilité plus transverse.
Il ne possède pas nécessairement le sujet. Il possède un niveau de cohérence, de qualité ou de risque sur un domaine.
Un Product Engineer peut donc être owner d'une fonctionnalité sans devenir garant de toute l'architecture de l'entreprise.
Un PM peut être owner d'un lancement sans devenir garant de tout le Design.
Un Sales peut être owner d'un outil métier sans devenir garant de la sécurité des applications internes.

Cette distinction rend la responsabilité du résultat plus claire, sans exiger d'une seule personne une expertise dans tous les domaines.

AMPECO illustre une version de ce mouvement : un Product Owner et un engineer alignent le « quoi » et le « pourquoi », puis les agents prennent en charge une grande partie de l'analyse, du développement et des tests. L'exécution se distribue, mais un humain reste responsable de la validation finale. [6]

L'IA peut donc réduire le nombre de personnes nécessaires pour exécuter un sujet tout en augmentant la nécessité de savoir très clairement **qui en répond**.

Le rôle n'est peut-être plus la bonne unité de lecture

Cette évolution pose aussi une question plus large : faut-il encore organiser l'entreprise principalement par titres de poste ?

Le débat autour du Product Owner est révélateur. Sur Malorean, j'ai défendu l'idée que le PO est d'abord une fonction — un mandat de décision et de responsabilité — plutôt qu'un métier qui trouverait sa justification dans la tenue d'un backlog. [7]

Cette distinction devient particulièrement utile avec l'IA.

Si l'on définit un rôle par la liste des tâches qu'il accomplit, l'IA rend cette définition instable. Beaucoup de tâches deviennent accessibles à davantage de profils.

Si l'on définit une responsabilité par ce dont une personne ou une équipe doit répondre, elle résiste beaucoup mieux.
Le PM peut écrire moins de specs sans cesser d'être responsable de la qualité des choix Produit.
Le développeur peut écrire moins de code manuellement sans cesser d'être responsable de la qualité technique.
Le designer peut ne plus être l'unique personne à déplacer des pixels sans cesser d'être garant de la cohérence de l'expérience.
Chez Cursor, où les frontières entre Produit, Design et développement sont également moins rigides, le responsable Produit rappelle que l'essentiel demeure : choisir ce qui mérite d'être construit et donner suffisamment de direction pour éviter de s'en écarter. [8]
Le métier change donc moins parce qu'il perd toutes ses responsabilités que parce qu'il perd le monopole de certaines formes d'exécution.

7

Le Garant ne doit pas devenir une nouvelle bureaucratie

Il y a toutefois un piège évident.
Si chaque intervention d'un PM dans le code doit être validée par Engineering, si chaque décision locale d'un développeur doit repasser par le PM, et si chaque écran produit par quelqu'un d'autre qu'un designer doit attendre un comité Design, alors nous n'avons rien résolu.
Nous avons simplement recréé les anciens passages obligés sous un nouveau nom.

Le Garant ne doit pas devenir une validation systématique. Son existence sert précisément à permettre davantage d'autonomie : chacun peut agir à l'intérieur d'un cadre connu, sans demander une autorisation permanente, parce que l'organisation sait à la fois où se trouvent les responsabilités et dans quels cas elles doivent être réengagées.

Dans la vraie vie, ce modèle ne fonctionne d'ailleurs jamais parfaitement.

Des choses arrivent en production que le Garant n'aurait pas validées exactement de cette façon. Des décisions locales passent sous le radar. Des équipes prennent parfois plus de liberté que prévu. À l'inverse, certaines personnes escaladent trop tôt par prudence.

Ce n'est pas nécessairement un échec du modèle.

Une organisation réellement autonome accepte qu'une partie des décisions soit prise sans inspection préalable. Sinon, ce n'est pas de l'autonomie.

Le problème à résoudre n'est donc pas de garantir qu'aucun écart n'arrive.

Le problème est de définir **quels écarts sont acceptables, lesquels doivent être détectés par le système, et lesquels exigent l'intervention d'un Garant avant de devenir coûteux**.

C'est là que la question change.

Nous avons distingué deux responsabilités : celle de la personne qui porte un sujet et celle du Garant d'un domaine.

Reste à résoudre une autre question :

quand faut-il laisser l'owner décider seul, et quand faut-il faire intervenir le Garant ?

Chez Malorean Corp., cette frontière entre initiative et responsabilité commence à provoquer des frictions. Nous les retrouverons dans le prochain épisode, avant d'examiner les garde-fous et les règles d'escalade.

Bibliographie

[1] Are Roles and Responsibilities a Thing of the Past? — Figma

- URL : https://www.figma.com/blog/2025-shifting-roles-report/
- Étudie l'élargissement des responsabilités et la porosité croissante des métiers au sein des équipes modernes de développement produit.

[2] PM, développeurs et IA : les rôles se brouillent, les responsabilités restent — Malorean

- URL : https://malorean.net/articles/2026-08-01-pm-developpeurs-et-ia-les-roles-se-brouillent-les-responsabilites-restent.html
- Développe le constat auteur : l'IA élargit qui peut agir, sans supprimer les responsabilités Produit, Design et Engineering.

[3] Designers' workflow for shipping code — Atlassian

- URL : https://www.atlassian.com/blog/ai-at-work/designers-workflow-for-shipping-code

- Décrit des designers modifiant directement le code produit, avec une validation Engineering maintenue pour qualité, performance et production.

[4] Product Engineer — Linear Careers

- URL : https://linear.app/careers/0c7c2e26-0a98-42cf-a47c-9a3999fb513b
- Offre d'emploi précisant les attentes de Linear pour un Product Engineer, de la définition du problème à la livraison de fonctionnalités.

[5] La qualité appartient à ceux qui livrent — Malorean

- URL : https://malorean.net/articles/2026-06-03-la-qualite-appartient-a-ceux-qui-livrent.html
- Défend une responsabilité technique explicite : la qualité ne se délègue pas après livraison à une équipe chargée de réparer.

[6] How we built an AI-native engineering system — AMPECO

- URL : https://www.ampeco.com/blog/how-we-built-an-ai-native-engineering-system/
- Montre une exécution largement déléguée aux agents tout en maintenant l'intention, l'architecture et la validation sous responsabilité humaine.

[7] Le PO n'est pas un métier, c'est une fonction — et l'IA n'y change rien, elle accélère — Malorean

- URL : https://malorean.net/articles/2026-08-03-le-po-nest-pas-un-metier-cest-une-fonction.html

- Distingue un mandat de décision et d'ownership d'un métier défini principalement par la gestion d'artefacts ou de backlog.

[8] How Cursor Is Navigating the New Shape of Product Craft — Atlassian / Cursor

- URL : https://www.atlassian.com/blog/how-we-build/how-cursor-is-navigating-the-new-shape-of-product-craft
- Montre des frontières de rôle plus poreuses tout en maintenant le jugement Produit, les objectifs et les garde-fous comme responsabilités centrales.

7 — Malorean Corp. — Ce qu'Alice savait sans l'avoir écrit

Le jeudi matin, Lucas appela Alice pour la troisième fois de la semaine.

— J'en ai pour deux minutes, promit-il.

Alice avait appris à se méfier de cette phrase. Lucas ne lui demandait jamais de longues réunions. Il arrivait avec une question précise, ils en parlaient, il repartait travailler. C'était justement ce qui rendait ces interruptions difficiles à refuser.

— Vas-y.

— Sur la traçabilité des lots, le client voudrait pouvoir chercher par fournisseur. Ça, je peux l'ajouter à la démonstration. Mais si le fournisseur a livré plusieurs lots avec le même numéro interne, on affiche tout ensemble ou on lui demande de choisir ?

Alice réfléchit quelques secondes.

— Il ne faut surtout pas laisser croire qu'on a identifié un lot si plusieurs correspondent. Affiche les résultats distinctement. On pourra regarder avec le client comment il les différencie.

— C'est ce que je pensais. Mais je n'étais pas sûr.

— Alors pourquoi tu m'appelles ?

Lucas eut un rire gêné.

— Parce qu'après l'histoire des maquettes de Julie, je préfère vérifier avant de décider quelque chose qui ressemble à une règle Produit.

La semaine précédente, Alice lui avait demandé de ne pas transformer la demande d'un prospect en fonctionnalité standard. Elle trouvait maintenant dommage qu'il ait retenu de cette discussion qu'il valait mieux lui demander son avis au moindre doute.

— D'accord. Mais tu n'avais pas besoin de moi pour celle-là.

— Ça me rassure de te l'entendre dire.

Ils raccrochèrent. L'appel avait duré sept minutes. Alice revint au document qu'elle écrivait, relut sa dernière phrase et ne se souvint plus de ce qu'elle voulait ajouter.

Dix minutes plus tard, Lucas lui envoya un message.

Autre question : pour les fabrications anciennes où l'information sur les lots n'existe pas, on met « inconnu » ou on cache la colonne ?

Elle commença à taper une réponse. Puis elle s'arrêta.

La question était différente, mais le raisonnement était le même : ne jamais présenter comme certaine une information que l'ERP ne possédait pas. Elle aurait pu lui donner cette règle au lieu de trancher deux cas séparément.

Elle répondit malgré tout.

On affiche clairement que l'information manque. Ne pas masquer le problème.

Lucas réagit avec un pouce levé.

Alice regarda son agenda. Une discussion avec le support avait été décalée. Un rendez-vous client l'attendait l'après-midi. Entre les deux, elle devait relire les propositions de Julie pour son prospect multi-sites.

Elle aimait que Lucas lui parle directement. C'était précisément ce qu'elle avait voulu en lui confiant le sujet. Mais il avançait si vite qu'à chaque nouvelle possibilité apparaissait une nouvelle question. Et plus il la consultait, moins elle avait le temps de faire ce qu'elle était la mieux placée pour faire : comprendre les clients et réfléchir à la direction du produit.

Lui aussi perdait du temps. Parfois, il attendait sa réponse avant de reprendre son travail.

Une pensée désagréable lui traversa l'esprit : si cela continuait, ils allaient peut-être finir par remettre toutes les décisions dans des tickets. Ce serait plus facile à organiser. Elle spécifierait, Lucas exécuterait, et personne n'aurait plus à se demander jusqu'où allait son autonomie.

Exactement ce qu'ils avaient essayé de dépasser.

À midi, Alice retrouva Bob en visioconférence. Ils avaient pris l'habitude de se parler sans ordre du jour, simplement pour confronter ce qui leur arrivait.

— Comment ça se passe avec Lucas ? demanda Bob.

— Trop bien, d'une certaine manière.

— Ça n'a pas l'air d'être un compliment.

— Il avance très vite, il voit plein de choses, il parle aux clients. Mais il m'appelle tout le temps pour des décisions qu'il pourrait prendre seul.

— Et tu lui réponds ?

— Évidemment.

— Alors il a raison de t'appeler.

Alice leva les yeux au ciel.

— Merci, Bob.

— Je ne plaisante qu'à moitié. Si je peux avoir ta réponse en cinq minutes plutôt que prendre le risque de me tromper, pourquoi je m'en priverais ?

— Parce qu'on ne va jamais réussir à travailler comme ça.

Bob lui demanda un exemple. Elle lui raconta les deux questions du matin, puis la discussion avec Julie, puis une hésitation récente de Lucas : fallait-il ajouter un réglage demandé par un client si ce réglage ne servirait probablement qu'à lui ?

— Tu te rends compte ? dit Alice. Ce sont toujours des variantes de la même question.

— Et quelle est ta réponse ?

— Ça dépend.

Bob sourit.

— Voilà. C'est exactement le genre de réponse qui oblige les gens à te rappeler.

Alice allait protester. Mais il avait raison.

— Ce n’est pas « ça dépend » au hasard, reprit-elle. Je regarde si on répond à un problème récurrent, si ça sert nos clients prioritaires, si on va alourdir le produit pour tout le monde, si on pourra maintenir la solution…

Elle s’interrompit.

— Tu vois ? dit Bob. Tu viens de me donner une partie de la règle.

— Ce n’est pas une règle. C’est plutôt une façon de réfléchir.

— Encore mieux. C’est peut-être ça qu’il faudrait transmettre à Lucas.

Alice resta silencieuse.

Depuis des années, elle rédigeait des tickets. Elle y décrivait des comportements attendus, des cas particuliers, des critères d’acceptation. Mais elle ne rédigeait presque jamais ce qui lui permettait de choisir entre plusieurs comportements possibles. Ce raisonnement restait dans sa tête, ou disparaissait dans une réunion une fois le sujet livré.

— J’ai probablement des centaines de choses comme ça, dit-elle.

— Sans doute. Moi aussi.

— Et on n’a jamais eu de raison de les écrire.

— On avait des raisons. Mais on n’en avait pas le temps. Et personne n’ouvrait un ticket intitulé « expliquer pourquoi on refuse parfois une bonne idée ».

Ils rirent.

Après l'appel, Alice ouvrit un nouveau document. Elle écrivit : *Principes de décision Produit*.
Elle ajouta une première phrase, la supprima, en essaya une autre. Au bout de quinze minutes, elle avait produit un paragraphe qui ressemblait à une politique interne. Il était correct. Il ne ressemblait pas du tout à ce qu'elle avait expliqué à Bob.
Elle referma le document.
Puis elle ouvrit ChatGPT.

— Aide-moi à sortir de ma tête ce que je sais de notre produit, dicta-t-elle. Je ne veux pas une procédure théorique. Pose-moi des questions, aide-moi à retrouver des exemples et à distinguer ce qui est une vraie règle de ce qui est simplement mon habitude.
La première réponse lui proposa de commencer par les décisions récentes.
Alice parla.
Elle parla du prospect de Julie, des maquettes séduisantes et du risque de transformer un besoin particulier en promesse pour tous les clients. Elle expliqua qu'un grand compte pouvait représenter beaucoup d'argent sans devenir automatiquement la bonne direction pour le produit.
— Attends, il faut que je t'explique pourquoi, ajouta-t-elle à voix haute. Notre ERP sert des entreprises très différentes. Si on ajoute chaque règle particulière au cœur du logiciel, on

finit par compliquer la vie de tout le monde. Mais parfois, une demande particulière révèle un problème beaucoup plus général. Il ne faut pas non plus tout refuser.

ChatGPT lui demanda comment elle faisait la différence.

Alice décrivit les échanges avec le support, les cas observés chez plusieurs clients, les demandes des commerciaux, les contraintes du déploiement. Elle retrouva des discussions anciennes dont elle n'avait gardé aucun compte rendu. Elle expliqua pourquoi certaines options avaient été abandonnées, pourquoi un réglage avait été préféré à une nouvelle fonctionnalité, pourquoi une exception de paie ne se traitait pas comme une préférence d'interface.

Par moments, elle se surprenait à répondre à une question qu'elle n'avait jamais pensé poser explicitement.

L'IA reformulait, regroupait, proposait un principe. Alice le corrigeait.

— Non, ce n'est pas « nous ne développons jamais de spécifique ». On peut le faire dans certaines conditions, mais on doit décider consciemment ce que ça nous coûte. Et surtout ne pas faire passer une hypothèse commerciale pour une décision Produit.

Un peu plus tard, ChatGPT avait écrit qu'une fonctionnalité de traçabilité pouvait garantir la reconstitution de tous les lots historiques.

— Certainement pas, dit Alice devant son écran. Si les données n'ont jamais été enregistrées, on ne peut pas les

reconstituer par magie. Ça, c'est une limite à afficher, pas une capacité à promettre.

Elle fit corriger le passage. Elle demanda que le document distingue ce qui avait été décidé, ce qu'elle pensait probable et ce qu'il faudrait vérifier auprès du support, de Lucas ou des clients. Elle évita d'y coller des informations confidentielles issues des comptes réels ; les exemples étaient reformulés.

Une heure passa. Puis deux.

Le document se remplissait de choses qu'Alice connaissait depuis longtemps : les clients que Malorean voulait servir en priorité, les problèmes qui revenaient, les compromis acceptés, les engagements à ne pas prendre, les limites des données anciennes, les raisons pour lesquelles un prototype n'était pas une fonctionnalité prête à vendre.

Elle ajouta les liens vers les informations déjà disponibles : retours du support, décisions encore retrouvables, notes de rendez-vous, documentation du produit. Là où il manquait une preuve, le texte devait le dire.

Vers la fin de la troisième heure, ChatGPT lui proposa une synthèse.

Alice la relut. Certaines pages étaient trop longues. D'autres mélangeaient une règle générale et un exemple. Elle corrigea une dernière fois les formulations les plus catégoriques, ajouta une date et le périmètre du document.

Puis elle exporta le tout en PDF.

Elle fit défiler les pages, étonnée par leur nombre.

Trois heures auparavant, elle pensait surtout expliquer à Lucas quand il pouvait prendre une décision sans elle. Elle tenait maintenant entre les mains un portrait imparfait, mais étonnamment riche, de ce que l'équipe savait du produit sans l'avoir jamais mis en commun.

Ce n'était pas une vérité définitive. Certains passages demanderaient d'être confirmés ou actualisés. Le PDF ne remplacerait ni les clients, ni les données, ni le jugement des personnes responsables.

Mais au moins, la prochaine personne n'aurait pas à repartir de zéro.

Lucas ouvrit le fichier le lendemain matin.

— Tu as écrit tout ça pour moi ?

— Pas toute seule, répondit Alice. J'ai passé trois heures à parler à ChatGPT. Il m'a aidée à poser les questions, à organiser mes réponses et à repérer ce qui manquait. J'ai relu et corrigé. Mais la plupart de ces choses, je les savais déjà.

— Alors pourquoi je ne les avais jamais vues ?

Alice sourit, un peu embarrassée.

— Parce qu'elles n'entraient dans aucun ticket.

Lucas parcourut le sommaire. Il y trouva des principes, des exemples de décisions, des contraintes métier, des informations sur les clients, des liens vers les sources, des questions encore ouvertes.

— Je peux donner ça à mon IA quand je travaille sur la traçabilité ?

— Oui, dans notre environnement autorisé. Mais dis-lui de s'en servir comme contexte, pas comme preuve que tout est vrai. Et si une règle est importante, on retourne à la source ou à la personne qui en répond.

— Ça me ferait gagner un temps fou. Parfois, je demande à l'IA de m'aider à choisir entre deux solutions, mais elle ne connaît que ce que je lui ai raconté. Elle peut me proposer une réponse très convaincante qui ne tient pas compte d'un choix fait il y a trois ans.

Alice connaissait cette impression. Une réponse raisonnable prise isolément pouvait être absurde pour leur ERP.

Lucas s'arrêta sur un passage.

— Là, tu dis qu'il ne faut jamais masquer une information métier manquante si cela peut faire croire que le résultat est complet.

— C'est ce que je t'ai dit hier.

— Oui. Et si j'avais eu cette phrase hier, je ne t'aurais pas appelé.

Il continua sa lecture. Puis il releva la tête.

— D'accord, mais j'ai encore un problème.

Alice eut un mouvement de découragement.

— Lequel ?

— Maintenant je comprends mieux comment tu raisonnes. Je ne sais toujours pas à partir de quand j'ai le droit de décider tout seul.

Elle posa son café.

Il venait de trouver le trou le plus important du document.

Ils reprirent le sujet ensemble. Alice partagea une page blanche.

— Essayons quelque chose de simple. Avant de m'appeler, tu te poses trois questions : est-ce que je comprends le résultat recherché ? Est-ce que ma décision reste dans le cadre qu'on connaît ? Et qu'est-ce qui pourrait mal se passer si je me trompe ?

— Il va falloir définir « dans le cadre ».

— Justement.

Ils écrivirent trois niveaux.

Vert : Lucas décidait seul lorsque le choix restait local, réversible, conforme aux règles connues et sans promesse nouvelle au client. Il pouvait expliquer son raisonnement et vérifier le résultat.

Orange : il consultait Alice lorsqu'il rencontrait une situation nouvelle, une contradiction entre objectifs ou un doute significatif sur le comportement Produit. Pour un risque technique, il sollicitait aussi les personnes compétentes, comme Thomas lorsqu'il s'agissait de l'historique de l'ERP.

Rouge : personne ne décidait seul lorsqu'il s'agissait de modifier une règle structurante, d'engager Malorean commercialement, de toucher à des données sensibles ou de prendre un risque majeur et difficilement réversible. Il fallait un arbitrage explicite avec les responsables concernés, et parfois Hubert.

Lucas relut la grille.

— Donc si je change la façon de présenter les résultats dans la démo pour éviter une confusion, je suis dans le vert.

— Oui.

— Si je veux ajouter un comportement qui pourrait concerner tous nos clients, je viens te voir.

— On en discute, oui. Surtout si on n'a pas compris le besoin.

— Et si je veux modifier les données historiques pour enregistrer les lots ?

— Ce n'est pas à moi toute seule de te donner le feu vert. Il faut l'analyse technique et les contrôles appropriés. Tu me l'as assez rappelé.

Lucas sourit.

— Je voulais être sûr que tu t'en souvenais.

Ils ajoutèrent une règle : l'absence de réponse d'Alice ne transformerait jamais une décision orange ou rouge en décision verte. En revanche, il ne fallait pas non plus lui demander une autorisation pour chaque décision verte.

— Et si je me trompe de couleur ? demanda Lucas.

— Tu me le dis. On ajuste le cadre. Je préfère qu'on améliore la règle plutôt qu'on passe notre temps à te reprocher de ne pas avoir deviné ce que je pensais.

Lucas hocha la tête.

— Je pourrais même demander à mon IA de confronter mes choix à cette grille, avec les raisons et les informations manquantes.

— Oui. À condition qu'elle ne décide pas à notre place si un risque est acceptable.

Alice ajouta la grille au document de référence. Elle commençait à comprendre la différence entre transmettre des informations et donner réellement les moyens de décider.

Ils auraient pu s'arrêter là. Mais Lucas revint sur la démonstration de traçabilité.

— On parle de mon autonomie depuis une heure. Je vais te poser une question sur le sujet lui-même : qu'est-ce qui nous dira qu'on l'a réussi ?

Alice répondit presque automatiquement :

— Si le client peut retrouver les produits fabriqués avec un lot donné.

— Oui, mais comment on le vérifie ? Et à quel point ça doit être fiable ?

Cette fois, elle ne chercha pas immédiatement une réponse. Ils avaient passé des jours à explorer des écrans, des règles et des structures de données. Ils avaient une démonstration

prometteuse. Pourtant, personne n'avait encore écrit précisément comment juger le résultat.

Alice se souvint du responsable qualité qui avait passé une bonne partie de la journée à croiser des fichiers après une alerte fournisseur.

— On devrait repartir de là, dit-elle. Avec un numéro de lot signalé, est-ce qu'il peut retrouver les fabrications concernées sans refaire toutes ses recherches à la main ? Combien de temps lui faut-il ? Est-ce que le résultat est complet ? Et quand on ne possède pas l'information, est-ce qu'on l'affiche clairement ?

— On peut le tester avec lui. Pas seulement lui demander si notre démo lui plaît.

— Exactement.

Alice ouvrit une nouvelle page.

— En fait, on aurait dû commencer par ça avant de se lancer.

Lucas éclata de rire.

— Tu me l'aurais demandé il y a deux semaines, je t'aurais dit que tu essayais de me refaire un ticket.

— Peut-être. Mais regarde ce qu'on écrit.

Ils inscrivirent le problème observé, les utilisateurs concernés, le résultat attendu, la manière de le mesurer, les limites des données, ce qui restait hors périmètre et les décisions à faire valider. Une page, pas davantage.

Lucas proposa de l'appeler *Mini PRD*.

— On n’est pas en train de me dire quel écran construire, remarqua-t-il. On est en train de se mettre d’accord sur ce qui rendra le travail utile.

— Et sur les limites à ne pas franchir.

Alice réfléchit à voix haute :

— Pour les prochains sujets, ça devrait être notre point de départ. Avant de demander ce qu’on va développer, on se demande quel résultat on veut obtenir et comment on le vérifiera. Ensuite seulement, on cadre et on décide de ce qui mérite d’être construit.

— Et si je sais remplir ce document avec les bonnes informations, je peux commencer sans attendre que tu me prépares un backlog.

— Tu peux proposer le cadrage. Ça ne veut pas dire qu’on valide n’importe quel engagement sans le regarder. Mais oui, c’est le but.

Ils complétèrent le Mini PRD de la traçabilité en laissant ouvertes les questions qu’ils ne savaient pas encore trancher. Il restait à vérifier certains cas avec le client et à confirmer des contraintes techniques.

Le document ne donnait aucune illusion de certitude. Il disait ce qu’ils cherchaient à apprendre.

Alice le trouva plus utile qu’une liste de fonctionnalités à cocher.

Le lundi suivant, le rendez-vous hebdomadaire d'Alice avec Hubert fut différent.

Alice ne partagea ni son tableau de suivi habituel ni la liste des fonctionnalités produites par Lucas.

Elle montra la grille de décision, puis le Mini PRD.

Hubert lut les premières lignes.

— Vous avez passé autant de temps à faire des documents alors que Lucas va maintenant beaucoup plus vite ?

Alice connaissait l'objection. Elle l'avait formulée elle-même quelques jours plus tôt.

— Je pense justement qu'on n'en avait pas assez. On écrivait ce qu'il fallait construire, mais rarement les informations qui permettaient de choisir. Résultat : Lucas me demandait constamment de trancher. Avec ça, il peut prendre une partie des décisions sans moi.

— Et comment tu sais qu'il reste dans les limites ?

— On a défini quand il peut décider seul et quand il doit nous consulter. Pour le reste, on ne va pas tout inspecter. On vérifiera les résultats et on corrigera le cadre si nécessaire.

Hubert s'arrêta sur la ligne concernant les données anciennes.

— Vous ne pourrez pas garantir la traçabilité de ce qui n'a jamais été enregistré.

— Non, répondit Alice. Et le document le dit. On doit le montrer au client clairement.

Hubert approuva.

— Ça, je préfère le savoir avant qu'un commercial promette le contraire.

— Justement, dit Alice. Le contexte ne servira pas qu'à Lucas. Julie pourrait aussi y trouver ce qu'on sait et ce qu'on n'a pas décidé. Mais il faudra que les informations restent à jour et que chacun sache quelles parties font autorité.

Ils convinrent de conserver une version de travail modifiable, dont le PDF serait une photographie datée. Les décisions importantes seraient tracées avec leur raison ; les principes récurrents pourraient, quand c'était pertinent, devenir des instructions pour les agents ou des contrôles dans les outils. Cela n'enlèverait rien aux tests techniques ni aux validations nécessaires avant une mise en production.

— Je n'ai pas envie de découvrir dans six mois que l'IA applique une vieille règle que plus personne ne défend, dit Hubert.

— Moi non plus, répondit Alice.

Pour la première fois, Hubert sembla surtout intéressé par le Mini PRD.

— Et vous allez regarder si le client arrive réellement à retrouver ses lots ?

— Oui, répondit Alice. On va lui donner des cas à tester avec nous. Si le résultat n'est pas fiable, le nombre d'écrans terminés n'aura aucune importance.

Hubert referma le document.

— Très bien. Je veux savoir ce que vous apprenez, pas seulement ce que vous livrez.

Le mercredi, Alice remarqua quelque chose d'étrange dans sa messagerie.
Aucun message de Lucas depuis le matin.
Elle vérifia le canal du projet. Il y avait pourtant de l'activité : Lucas avait testé plusieurs manières d'afficher les résultats sur l'instance de démonstration. Pour l'une d'elles, il avait expliqué qu'il avait choisi de séparer les lots portant des références identiques afin de ne pas suggérer un lien inexistant. Il citait le principe du contexte Produit, indiquait les données utilisées et proposait de vérifier la compréhension de l'écran avec le client.
Il n'avait pas demandé la permission.
Alice lut ses explications, puis retourna travailler. Elle n'avait rien à ajouter.
En fin d'après-midi, Lucas l'appela.
— J'ai un orange qui vire peut-être au rouge, dit-il.
— Je t'écoute.
— Le client aimerait pouvoir rechercher aussi dans ses anciennes fabrications. Pour une partie de l'historique, on n'a pas de lien fiable entre les matières premières et les produits finis. Je peux afficher ce qu'on sait, mais si on leur présente une recherche complète, ils risquent de croire que l'absence de résultat signifie qu'aucun produit n'est concerné.

Alice se redressa.

— Tu as raison. On ne peut pas laisser cette ambiguïté.

— Je pourrais ajouter un avertissement et distinguer les périodes où les données sont exploitables. Mais je voudrais qu'on décide ensemble de ce qu'on pourra promettre au client.

— Là, oui. Et il faudra aussi voir avec le support ce qu'il comprendrait de cette limite.

Lucas partagea un écran. Ils commencèrent à travailler sur le problème. Ce n'était pas une validation de bouton ni une réponse qu'Alice connaissait déjà. Ils avaient devant eux une décision susceptible de changer la confiance qu'un client accorderait au logiciel.

Pendant l'échange, Alice remarqua qu'elle n'avait pas pensé une seule fois au temps qu'il lui prenait.

Avant de raccrocher, Lucas hésita.

— Tu sais, ton PDF n'a pas répondu à toutes mes questions.

— Heureusement, dit Alice.

— Heureusement ?

— Si j'avais réussi à y mettre toutes les réponses, ça voudrait dire que tu n'aurais plus rien à découvrir.

Lucas sourit.

— D'accord. Mais je crois que je comprends mieux quand venir te voir.

Après l'appel, Alice rouvrit son agenda. Elle disposait d'une heure libre avant son prochain rendez-vous client.

Une heure entière.

Elle ne savait pas encore si leur expérience fonctionnerait à plus grande échelle, ni combien d'erreurs ils feraient en chemin. Elle savait seulement que Lucas avait avancé sans elle sur plusieurs décisions, puis était revenu au moment où son jugement comptait vraiment.

Elle ajouta au document partagé une nouvelle question à clarifier sur la présentation des données historiques. Pas un ticket demandant à Lucas quoi coder. Une décision à instruire, avec les personnes concernées.

Leur manière de travailler n'était pas devenue moins exigeante.

Elle commençait simplement à déplacer l'exigence au bon endroit.

8 — Plus d’autonomie exige plus de cadre, pas moins

Une organisation peut décider que davantage de personnes doivent agir sans attendre un spécialiste.
La porosité des rôles et les responsabilités étudiées plus haut crée un problème nouveau de coordination : plusieurs personnes peuvent maintenant agir sur un même sujet sans disposer nécessairement des mêmes repères pour décider.
Mais si chaque nouvelle capacité d’action doit être suivie d’une validation humaine systématique, l’autonomie reste théorique.
On a simplement déplacé les passages obligés d’un métier à l’autre.
Le vrai enjeu consiste donc à créer un système dans lequel davantage de décisions peuvent être prises localement sans perdre la cohérence, la sécurité, la qualité ou la direction Produit.
C’est là qu’apparaît un paradoxe important :
plus on veut distribuer l’autonomie, plus le cadre doit être explicite.
L’autonomie ne vient pas de l’absence de règles.
Elle vient de la possibilité d’agir sans demander une autorisation permanente parce que les objectifs, les limites, les risques et les responsabilités sont suffisamment clairs.

Si tout doit être validé, il n'y a pas d'autonomie

Imaginons un Product Engineer à qui l'on donne la responsabilité d'un sujet.

Il parle aux utilisateurs, explore une solution, prototype, développe et prépare le lancement.

Sur le papier, il est autonome.

Mais s'il doit demander au PM chaque micro-arbitrage Produit, au designer chaque détail d'interface et au Tech Lead chaque décision technique, l'organisation n'a pas réellement déplacé la responsabilité. Elle a seulement demandé à une personne de traverser plus de validations.

Le même obstacle concerne les autres métiers : permettre à quelqu'un de construire n'apporte pas grand-chose si chaque geste impose le processus et les validations d'autrefois.

La question n'est donc pas seulement de savoir **qui peut agir**. Il faut savoir **dans quel cadre cette personne peut décider seule**.

Chez Cursor, cette logique est formulée autour d'objectifs et de garde-fous suffisamment clairs pour permettre à des builders d'agir sans dériver de la vision ou de la roadmap. [1] L'organisation remplace alors une partie de la prescription détaillée par un cadre de décision.

Remplacer une partie des tickets par un cadre de décision

Nous avons vu, avec Linear, pourquoi le ticket entièrement prescrit peut limiter l'initiative. [2] La question devient

maintenant pratique : quelles informations une équipe doit-elle posséder pour arbitrer elle-même sans attendre une nouvelle spécification ?

Cela ne suppose pas de supprimer les tickets. Sur Malorean, j'ai défendu le backlog comme outil d'action et de coordination, non comme mémoire intégrale du Produit. [3]

Un cadre d'autonomie pourrait donc expliciter les informations qui orientent les décisions locales sans figer à l'avance chaque solution.

Ce cadre peut contenir :

- le client ou l'ICP prioritaire ;
- le problème à résoudre ;
- le résultat attendu ;
- les limites du périmètre ;
- les contraintes importantes ;
- le niveau de qualité attendu ;
- ce que l'organisation accepte explicitement de ne pas faire ;
- les seuils au-delà desquels une décision doit être escaladée.

Ramp fournit une illustration concrète de cette logique : clarifier les ICP cœur, les problèmes cœur et ce que l'entreprise veut résoudre particulièrement bien permet aussi d'accepter de refuser des opportunités lucratives lorsqu'elles tirent le produit dans une autre direction. [4]

Figma rappelle de son côté qu'aller rapidement de 0 à 1 ne suffit pas. Les cas réels, les exceptions et le niveau de qualité

restent déterminants pour transformer un prototype rapide en bon produit. [5]

Cette grille illustre un choix d'organisation : ne pas seulement préciser **ce qu'il faut faire**, mais rendre visibles les limites qui doivent être respectées. Elle n'est pas présentée ici comme une méthode universelle.

Le contexte devient une infrastructure

Pour prendre de bonnes décisions localement, il faut avoir accès aux informations qui permettent de comprendre le problème.

C'est probablement l'un des changements organisationnels les plus importants de l'ère des agents.

Une organisation ne peut pas demander à davantage de personnes de décider tout en conservant l'essentiel du contexte dans la tête de quelques spécialistes. Linear souligne que les retours clients, la direction stratégique, les décisions et le code doivent être accessibles aux humains comme aux agents. [2]

Sur Malorean, j'ai décrit le rôle du PM comme celui d'un **architecte du Contexte** : transformer une connaissance dispersée — interviews, objections commerciales, support, décisions, données, sources et signaux faibles — en un actif organisationnel réutilisable. [6]

Cette idée change profondément la fonction de la documentation.

L'objectif n'est plus de produire davantage de pages.

Il est de rendre accessibles les éléments dont quelqu'un a besoin pour prendre une décision sans reconstruire tout le raisonnement depuis zéro.

Un bon contexte permet à une personne de comprendre :

- ce que les clients ont réellement dit ;
- quelles hypothèses ont déjà été testées ;
- quelles contraintes comptent ;
- quelles décisions structurantes ont été prises ;
- quels compromis ont déjà été acceptés ;
- quelles métriques permettent de juger le résultat.

Plus l'autonomie augmente, plus ce contexte doit être fiable et partagé.

Sinon, l'entreprise distribue de la capacité d'action sans distribuer la capacité à comprendre.

Garder la mémoire du pourquoi

Cette redistribution crée un autre problème.

Davantage de décisions sont prises localement, parfois par des personnes différentes, parfois avec des agents qui n'étaient pas présents lors des arbitrages précédents.

Or une décision sans mémoire redevient vite négociable.

On se souvient de la conclusion, mais plus du contexte.

On sait qu'une règle existe, mais plus pourquoi elle a été choisie.

Un nouveau client arrive, un nouveau manager prend son poste, une équipe change, et une discussion déjà tranchée recommence avec moins d'information que la première fois.
C'est précisément le rôle que j'attribue aux **Product Decision Records**. [7]
Un PDR ne documente pas chaque micro-arbitrage.
Il trace les décisions Produit suffisamment structurantes pour guider plusieurs équipes, plusieurs produits ou plusieurs situations futures.
Il conserve notamment :

- le contexte ;
- les alternatives envisagées ;
- la décision retenue ;
- les raisons du choix ;
- les conséquences positives et négatives ;
- les risques acceptés ;
- les conditions qui justifieraient de rouvrir la décision ;
- les personnes qui ont arbitré ou été consultées.

Le but n'est pas d'ajouter de l'administration.
Au contraire, un bon PDR doit éviter de rouvrir cinquante fois la même discussion.
Une distinction devient alors utile :
le code garde la trace de ce qui a été construit ; la mémoire de décision doit garder la trace de pourquoi certaines règles structurantes ont été choisies.
L'autonomie durable a besoin des deux.

Déplacer les garde-fous dans le système

Même avec un bon contexte, il serait inefficace de demander à chaque collaborateur de mémoriser toutes les règles de l'organisation.

Certaines entreprises déplacent ainsi des règles répétitives de la validation humaine vers les outils. Il s'agit d'un mécanisme observable, pas d'une garantie que l'automatisation suffise à assurer la qualité.

Meta utilise un Diff Risk Score pour estimer la probabilité qu'un changement provoque un incident et alimenter des workflows sensibles au risque. [8]

Cloudflare transforme des standards Engineering explicites en contrôles automatisés. Certaines règles obligatoires peuvent empêcher l'approbation d'un changement tant qu'il n'a pas été corrigé ou qu'une dérogation explicite n'a pas été accordée. [9]

Ce mouvement est important parce qu'il change le rôle des experts.

Un Garant technique n'a pas besoin de rappeler manuellement la même règle dans cinquante pull requests si cette règle peut être vérifiée automatiquement.

Un Garant Design n'a pas besoin de valider chaque bouton si le Design System rend le bon choix facile et le mauvais choix visible.

Un Garant Produit n'a pas besoin de réexpliquer chaque semaine les mêmes contraintes si les décisions structurantes, la cible et les objectifs sont accessibles.

Le cadre devient une infrastructure.

Tests, CI, Design System, standards, règles de sécurité, templates, instructions pour agents, métriques et contexte partagé prennent en charge une partie du contrôle autrefois assuré par des personnes.

L'expertise humaine peut alors se concentrer sur les exceptions.

Toutes les décisions ne méritent pas le même contrôle

C'est ici que la notion de risque devient centrale.

Une traduction de texte et une modification du système de permissions ne doivent pas suivre le même circuit de validation.

Un changement local et réversible n'a pas besoin du même niveau de validation qu'une migration de données ou une décision qui modifie le positionnement de l'entreprise.

Une façon de lire ces organisations consiste à remplacer la question binaire — « faut-il valider ? » — par une question graduée : **quel niveau de contrôle ce risque justifie-t-il ?**

Les trois niveaux qui suivent constituent **une grille de lecture proposée ici**, et non une pratique attestée dans toutes les entreprises citées.

Autonomie locale

La décision reste dans les objectifs et les standards connus.

Elle est réversible, ses conséquences possibles sont limitées et elle ne change pas une règle structurante.

L'owner décide.

Validation ciblée

La décision introduit une nouveauté, une dérogation ou un risque significatif dans un domaine.

Le Garant concerné intervient : Dev, Design, Produit ou autre.

Il ne reprend pas nécessairement le sujet. Il apporte le jugement spécifique qui manque.

Escalade

La décision change les limites du système lui-même.

Côté technique : architecture structurante, sécurité, données sensibles, migration critique, risque réglementaire.

Côté Design : nouvelle interaction importante, rupture avec le système de design ou conséquences majeures pour d'autres parcours.

Côté Produit : changement de cible, proposition de valeur, pricing, investissement majeur, roadmap ou arbitrage stratégique.

Dans ce cas, l'owner n'est plus simplement en train de décider **à l'intérieur** du cadre.

Il propose de modifier le cadre.

Dans le modèle proposé ici, c'est le type de situation qui appelle un Garant ou un niveau de décision supérieur.

Sans limites explicites, deux risques opposés

L'absence de limites de décision claires expose l'organisation à deux risques opposés.

La première est la bureaucratie.

Tout remonte. Les experts deviennent des files d'attente. Les responsables doivent approuver des décisions triviales et l'autonomie disparaît.

La seconde est la dilution.

Chacun décide localement sans savoir quand une décision cesse d'être locale. Des choix contradictoires s'accumulent jusqu'à produire un problème visible pour le client ou l'entreprise.

Sur Malorean, j'ai décrit plusieurs outils de cohérence organisationnelle qui servent précisément à rendre ces frontières plus explicites : langage partagé, one-pager, hiérarchie des métriques, journal de décisions et règles d'escalade. [10]

Leur valeur n'est pas de produire plus de documentation.

Elle est de forcer certaines questions à devenir explicites :

- qui tranche ?
- sur quels sujets ?
- à quel moment ?
- qu'est-ce qui peut rester local ?
- qu'est-ce qui doit remonter ?
- dans quelles conditions une décision peut-elle être rouverte ?

Ces deux risques conduisent à une hypothèse d'organisation : **mieux situer les décisions** peut compter davantage que multiplier ou supprimer les validations.

Décider signifie aussi accepter une perte

Les garde-fous ne peuvent pas être seulement techniques.

Une partie essentielle du cadre concerne la capacité à faire de vrais arbitrages.

Les organisations produisent souvent des décisions suffisamment vagues pour maintenir l'apparence du consensus. Chacun peut y projeter sa propre priorité et l'exécution découvre plus tard que les contraintes étaient incompatibles.

J'appelle cela les **décisions molles**. [11]

Elles sont particulièrement dangereuses dans un système très autonome.

Si l'objectif est ambigu, chaque équipe peut prendre une décision localement raisonnable mais globalement incohérente.

Donner davantage d'autonomie exige donc aussi de rendre visibles les pertes acceptées.

Quel client passe en premier ?

Quel segment ne sera pas servi maintenant ?

Quel niveau de qualité est non négociable ?

Quelle contrainte peut être sacrifiée temporairement ?

Quel objectif domine lorsqu'il entre en conflit avec un autre ?

L'autonomie n'a pas besoin d'un cadre qui dit tout.

Elle a besoin d'un cadre qui rend les vrais arbitrages suffisamment nets pour que les décisions locales convergent.

Mesurer le résultat devient le dernier garde-fou

L'autonomie ne s'arrête pas au moment où quelque chose est livré.

Si une personne possède réellement un sujet, elle doit aussi regarder ce qui se passe après.

Le premier chapitre distinguait déjà quantité produite et valeur créée. À ce stade, le sujet est différent : **comment la personne qui a pris une décision sait-elle ensuite si elle était pertinente ?**

Canva fournit un mécanisme concret : ses équipes peuvent relier leurs expérimentations à une estimation de leur impact commercial. [12]

Pour chaque sujet, cela conduit à préciser :

- quelle transformation était attendue ;
- quelle mesure permet de l'observer ;
- quelle qualité devait être préservée ;
- ce que les utilisateurs ont réellement fait ;
- s'il faut continuer, corriger, abandonner ou approfondir.

La mesure ferme la boucle de responsabilité.

Sans elle, l'autonomie peut devenir une simple autorisation de produire.

Avec elle, elle devient une responsabilité sur le résultat.

L'autonomie n'est pas un retrait du management

Le modèle d'équipes autonomes de Marty Cagan, présenté au chapitre 1.4, aide ici à préciser le rôle du management. Confier un problème à une équipe sans lui donner de solution prédéfinie exige de clarifier les objectifs et le contexte stratégique. [13] Cagan insiste également sur le développement des compétences, l'accompagnement et le leadership : laisser l'équipe décider n'est pas l'abandonner. [14]

Le rapprochement a toutefois une limite importante. Cagan décrit la responsabilité d'équipes Produit aux compétences complémentaires. Le concept de Garant, proposé dans ce livre, répond à un autre problème : comment préserver la responsabilité d'un domaine quand l'exécution passe désormais d'un métier à un autre ?

Le Garant devient le gardien du cadre, pas le valideur de chaque geste

Le chapitre 1.6 introduisait le Garant comme responsable de la cohérence d'un domaine, même lorsqu'il n'exécute pas lui-même chaque tâche. Ici, son rôle devient concret : définir des règles compréhensibles, automatiser les vérifications répétitives et intervenir lorsque le risque ou la nouveauté dépasse ce qu'une équipe peut décider seule.

S'il doit tout inspecter, il redevient un goulot d'étranglement. Si personne ne répond plus de rien, la responsabilité se dilue.

L'autonomie ne consiste donc ni à valider chaque action ni à supprimer tout contrôle : elle consiste à rendre possibles davantage de décisions locales tout en conservant des responsabilités claires.

Mais cette transformation soulève une question qui dépasse les seuls métiers Produit et Engineering.

Si le contexte circule directement, si les décisions peuvent être prises plus localement et si les agents absorbent une partie de la coordination, **qu'advient-il des couches de management qui assuraient auparavant ces fonctions ?**

Malorean Corp. nous montrera d'abord comment cette question se pose à des personnes concrètes. Puis nous examinerons ce qu'elle implique à l'échelle de l'entreprise.

Bibliographie

[1] How Cursor Is Navigating the New Shape of Product Craft — Atlassian / Cursor

- URL : https://www.atlassian.com/blog/how-we-build/how-cursor-is-navigating-the-new-shape-of-product-craft
- Montre comment objectifs, garde-fous et jugement Produit permettent à des rôles plus poreux d'agir sans dériver de la direction choisie.

[2] Issue tracking is dead — Linear

- URL : https://linear.app/next

- Défend un système où feedback client, stratégie, décisions et code partagent davantage de contexte que dans une chaîne centrée tickets.

[3] Le backlog n'est pas un dépotoir : c'est un outil d'action — Malorean

- URL : https://malorean.net/articles/2026-06-03-le-backlog-nest-pas-un-depotoir.html
- Distingue le backlog comme outil temporaire de coordination de la mémoire, du contexte et du raisonnement Produit durables.

[4] 5 insights from Ramp's CPO on building for hypergrowth — Ramp

- URL : https://ramp.com/blog/5-insights-from-ramps-cpo-reddit-ama
- Insiste sur les ICP et problèmes cœur, ainsi que sur la nécessité de refuser les opportunités qui dispersent le produit.

[5] What It Takes to Build Great Products Now — Figma

- URL : https://www.figma.com/blog/what-it-takes-to-build-great-products-now/
- Rappelle que prototyper rapidement ne suffit pas : cas réels, exceptions et qualité restent décisifs pour construire un produit durable.

[6] Le PM comme architecte du Contexte — Malorean

- URL : https://malorean.net/articles/2026-06-08-le-pm-comme-architecte-du-contexte.html
- Propose de transformer signaux, sources, décisions et arbitrages dispersés en contexte réutilisable pour améliorer les décisions futures.

[7] Product Decision Record : tracer les choix produit qui structurent l'entreprise — Malorean

- URL : https://malorean.net/articles/2026-06-03-product-decision-record-tracer-les-choix-produit.html
- Formalise une mémoire courte des décisions structurantes : contexte, alternatives, choix, conséquences et conditions explicites de réévaluation.

[8] Diff Risk Score: AI-driven risk-aware software development — Meta

- URL : https://engineering.fb.com/2025/08/06/developer-tools/diff-risk-score-drs-ai-risk-aware-software-development-meta/
- Présente un score de risque permettant d'adapter les workflows de contrôle à la probabilité d'incident d'un changement logiciel.

[9] How Cloudflare enforces engineering standards using AI — Cloudflare

- URL : https://blog.cloudflare.com/engineering-standards-enforcement/

- Montre comment des standards explicites deviennent des contrôles automatisés capables de bloquer certaines violations avant mise en production.

[10] Les outils de cohérence organisationnelle — Malorean

- URL : https://malorean.net/articles/2026-06-02-les-outils-de-coherence-organisationnelle.html
- Relie langage commun, objectifs, métriques, mémoire des décisions et règles d'escalade pour rendre les arbitrages organisationnels explicites.

[11] Pourquoi les organisations préfèrent les décisions molles — Malorean

- URL : https://malorean.net/articles/2026-06-02-pourquoi-les-organisations-preferent-les-decisions-molles.html
- Analyse les non-décisions consensuelles qui masquent les pertes, diluent la responsabilité et déplacent les contradictions vers l'exécution.

[12] Measuring Commercial Impact at Scale — Canva

- URL : https://www.canva.dev/blog/engineering/measuring-commerical-impact-at-scale/
- Décrit une mesure self-service de l'impact commercial reliant expérimentations locales, décisions Produit et création de valeur business.

[13] Team Objectives – Empowerment — Marty Cagan, Silicon Valley Product Group (2020)

- URL : https://www.svpg.com/team-objectives-empowerment/

- Défend l'attribution aux équipes de problèmes à résoudre, accompagnés du contexte stratégique nécessaire aux décisions locales.

[14] Product Leadership Is Hard — Marty Cagan, Silicon Valley Product Group (2020)

- URL : https://www.svpg.com/product-leadership-is-hard/
- Explique pourquoi des équipes Produit autonomes demandent un leadership plus exigeant et un accompagnement actif, plutôt qu'un retrait du management.

9 — Malorean Corp. — Deux décisions raisonnables

Le lundi matin, Vivien, responsable de la coordination des deux équipes de développement, commença sa réunion par la même question que depuis des années.

— Qui peut me dire où on en est sur les sujets de la semaine ?

Il avait déjà son tableau ouvert. Les tickets, les responsables, les échéances et les blocages étaient rangés dans les colonnes qu'il avait patiemment perfectionnées. Il connaissait les deux équipes de cinq développeurs, leurs forces, leurs habitudes, les sujets sur lesquels il fallait insister et ceux qu'il valait mieux laisser mûrir.

Personne ne répondit immédiatement.

Puis Lucas prit la parole.

— Pour la traçabilité, j'ai mis à jour le Mini PRD vendredi. On a testé la recherche par lot avec le client. La prochaine étape, c'est de valider les cas où les données sont incomplètes. J'ai noté les questions ouvertes dans le contexte.

— Et ta date de livraison ? demanda Vivien.

— Elle dépend encore des contrôles sur le stock historique. Thomas doit regarder un point avec moi.

Thomas acquiesça. De son côté, il travaillait sur une amélioration du circuit de préparation des expéditions. Il avait également mis à jour l'avancement dans leur outil.

Vivien passa aux autres développeurs. Les réponses se ressemblaient. Chacun connaissait mieux son problème, avait pris certaines décisions et pouvait expliquer la prochaine étape. Quelques-uns demandaient encore qu'on les aide à découper leur travail. Les autres n'avaient pas attendu la réunion pour avancer.

À la fin, Vivien regarda son tableau.

Tout ce qu'il venait d'entendre y figurait déjà, parfois avec davantage de détails.

— Bon, dit-il. On a fait le tour.

La réunion avait pris douze minutes.

Avant, il lui en aurait fallu quarante-cinq.

Vivien ne savait pas très bien s'il devait s'en réjouir.

Il travaillait chez Malorean depuis assez longtemps pour avoir connu les développeurs tous réunis au bureau du Havre. Il avait organisé les livraisons, géré les retards, négocié les priorités avec le Produit et expliqué au directeur pourquoi une demande annoncée pour vendredi ne sortirait pas avant le mardi suivant.

Quand l'entreprise était passée au télétravail, après le Covid, il avait fait évoluer sa méthode. Les discussions de couloir ne suffisaient plus. Il avait demandé que les décisions soient écrites, que les blocages soient visibles, que chacun note ce qu'il avait convenu avec un autre service.

C'était devenu son travail quotidien : faire circuler l'information que personne n'avait entièrement sous les yeux. Il aimait ça. Il aimait particulièrement les moments où un projet mal engagé retrouvait son chemin parce que quelqu'un avait enfin rapproché les bonnes personnes.

Depuis l'arrivée des outils d'IA dans le travail des équipes, pourtant, quelque chose avait changé. Les développeurs produisaient plus vite, mais surtout certains avaient commencé à se saisir des sujets autrement. Lucas discutait directement avec des clients. Alice partageait un contexte Produit qui expliquait les raisons des décisions, pas seulement la liste des fonctionnalités à développer. Bob commençait à adopter la même pratique sur ses propres sujets.

Et désormais, un agent préparait pour Vivien une partie des synthèses qu'il construisait autrefois à la main.

Le travail n'avait pas disparu. Il lui échappait par morceaux, ou se faisait ailleurs.

Il n'en avait encore parlé à personne.

Le mercredi suivant, Lucas montra à Alice une évolution de la démonstration de traçabilité.

— On a testé ce qui se passe quand un fournisseur signale un lot potentiellement défectueux. On peut retrouver les fabrications qu'on connaît. Et j'ai ajouté un état « sous

vérification » pour éviter de faire croire que le lot est libre de toute question.

Alice observa l'écran.

Un bandeau indiquait que la situation devait être examinée. Pour les fabrications anciennes, un second message précisait que certaines informations n'étaient pas disponibles.

— C'est plus honnête que la première version, dit-elle.

— Oui. Le responsable qualité du client nous a justement demandé de ne pas mélanger « aucun problème identifié » et « nous n'avons pas les informations nécessaires ».

Alice approuva. C'était une distinction dont ils avaient déjà parlé dans le document de contexte.

— Tu l'as notée dans le Mini PRD ?

— Oui. Et j'ai ajouté des cas de test.

— Alors continue.

Elle n'eut pas besoin de décider de l'emplacement du bandeau ou du texte exact du message. Lucas savait quel résultat ils cherchaient à obtenir. Il avait choisi une présentation et prévu de la confronter au client.

Un peu plus tard, Thomas présenta à Vivien une amélioration du module d'expédition.

Le problème était différent. Chez certains clients, une commande pouvait rester bloquée par des états intermédiaires ou mal renseignés, même lorsqu'aucune interdiction explicite ne s'appliquait. Les équipes logistiques

perdaient du temps à faire corriger des dossiers sans risque identifié.

Thomas avait donc clarifié les règles de préparation : un lot marqué « bloqué » empêchait le départ ; les autres situations restaient visibles et suivaient le circuit habituel, avec les contrôles existants.

— Tu es sûr de ne rien changer au comportement des autres modules ? demanda Vivien.

— Sur les cas existants, oui. J'ai ajouté les tests de non-régression. Je n'ai pas touché aux règles de blocage.

Vivien n'avait aucune raison de douter de la prudence de Thomas. Celui-ci connaissait parfois des exceptions que même la documentation historique ne mentionnait plus.

— Très bien. Fais revoir la modification avant qu'on aille plus loin.

Thomas acquiesça.

Les deux changements n'avaient pas encore été réunis dans un même scénario.

Le problème apparut le vendredi, sur une instance de test intégrée.

Une personne du support avait préparé un parcours complet : réception d'une matière première, fabrication, alerte fournisseur, identification des produits concernés, puis préparation d'une expédition.

Lucas suivait la démonstration en visioconférence. Thomas était également présent. Alice avait demandé à être invitée pour observer comment le futur fonctionnement serait compris par quelqu'un qui ne l'avait pas conçu.

Tout se passa bien pendant les premières minutes.

Le numéro de lot était retrouvé. Les fabrications apparaissaient. Le bandeau « sous vérification » s'affichait. Puis la personne du support passa dans le module d'expédition.

Le bon de livraison restait préparatoire. Aucun message n'empêchait de poursuivre.

— Attendez, dit Alice. On parle bien du même lot ?

Lucas se pencha vers son écran.

— Oui. Pourquoi ?

— Il est sous vérification là-haut, mais ici on pourrait continuer comme s'il n'y avait rien de particulier.

Thomas ouvrit le détail de la commande.

— Il n'est pas bloqué. Il est sous vérification. Ce n'est pas le même état.

— Pour moi, répondit Lucas, « sous vérification » voulait justement dire qu'on ne devait pas le considérer comme validé tant que quelqu'un n'avait pas tranché.

Thomas tourna la tête vers sa caméra.

— Alors pourquoi tu ne l'as pas marqué « bloqué » ?

— Parce que je ne voulais pas inventer une interdiction générale. On a des clients qui doivent pouvoir enquêter sans

arrêter toute leur activité. C'est à leur responsable qualité de décider ce qui est réellement concerné.

— Et dans mon module, répliqua Thomas, on a des lots anciens avec des informations incomplètes. Si je commence à bloquer tout ce qui n'est pas explicitement validé, certaines commandes n'avanceront plus du tout.

Alice regarda les deux écrans.

Elle comprenait les raisons de Lucas. Elle comprenait celles de Thomas.

Le problème, c'était qu'un utilisateur pouvait lire « sous vérification » dans une partie de l'ERP et ne rencontrer aucun avertissement bloquant dans une autre.

— On arrête là pour la démonstration, dit-elle.

Personne ne protesta. Aucune donnée client réelle n'avait été touchée, et les changements n'étaient pas partis en production. C'était précisément à cela que servait le test.

Mais le silence qui suivit était moins confortable qu'après une démonstration réussie.

Vivien rejoignit l'appel lorsqu'on lui expliqua ce qui venait de se passer.

Il laissa Lucas décrire sa décision, puis Thomas raconter la sienne. Il posa quelques questions sur les tests et sur le périmètre de chaque évolution.

— Vous avez chacun votre Mini PRD ? demanda-t-il.

— Oui, répondit Lucas.

— Et vos règles de décision ?

— Oui, répondit Thomas. Je n'ai pas changé le blocage existant. C'était une amélioration locale du parcours de préparation.

— Moi non plus, dit Lucas. J'ai ajouté de l'information pour éviter de masquer un doute. On avait justement écrit que c'était important.

Vivien ouvrit les deux documents côte à côte.

Ils n'étaient pas faux. Ils ne parlaient simplement pas de la même chose.

Le premier décrivait comment signaler une incertitude dans la traçabilité. Le second décrivait quand une expédition devait s'arrêter. Aucun des deux n'avait été rédigé pour répondre à la question qui les reliait : que devait faire le logiciel lorsqu'un lot était suspect, mais pas encore explicitement bloqué ?

— On a un trou entre les deux sujets, dit Vivien.

Thomas acquiesça.

— Techniquement, on peut choisir plusieurs comportements. Mais il faut savoir lequel on veut.

— Et il faut demander au client ce que son responsable qualité fait réellement dans ce cas, ajouta Lucas. J'ai des hypothèses, pas une règle valable pour tout le monde.

Alice prit la parole.

— Je vais clarifier le comportement Produit attendu. Pas seule. On a besoin du support, de Lucas, de Thomas et probablement d'un client pour vérifier les conséquences.

Vivien releva les yeux.

— Alors je vous organise une réunion demain.

Alice hésita.

— Tu peux nous aider à réunir les bonnes personnes, oui. Mais ce n'est pas à toi de porter la décision à notre place.

Elle ne lui disait pas cela pour le mettre à l'écart. Pourtant, Vivien ressentit la phrase comme un déplacement.

Il avait passé des années à être celui qui ramenait les problèmes dans le bon circuit. Cette fois, les personnes concernées savaient déjà qu'elles devaient se parler. Elles n'avaient pas besoin qu'il transforme leur désaccord en une suite de tickets.

Il leur fallait autre chose : un arbitrage qu'aucun des deux développeurs ne pouvait prendre seul.

Le lendemain, Alice commença par demander à chacun ce qu'il cherchait à éviter.

Lucas ne voulait pas qu'un responsable qualité croie disposer d'une traçabilité complète lorsque des données manquaient.

Thomas ne voulait pas qu'une règle imprécise arrête inutilement les opérations de clients dont les historiques étaient incomplets.

Le support rappela qu'une alerte fournisseur n'était pas toujours synonyme de rappel immédiat. Certains clients devaient d'abord identifier le périmètre concerné, consulter leurs équipes qualité et décider des actions à mener.

— Très bien, dit Alice. Donc notre erreur est peut-être d'avoir essayé de faire porter deux sens différents au même état.

Elle partagea une proposition.

Un lot pouvait être identifié comme **à examiner** sans que l'ERP prétende prendre, à la place de l'entreprise cliente, une décision réglementaire ou qualité. En revanche, les parcours susceptibles d'utiliser ou d'expédier ce lot devaient rendre l'alerte visible et imposer le circuit de confirmation prévu, au lieu de la laisser disparaître.

Les informations historiques incertaines devaient rester signalées comme telles. Et les cas où le logiciel ne disposait pas des données nécessaires ne pouvaient pas être présentés comme des absences de risque.

— Ça, dit Thomas, on peut le vérifier dans le module d'expédition. Mais je veux qu'on teste aussi les anciennes commandes.

— Et je veux qu'on revoie les termes avec le client, ajouta Lucas. « À examiner » n'aura peut-être pas le même sens pour son responsable qualité.

Alice acquiesça.

Ils n'avaient pas terminé la conception. Ils avaient enfin identifié qui devait choisir quoi.

Elle nota la décision provisoire et les questions ouvertes dans le contexte partagé, avec les raisons et les cas à vérifier. Elle ajouta une condition à leur grille : lorsqu'une décision locale modifiait la signification d'une information utilisée par

plusieurs modules, elle ne relevait plus automatiquement du vert, même si le code semblait réversible.

— On aurait dû voir ça avant, dit Lucas.

— Oui, répondit Alice. Mais c'est justement pour ça qu'on teste. Maintenant, on sait quoi regarder la prochaine fois.

Thomas étudia les deux documents.

— Il faudra aussi un test qui traverse les modules. On en a beaucoup qui vérifient chaque comportement séparément. Beaucoup moins qui vérifient ce qui se passe quand ils se rencontrent.

Vivien, qui écoutait depuis quelques minutes, prit une note. Pour la première fois depuis le début de l'appel, il retrouvait un problème qu'il connaissait bien : personne n'avait une vue complète de toutes les dépendances.

Mais il était frappé par la manière dont la solution se construisait. Alice n'avait pas dicté les écrans. Lucas et Thomas n'attendaient pas un découpage de tâches. Le support apportait le vécu client. Son propre rôle avait consisté à mettre en évidence le manque et à aider les échanges.

Cela avait été utile.

Cela n'avait presque rien à voir avec la réunion de livraison qu'il organisait chaque lundi.

Dans l'après-midi, Thomas reçut un message d'un développeur moins expérimenté de l'autre équipe.

Il ne comprenait pas pourquoi un ancien traitement de stock effectuait deux vérifications apparemment identiques. L'agent de code proposait d'en supprimer une.

Thomas ouvrit le fichier.

— Ne touche pas à ça avant qu'on ait regardé ensemble, répondit-il.

Ils passèrent une demi-heure en partage d'écran. La seconde vérification couvrait un cas hérité d'une ancienne évolution du produit, mal expliqué dans le code. Thomas ne savait plus exactement à quelle livraison elle remontait, mais il savait où chercher.

— Ton outil n'a pas inventé une mauvaise réponse, expliqua-t-il. Il a vu deux contrôles proches et il a proposé de simplifier. Sans l'histoire du logiciel, je comprends pourquoi.

— Tu utilises vraiment jamais l'IA pour coder ? demanda le développeur.

— Pas pour me proposer des modifications à l'aveugle sur ces sujets-là.

— Mais tu viens de passer trente minutes sur quelque chose que Lucas aurait peut-être vérifié en cinq.

Thomas réfléchit.

— Peut-être. Et Lucas aurait peut-être trouvé la même chose. Ce qui compte, c'est que tu comprennes pourquoi le contrôle existe. La prochaine fois, tu sauras poser une meilleure question, à ton outil comme à moi.

Le développeur lui demanda s'il pouvait documenter l'explication.

— Surtout, oui. J'en ai assez d'être le seul à m'en souvenir.

Thomas raccrocha, surpris lui-même par sa dernière phrase. Depuis des années, sa connaissance des exceptions faisait de lui une personne indispensable. Il avait parfois craint que l'IA rende cette expérience moins précieuse. La discussion lui avait montré autre chose : il pouvait aider les autres à acquérir le jugement qui leur manquait, au lieu de garder chaque réponse pour lui.

Il ouvrit le document partagé et rédigea quelques lignes.

Lors de leur rendez-vous du lundi, Hubert demanda à Alice comment avançait l'expérience.

Elle lui raconta les deux décisions incompatibles, le test qui les avait révélées et la correction du cadre.

— Donc vous avez laissé deux développeurs décider, et ils ont failli produire une incohérence ? résuma Hubert.

— Oui.

— Ce n'est pas très rassurant.

— On l'a détectée avant toute mise en production, répondit Alice. Le problème n'est pas que deux personnes aient pris des décisions. Il est qu'aucune de nos règles ne disait comment leurs choix devaient se rencontrer.

Hubert regarda les notes de l'incident.

— Avant, on aurait probablement eu une réunion de validation supplémentaire.

— Peut-être. Mais une réunion de plus n'aurait pas nécessairement fait apparaître la question. Il fallait quelqu'un qui teste le parcours complet et des responsabilités claires quand un choix touche plusieurs domaines.

— Et maintenant, qui décide ?

— Lucas porte son sujet. Thomas porte le sien et nous aide sur les risques techniques. Moi, je suis garante de la cohérence Produit. Pour les cas sensibles, on consulte les spécialistes concernés. On ne peut pas transférer tout ça à un outil.

Hubert hocha lentement la tête.

— Et Vivien ?

Alice prit quelques secondes avant de répondre.

— Il nous a aidés à rapprocher les deux équipes et à comprendre où ça coinçait. Mais il n'avait pas à distribuer les tâches ou à décider du comportement Produit.

Hubert sembla réfléchir à autre chose.

— Je l'ai vu vendredi, dit-il. Il m'a montré le nouveau suivi des projets. Son compte rendu est très bon. Mais je crois que la machine pourrait préparer quatre-vingts pour cent de ce qu'il m'a raconté.

— Lui aussi le sait, répondit Alice.

— Ce que je ne sais pas, c'est ce qu'on doit en conclure.

Alice n'avait pas de réponse toute faite.

— Pas qu'il ne sert plus à rien, dit-elle enfin. On vient de voir qu'il sait faire travailler les gens ensemble. Seulement, ce n'est plus forcément un poste qui consiste à tout faire passer par lui.

Le regard d'Hubert resta fixé sur son écran.

— Il va falloir qu'on en parle avec lui.

Vivien reçut l'invitation pour le lendemain.

Il pensa d'abord à une réorganisation. Depuis quelques semaines, ses journées comportaient davantage de plages vides. Les développeurs tenaient leurs propres sujets. Les outils collectaient automatiquement une partie de l'information qu'il passait auparavant ses soirées à rassembler. Il continuait à gérer des difficultés humaines, à intégrer les nouveaux et à débloquer des situations complexes, mais ces interventions ne remplissaient plus son agenda comme avant.

Il avait aimé être indispensable aux livraisons de Malorean.

Il n'était pas certain d'aimer découvrir qu'il l'était moins.

La réunion commença pourtant par une question qui le surprit.

— Qu'est-ce que tu voudrais continuer à faire, Vivien ? demanda Hubert.

— Mon travail, répondit-il.

Hubert sourit.

— C'est justement ce qu'on essaie de comprendre.

Vivien regarda Alice, invitée pour parler du fonctionnement des équipes, puis revint vers Hubert.

— Je ne veux pas qu'on fasse semblant. Les équipes n'ont plus besoin que je leur demande chaque semaine où elles en sont. Et je n'ai plus envie de passer mon temps à relire des résumés qu'un agent a déjà préparés.

— Qu'est-ce qui leur manque encore ? demanda Alice.

— Les conflits qui ne sont écrits nulle part. Les gens qui hésitent à dire qu'ils sont perdus. Les nouveaux qui ne savent pas à qui demander. Les décisions qui traversent plusieurs équipes. Et parfois, quelqu'un qui prend simplement le temps d'écouter.

Il marqua une pause.

— Mais ça ne veut pas dire que ces choses doivent continuer à passer par moi.

Hubert se redressa.

— Non. Et il faudra qu'on dise clairement qui en répondra si ton rôle change. Je ne veux pas qu'on supprime un titre et qu'on découvre ensuite que plus personne ne s'occupe des gens.

Vivien approuva.

Ils discutèrent de l'accompagnement des développeurs, des arbitrages transverses et de ce qui devait rester suivi par la direction. Alice parla du rôle de Thomas auprès des collègues qui apprenaient à mieux comprendre le code historique. Aucune solution définitive ne sortit de l'appel.

Quand il se termina, Vivien resta quelques instants devant son calendrier.

Il se sentait moins menacé que la veille, mais pas rassuré pour autant. Son métier n'avait pas disparu. Il était en train de se décomposer en plusieurs responsabilités dont certaines pouvaient être assumées autrement.

Ce soir-là, il ferma plus tôt son ordinateur.

Pour une fois, il n'avait pas de retard de livraison à rattraper.

Et il ne savait pas encore ce qu'il allait faire de ce temps retrouvé.

10 — L'IA va-t-elle faire disparaître les couches intermédiaires de l'entreprise ?

Une équipe peut aujourd'hui consulter directement ses décisions, agir dans un cadre connu et confier une partie du suivi à des agents. Elle a moins souvent besoin d'un responsable pour lui transmettre l'information ou organiser chaque étape du travail.

Dès lors, **que deviennent les couches hiérarchiques qui remplissaient ce rôle ?**

La réponse semble évidente : si l'IA reprend la coordination, les managers intermédiaires perdent leur utilité. Pourtant, un manager ne fait pas que transmettre des consignes. Il arbitre, accompagne, relie des équipes et prend en charge les problèmes qui n'entrent dans aucune procédure.

La vraie question est donc moins de savoir combien de postes vont disparaître que de comprendre **où iront leurs fonctions et leur pouvoir**.

Une hiérarchie construite pour coordonner les décisions

Un client important signale un problème. Le Support le transmet à son responsable, qui contacte le Produit. Le Produit demande une estimation à l'Engineering. Plusieurs responsables réunissent leurs informations, comparent les urgences, puis l'arbitrage remonte éventuellement encore d'un niveau.

Ce parcours n'est pas nécessairement absurde. Chaque responsable détient une partie du contexte et peut engager une partie de l'entreprise. Il devient pesant lorsque cette chaîne constitue la seule manière de comprendre un problème et de décider.

Historiquement, la hiérarchie a permis aux entreprises de grandir malgré une information fragmentée. Les échelons intermédiaires traduisent les objectifs de la direction en priorités, coordonnent les équipes, surveillent les dépendances et arbitrent les conflits de ressources. Ils participent aussi à l'évaluation des collaborateurs, à leur progression et à la résolution des difficultés humaines.

Mais toutes ces activités n'ont pas la même nature. Préparer un compte rendu d'incident est différent de décider d'abandonner une opportunité commerciale. Suivre une échéance n'exige pas le même jugement que résoudre un conflit ou aider quelqu'un à progresser.

J'avais déjà défendu sur Malorean une distinction proche à propos du Product Owner : un poste peut perdre sa justification comme simple relais sans que la fonction qu'il était censé assurer cesse d'être nécessaire. Il faut interroger le travail accompli, pas seulement le titre porté par celui qui l'accomplit. [1]

Si les niveaux intermédiaires se réduisent, qui assurera les fonctions qu'ils exerçaient ?

Le travail à distance avait déjà changé les règles de la coordination

L'IA n'a pas inventé cette question. Les entreprises qui travaillent à distance, parfois dans plusieurs pays et fuseaux horaires, ont dû apprendre à décider sans compter sur les conversations de couloir ou les réunions improvisées.

Chez GitLab, le principe « handbook-first » consiste à rendre les règles et le contexte accessibles par écrit. Chacun peut proposer une modification ; les responsables du domaine décident de son intégration. Un salarié peut ainsi retrouver une information sans attendre qu'un manager la lui transmette. [2]

Alan décrit une pratique différente mais complémentaire. Les décisions importantes sont discutées dans des fils GitHub : une personne prend la décision et la met en œuvre, une autre reste responsable du succès du sujet, tandis que les personnes concernées sont consultées ou informées. Le débat peut être ouvert sans exiger l'unanimité. [3]

Chez 37signals, à l'origine de Basecamp, une part importante de la coordination passe par des échanges écrits, des bilans et des informations conservées avec les projets plutôt que dispersées dans des conversations éphémères. [4]

Il ne s'agit pas de trois entreprises sans management. Il s'agit de trois façons de **réduire la dépendance à une personne qui détient l'information ou doit rappeler sans cesse ce qui a été décidé**.

Ce précédent compte pour les agents IA. Un système peut consulter une règle, retrouver un arbitrage et identifier un responsable à condition que l'organisation ait rendu ces informations explicites. Une décision restée dans la tête d'un manager reste inaccessible aux collègues comme aux agents.

Attention toutefois à ne pas idéaliser le travail à distance. Une étude menée chez Microsoft auprès de plus de 61 000 employés a notamment observé des réseaux de collaboration devenus plus cloisonnés après le passage au télétravail. Écrire davantage ne garantit ni le partage du contexte ni la coopération entre équipes. [5]

Le véritable héritage du travail à distance n'est pas la disparition du bureau : c'est la possibilité d'organiser une partie du travail autour d'informations et de décisions accessibles sans intermédiaire permanent. L'IA peut s'appuyer sur cet acquis. Elle ne le crée pas à elle seule.

L'IA commence à reprendre une partie de ce travail

Un document conserve une règle ; un tableau de bord affiche un retard. Un agent peut aller plus loin : rapprocher plusieurs informations, préparer un état d'avancement, détecter une contradiction, proposer un traitement ou exécuter une action autorisée.

Ce sont précisément des activités qui occupaient une partie du temps des managers : consolidation, reporting, suivi et préparation des arbitrages. Leur coût peut diminuer. Leur disparition n'est pas acquise.

Une analyse exploratoire de HEC Paris, nourrie notamment de 45 entretiens, suggère que les tâches managériales structurées et riches en données se prêtent mieux à l'automatisation que celles qui dépendent de la confiance, des relations humaines et du jugement. Elle ne mesure ni les heures de management réellement supprimées ni les postes devenus inutiles. [6]

Une expérience menée chez Procter & Gamble montre également qu'en situation d'innovation Produit, une personne travaillant avec l'IA peut obtenir des résultats comparables à un binôme sans IA. L'outil aide notamment à croiser des perspectives de métiers différents. Mais faciliter une collaboration ponctuelle ne signifie pas remplacer un responsable sur la durée. [7]

Dans une autre expérience, auprès de 7 137 salariés de 66 entreprises, les utilisateurs réguliers d'un assistant IA ont économisé environ deux heures de courriels par semaine durant la seconde moitié de l'étude. Les chercheurs n'ont pourtant pas observé de transformation significative de la composition des tâches sur six mois. L'étude porte sur un assistant bureautique, pas sur des agents qui exécuteraient seuls des processus entiers. [8]

Ce décalage est décisif : **un gain de productivité individuel n'est pas encore une transformation organisationnelle**. Pour qu'un échelon devienne moins nécessaire, l'information doit réellement atteindre les équipes, les droits de décision

changer et les situations inhabituelles trouver un lieu d'arbitrage. Ce n'est plus seulement un sujet d'outillage, mais d'organisation.

Trois entreprises, trois trajectoires très différentes

Les organisations ne réagissent pas toutes de la même manière. Trois cas permettent de distinguer une réorganisation déjà engagée, une ambition construite autour de l'IA et les difficultés d'une transformation accélérée.

Bayer : une hiérarchie réellement réduite

Avec son modèle *Dynamic Shared Ownership* (DSO), Bayer a redistribué une partie des décisions vers des équipes pluridisciplinaires. Dans sa lettre aux actionnaires pour 2025, le dirigeant du groupe déclare avoir supprimé jusqu'à six niveaux organisationnels et réduit d'environ deux tiers les postes de management. Il revendique des décisions plus rapides et des délais de développement raccourcis. [9]

L'ampleur mérite l'attention, même si les effets sont ici rapportés par Bayer. La transformation suppose de changer les responsabilités quotidiennes, pas seulement les intitulés sur un organigramme : des équipes décident davantage localement, des responsables se consacrent plus à l'accompagnement et certaines fonctions de coordination sont réorganisées.

Une précision change toutefois la lecture du cas : **DSO a été engagé avant l'essor des agents d'IA générative**. Bayer

démontre qu'un grand groupe peut fortement réduire sa hiérarchie ; il ne prouve pas que l'IA en soit la cause.

Block : faire circuler le contexte sans passer par toute la hiérarchie

Block va plus loin dans la place qu'elle souhaite donner à l'IA. Dans un texte cosigné par Jack Dorsey et l'investisseur Roelof Botha, ses dirigeants décrivent une organisation appuyée sur un système d'information partagé.

L'ambition est qu'un collaborateur comprenne une situation et puisse agir sans attendre qu'un responsable lui transmette le contexte. Les auteurs distinguent les contributeurs, les personnes chargées de résultats transversaux (DRI) et des responsables qui continuent à produire tout en accompagnant les autres, appelés *player-coaches*. [10]

Le manager qui servait surtout de relais perdrait ainsi de son importance. En revanche, le développement des personnes, la responsabilité des résultats et les décisions de direction demeureraient.

Il s'agit d'une ambition décrite par l'entreprise et son investisseur, pas d'un bilan indépendant prouvant le succès durable de ce modèle.

Meta : quand le débit dépasse la capacité de contrôle

Le contrepoint vient d'une enquête Reuters sur les projets de réorganisation de Meta en 2026 : équipes plus petites, forte place donnée à l'IA et réduction envisagée ou engagée de certains niveaux. Une deuxième vague de restructuration a finalement été abandonnée.

D'après des documents internes cités par Reuters, les changements sur des logiciels et infrastructures internes ont augmenté bien plus vite que les nouvelles fonctionnalités livrées aux utilisateurs. Dans le même temps, les incidents techniques et le temps consacré à les résoudre augmentaient. Meta a contesté ou nuancé certains aspects du reportage. [11] Ces chiffres ne permettent pas d'attribuer les incidents à l'IA ou à la réduction des managers : plusieurs transformations avaient lieu simultanément. Mais ils soulèvent une question décisive : à quoi sert de produire davantage si l'entreprise ne parvient plus à vérifier, stabiliser et absorber ce qu'elle produit ?

Bayer montre la possibilité d'un aplatissement, Block en dessine une version conçue pour l'IA, et Meta rappelle les risques d'une réorganisation dont le débit augmente plus vite que la coordination. Aucune de ces trajectoires n'annonce à elle seule le futur du management.

Les fonctions de management ne disparaissent pas avec leurs titulaires

Prenons un exemple : un client signale un dysfonctionnement logiciel. Tant que le problème est connu et que les règles de résolution sont claires, l'équipe peut agir sans attendre une succession d'autorisations. Avec des données fiables, elle peut enquêter, appliquer une correction et informer les autres.

Puis deux diagnostics se contredisent. L'agent recommande une action, un développeur une autre. La correction pourrait perturber un service voisin.

Là, quelqu'un doit faire émerger le désaccord, réunir les compétences nécessaires et décider quel risque prendre. L'autonomie n'a pas supprimé la coordination ; elle a changé la manière de l'exercer.

Une expérience de Martin Krzywdzinski et ses coauteurs fournit un indice intéressant. Sur une chaîne de production simulée, 87 binômes ont affronté des pannes avec l'aide d'un système IA parfois erroné. Les équipes auto-organisées communiquaient davantage et résolvaient mieux les problèmes que celles dotées d'un chef désigné. L'analyse souligne le rôle déterminant de la communication. [12]

Deux personnes dans un laboratoire ne représentent pas une grande entreprise. Mais l'étude rappelle une distinction précieuse : **une équipe peut travailler sans chef formel ; elle ne peut pas se passer du travail de coordination.**

Ce travail peut passer par les pairs, un expert, une personne mandatée pour décider ou un responsable hiérarchique. Les choix varient selon les compétences, les risques et les désaccords.

Il reste aussi ce qu'aucun compte rendu automatique ne suffit à accomplir : accueillir quelqu'un, le faire progresser, résoudre une tension, accompagner une décision difficile ou reconnaître un travail. Ces tâches ne sont pas forcément bien

réalisées par tous les managers et peuvent être réparties autrement. Mais il faut encore que quelqu'un en ait la mission et le temps.

Gallup observe que l'usage de l'IA ne fait pas disparaître les inquiétudes relatives à l'emploi et que le soutien perçu au travail est associé à une inquiétude moindre. Cela n'établit pas la nécessité d'un manager traditionnel, mais rappelle que l'organisation est aussi une communauté humaine. [13]

Enfin, qui formera ceux qui devront demain prendre les décisions difficiles ?

J'ai évoqué sur Malorean un risque pour les développeurs juniors : lorsqu'une IA écrit immédiatement le code et corrige les erreurs, elle peut supprimer des occasions d'apprendre à déboguer, à comprendre un échec et à construire son jugement. C'est une inquiétude à étudier, non la preuve d'une perte de compétence généralisée. [14]

Le même problème touche les futurs responsables. On apprend aussi en assistant à un arbitrage, en recevant du feedback et en voyant comment une équipe sort d'un conflit. Une entreprise qui retire des échelons doit prévoir d'autres voies pour transmettre cette expérience.

Au final, plusieurs destinations sont possibles pour chaque fonction : automatisation, prise en charge par l'équipe, transfert à un spécialiste ou remontée vers la direction. **Une tâche peut disparaître, changer de mains ou s'ajouter au travail de quelqu'un d'autre.**

Pour juger de l'efficacité d'une organisation plus plate, compter les postes supprimés ne suffit donc pas. Il faut comparer le suivi évité aux nouvelles charges de supervision des agents, de résolution des incidents, d'accompagnement et de formation. Nous ne disposons pas encore d'un bilan net robuste de cette redistribution.

Une organisation plus plate est-elle vraiment plus autonome ?

Reste une dernière question, moins visible sur les organigrammes : **qui décide, lorsque les intermédiaires disparaissent ?**

On peut imaginer des équipes mieux informées, libres de résoudre davantage de problèmes sans remonter chaque choix. Mais on peut aussi imaginer l'inverse : moins de chefs sur le terrain et davantage de décisions contrôlées directement par la direction à travers les systèmes.

Cette tension existait déjà avant l'IA générative. Dans une recherche sur des entreprises industrielles, Nicholas Bloom et ses coauteurs montrent que les outils donnant davantage d'informations aux salariés sont associés à plus d'autonomie locale. Ceux qui facilitent la communication avec les supérieurs peuvent au contraire favoriser la centralisation : il devient plus simple de demander une instruction au sommet. [15]

Ce précédent ne démontre rien sur les agents actuels. Il éclaire néanmoins un mécanisme : **diffuser les données n'est pas la même chose que distribuer le pouvoir.**

Les systèmes de management algorithmique rendent ce risque plus concret. Ils peuvent attribuer des tâches, proposer des priorités, mesurer des performances et déclencher des alertes. La recherche montre que certaines fonctions de direction ou d'évaluation peuvent être portées par ces outils, même quand des managers humains restent présents pour les appliquer ou traiter les exceptions. [16]

Un système n'apparaît pas comme un supérieur dans l'organigramme. Pourtant, s'il décide quelles tâches chacun reçoit, sur quels critères les résultats sont évalués et dans quelles conditions une action est autorisée, il structure une part de l'autorité.

Il faut alors regarder les décisions derrière les écrans : qui fixe les objectifs des agents ? Qui choisit les règles et peut les changer ? Qui peut les contester ? Et, lorsqu'un algorithme se trompe, qui a réellement le pouvoir d'annuler ce qu'il a déclenché ?

Une équipe peut disposer d'une capacité d'exécution considérable tout en étant incapable de changer ses priorités. Une autre peut conserver un manager tout en décidant vraiment de ses moyens et de ses arbitrages locaux.

J'ai déjà décrit sur Malorean ces « décisions molles » qui donnent l'apparence d'un accord sans rendre explicites le

choix, la perte acceptée ou la personne qui tranche. Retirer un échelon n'y change rien si le pouvoir reste aussi difficile à identifier. [17]

On peut donc imaginer un paradoxe : **moins de hiérarchie visible, mais plus de contrôle centralisé dans les outils**.

Ce n'est pas une fatalité. Les mêmes systèmes peuvent aider les équipes à décider plus localement et rendre les règles de pouvoir plus transparentes. Tout dépend de ce que l'entreprise choisit réellement de déléguer, pas du nombre de cases supprimées dans son organigramme.

Ce qui pourrait réellement disparaître

L'IA va-t-elle faire disparaître les couches intermédiaires de l'entreprise ?

Certains échelons pourraient devenir moins nécessaires. Mais leur disparition ne serait ni automatique ni synonyme de la fin du management.

Les fonctions qui consistent surtout à transmettre de l'information, consolider des états d'avancement ou organiser des validations répétitives peuvent perdre de leur poids. D'autres, faites d'arbitrages, de conflits, de transmission d'expérience et de responsabilité, doivent être assumées autrement.

Bayer montre qu'une réduction massive des niveaux est possible sans que l'IA en soit la cause identifiée. Block propose de confier davantage de coordination à un système partagé. Meta rappelle qu'accélérer l'exécution sans mesurer

la fiabilité et la charge humaine peut fragiliser l'organisation. Rien n'établit encore une disparition générale du management due aux agents.

Le déplacement pourrait néanmoins être profond. Un responsable servirait moins souvent de passage obligé pour obtenir une information ou une validation. Il pourrait consacrer davantage d'attention à la cohérence, aux situations exceptionnelles et au développement des personnes. Ailleurs, certaines de ces fonctions reviendraient directement aux équipes ; d'autres seraient inscrites dans les logiciels ou concentrées dans les mains de quelques décideurs.

Ce n'est pas forcément le management qui devient inutile. C'est la hiérarchie comme mécanisme obligatoire de coordination qui commence à être remise en question.

Pour un développeur, la conséquence est concrète. Quand le travail passe par moins d'intermédiaires, il ne reçoit pas seulement davantage de moyens pour produire. Il rencontre plus directement le besoin, les compromis et les résultats dont il faudra répondre.

C'est le mouvement décrit tout au long de ce premier livre : la capacité d'agir augmente, mais avec elle la nécessité de comprendre, de choisir et d'assumer.

Ce déplacement ne garantit ni de meilleures décisions ni des organisations plus humaines. Il oblige toutefois à les concevoir autrement : partager le contexte, rendre le pouvoir

identifiable, préserver la transmission de l'expérience et accepter de répondre des résultats.

L'enjeu n'est pas seulement de savoir ce que l'IA permet désormais de produire, mais quelle entreprise nous voulons construire autour de cette nouvelle capacité.

Dans le dernier épisode, nous retrouverons Malorean Corp. pour voir comment ses dirigeants et ses équipes vivent, au-delà de l'organigramme, les choix qu'ils ont faits.

Bibliographie

[1] Le PO n'est pas un métier, c'est une fonction — et l'IA n'y change rien, elle accélère — Malorean

- URL : https://malorean.net/articles/2026-08-03-le-po-nest-pas-un-metier-cest-une-fonction.html
- Essai d'auteur qui distingue le travail utile d'une fonction de l'existence d'un poste servant surtout de relais entre décision et exécution.

[2] Handbook First — GitLab

- URL : https://handbook.gitlab.com/handbook/company/culture/all-remote/handbook-first/
- Présente la documentation comme référence partagée et les modifications discutées publiquement, tout en conservant des responsables humains des décisions.

[3] Comment prendre des décisions dans un environnement de travail asynchrone — Alan

- URL : https://alan.com/fr-fr/blog/healthy-business/a/decisions-asynchrones-methode-alan
- Explique comment les décisions sont discutées par écrit, qui tranche, qui reste responsable et qui doit être consulté ou informé.

[4] The 37signals Guide to Internal Communication — 37signals

- URL : https://37signals.com/how-we-communicate
- Décrit une communication interne majoritairement asynchrone, avec des échanges écrits et un contexte conservé dans le temps.

[5] The effects of remote work on collaboration among information workers — Microsoft Research

- URL : https://www.microsoft.com/en-us/research/publication/the-effects-of-remote-work-on-collaboration-among-information-workers/
- Étudie les réseaux de collaboration après le passage au travail à distance et montre un risque de cloisonnement accru.

[6] AI Automates Routine Management and Raises the Value of Leadership — HEC Paris

- URL : https://www.hec.edu/en/dare/tech-ai/ai-automates-routine-management-and-raises-value-leadership

- Analyse exploratoire des tâches managériales à partir d'entretiens et d'évaluations de modèles, sans mesure causale de suppressions de postes.

[7] The Cybernetic Teammate: A Field Experiment on Generative AI Reshaping Teamwork and Expertise — NBER

- URL : https://www.nber.org/papers/w33641
- Expérience chez Procter & Gamble sur la contribution de l'IA à des tâches d'innovation et au croisement d'expertises.

[8] Shifting Work Patterns with Generative AI — NBER

- URL : https://www.nber.org/papers/w33795
- Expérience randomisée dans 66 entreprises : des économies de temps individuelles sans transformation significative des tâches collectives observées sur six mois.

[9] Chairman's Letter — Bayer Annual Report 2025

- URL : https://reports.bayer.com/annual-report-2025/en/to-our-stockholders/chairmans-letter.html
- Bayer rapporte une importante réduction de ses niveaux et postes de management dans sa transformation DSO, antérieure aux agents.

[10] From Hierarchy to Intelligence — Sequoia Capital / Block

- URL : https://sequoiacap.com/article/from-hierarchy-to-intelligence
- Jack Dorsey et Roelof Botha présentent une vision de coordination par système partagé, avec DRI et player-coaches.

[11] Mark Zuckerberg had a bold plan to replace Meta staff with AI. Here's how it imploded — Reuters

- URL : https://www.timeslive.co.za/news/sci-tech/2026-08-26-special-report-mark-zuckerberg-had-a-bold-plan-to-replace-meta-staff-with-ai-heres-how-it-imploded/
- Enquête de Reuters, reproduite par TimesLIVE, sur les tensions organisationnelles et techniques d'un projet de transformation chez Meta.

[12] How team organization influences the ability to solve automation failures — AI & Society

- URL : https://link.springer.com/article/10.1007/s00146-025-02761-5
- Étude expérimentale de binômes autogérés et hiérarchiques confrontés à des pannes d'automatisation, mesurant le rôle de la communication.

[13] Using AI More Does Not Reassure Workers, Managers Do — Gallup

- URL : https://www.gallup.com/workplace/713231/ai-not-reassure-workers-managers-do.aspx
- Enquête sur l'inquiétude des salariés face à l'IA, le sentiment de respect et le soutien perçu au travail.

[14] L'IA ne devrait pas seulement servir à produire dix fois plus. Elle devrait aussi nous obliger à comprendre dix fois mieux ce que nous produisons — Malorean

- URL : https://malorean.net/articles/2026-07-23-lia-produire-dix-fois-plus-comprendre-dix-fois-mieux.html
- Réflexion d'auteur sur les occasions d'apprentissage perdues lorsque l'IA effectue les tâches et corrections qui formaient les développeurs juniors.

[15] The distinct effects of Information Technology and Communication Technology on firm organization — NBER

- URL : https://www.nber.org/papers/w14975
- Distingue les effets opposés de l'accès à l'information et des technologies de communication sur l'autonomie décisionnelle locale.

[16] Algorithmic Management: From Technology to Politics and Theory — Weizenbaum Journal of the Digital Society

- URL : https://ojs.weizenbaum-institut.de/index.php/wjds/article/view/4_3_9/163
- Synthèse des fonctions de direction et d'évaluation exercées par des systèmes algorithmiques et des capacités d'intervention humaine.

[17] Pourquoi les organisations préfèrent les décisions molles — Malorean

- URL : https://malorean.net/articles/2026-06-02-pourquoi-les-organisations-preferent-les-decisions-molles.html

- Essai sur les consensus apparents qui retardent les arbitrages, masquent les choix assumés et rendent difficile l'attribution des responsabilités.

11 — Malorean Corp. — Le temps retrouvé

Deux mois après la démonstration du module de congés, Hubert Félix Duca ouvrit sa réunion du lundi sans demander où en étaient les développements.

— Alors, qu'est-ce qu'on a appris la semaine dernière ?

Alice partagea son écran.

Elle avait devant elle la synthèse des derniers retours clients, les résultats des expérimentations en cours et quelques décisions encore ouvertes. Un agent avait préparé un premier regroupement des informations pendant la nuit. Elle l'avait relu le matin, écarté plusieurs rapprochements trompeurs et ajouté deux questions qu'il n'aurait pas pu formuler tout seul.

— Sur la traçabilité, notre client pilote arrive à retrouver les fabrications concernées beaucoup plus rapidement quand les données sont complètes, commença-t-elle. Mais il reste des cas historiques pour lesquels on doit afficher clairement nos limites. On va tester une nouvelle présentation avec son responsable qualité.

Hubert acquiesça.

— Et sur les autres clients ?

— Trop tôt pour conclure, répondit Alice. C'est justement pour ça qu'on n'a pas généralisé le déploiement.

Bob prit la suite. Il travaillait sur un autre ensemble de sujets Produit et revenait de plusieurs échanges avec les équipes avant-vente et support.

— On a aussi trois demandes qui se ressemblent en apparence, dit-il. L'agent les a regroupées sous une seule idée de fonctionnalité. Mais il y a au moins deux problèmes différents derrière. Je voudrais qu'on clarifie la cible avant de décider d'investir.

— Tu veux combien de développeurs ? demanda Hubert.

Bob sourit.

— Aucun pour le moment.

Hubert lui rendit son sourire.

Bob regarda une notification, puis ajouta :

— Au fait, le support teste depuis hier un nouveau classement des demandes récurrentes. Une personne de l'équipe l'a préparé avec un agent, sur des copies anonymisées, pour voir si les informations arrivent plus vite aux bons interlocuteurs.

— Qui a lancé ça ? demanda Hubert.

— Le support. Alice a confirmé que le périmètre restait interne et réversible, avec une vérification humaine des résultats.

— Tu étais au courant ?

— De l'idée, oui. Pas des détails du test.

Hubert eut le réflexe de demander pourquoi personne ne l'avait prévenu avant de commencer. Il se retint. Ils avaient justement convenu que les initiatives limitées et réversibles n'exigeraient plus son accord.

— Vous me direz si ça fonctionne, dit-il finalement. Et si quelqu'un doit le maintenir, je veux qu'on sache qui.
Il n'avait pas eu à autoriser le test. Il restait responsable des moyens qu'ils consacreraient à une éventuelle généralisation.
Il n'y avait pas si longtemps, cette réponse aurait semblé étrange dans une réunion consacrée au Produit. On cherchait surtout à déterminer ce que les développeurs devaient réaliser et dans quel ordre. Désormais, Alice et Bob passaient davantage de temps à choisir les problèmes à résoudre, à vérifier que les initiatives restaient cohérentes et à intervenir quand une décision dépassait le cadre établi.
Leurs intitulés de poste n'avaient pas changé. Dans les documents administratifs, ils étaient toujours Product Managers.
Mais leur travail quotidien ne ressemblait plus tout à fait à celui d'il y avait deux mois.

7

Le changement le plus visible était peut-être celui qui ne faisait plus événement.
Un développeur parlait à un client. Un autre proposait un Mini PRD pour une amélioration de son module. Une commerciale fabriquait une variante de démonstration dans l'environnement prévu à cet effet. Un membre du support signalait directement une incohérence dans une règle Produit.
Cela ne provoquait plus de grande discussion sur les frontières entre métiers.

Malorean n’avait pas annoncé que tout le monde pouvait tout faire. L’entreprise avait simplement cessé de supposer qu’une action devait obligatoirement passer par le titulaire habituel du poste.

Les deux équipes de cinq développeurs existaient toujours. La majorité de leurs membres portaient désormais certains sujets de bout en bout : comprendre le problème, proposer le cadrage, discuter avec les personnes concernées, développer, tester, puis suivre les résultats.

Certains avaient adopté cette façon de travailler plus vite que d’autres. Deux développeurs préféraient encore être accompagnés de près sur le cadrage. Un autre était excellent sur des problèmes techniques complexes, mais n’avait aucune envie de multiplier les rendez-vous clients. Personne n’essayait de les transformer de force en copies de Lucas.

Il restait aussi des tickets.

On les utilisait pour coordonner une intervention, mémoriser une tâche précise ou suivre une correction. Mais on n’essayait plus d’y faire tenir la raison d’être d’un sujet, toutes les décisions passées et les informations utiles à un agent.

À la place, chaque responsable de sujet maintenait un espace de contexte dans leur outil documentaire commun. On y trouvait les objectifs, les décisions, les sources, les risques, les questions ouvertes et les liens vers ce qui existait déjà : tickets, devis, retours du support, comptes rendus de rendez-vous, tests et parfois anciennes discussions retrouvées.

Il n'était pas nécessaire de tout recopier. Il fallait surtout que quelqu'un puisse comprendre pourquoi une décision avait été prise, où vérifier une information et qui pouvait la changer.
Le premier document d'Alice avait servi d'exemple.
Bob avait repris la méthode, puis les développeurs, puis plusieurs personnes du support, du commerce et du déploiement. Beaucoup avaient découvert, comme Alice, qu'ils pouvaient passer une heure à expliquer oralement à ChatGPT des choix qu'ils auraient mis des jours à formaliser seuls.
Tout ce qui avait été dans la tête des gens n'était pas devenu miraculeusement accessible. Mais une partie des raisons, des exceptions et des savoirs tacites commençait à l'être.
Le télétravail avait rendu ce mouvement plus naturel. Depuis le Covid, Malorean avait déjà appris à écrire davantage pour que des collègues répartis dans toute la France puissent travailler sans attendre une conversation de couloir. L'IA avait accéléré l'extraction et l'organisation des connaissances. Elle n'avait pas inventé le besoin de les partager.

7

Alice ouvrit une page intitulée *Décisions Produit — règles communes*.
— On a encore eu une confusion la semaine dernière, expliqua-t-elle à Bob. La documentation d'une équipe utilisait « validé » pour dire « revu techniquement ». Celle du support l'utilisait pour dire « comportement confirmé avec le client ».
Bob soupira.

— Le mot n’a jamais voulu dire la même chose pour tout le monde.
— Justement. J’ai ajouté un renvoi vers la décision qu’on avait prise après l’histoire des lots.
Quelques semaines plus tôt, Lucas et Thomas avaient failli laisser coexister deux comportements incompatibles entre la traçabilité et l’expédition. Leurs changements avaient été arrêtés pendant les tests. Depuis, les équipes avaient appris à se méfier des termes qui semblaient évidents dans un module mais changeaient de sens ailleurs.
Le contexte avait aidé. Il avait aussi créé de nouveaux problèmes.
Certaines pages n’étaient plus à jour. Des décisions provisoires avaient été copiées comme s’il s’agissait de règles définitives. Un agent avait un jour ressorti une ancienne hypothèse commerciale comme si elle engageait l’entreprise.
Ils avaient dû imposer une discipline : chaque contexte indiquait son responsable, ses sources, ses décisions confirmées et les points encore incertains. Les changements structurants devaient être tracés ; les informations sensibles n’étaient pas accessibles indistinctement à tous les outils.
— On aurait pu éviter de multiplier les pages, remarqua Bob.
— Peut-être. Mais avant, les contradictions existaient aussi. Simplement, elles étaient dans nos têtes et personne ne pouvait les voir.
Alice relut la dernière modification de son collègue.

— Je valide le principe. Pas la formulation sur les clients allemands : on ne sait pas encore assez de choses.
Bob corrigea la ligne.
Le travail de Garant ressemblait souvent à cela. Non pas approuver chaque écran, mais empêcher qu'une hypothèse devienne une règle par inadvertance.

Dans l'autre équipe, Thomas venait de terminer un échange avec un développeur arrivé récemment.
Il lui avait consacré une heure sur le code historique de l'ERP. Le développeur avait utilisé un assistant pour comprendre un ancien mécanisme de paie, puis avait demandé pourquoi une exception semblait contredire la règle générale.
Thomas avait retrouvé la raison de cette exception dans une ancienne livraison.
— Si tu te contentes du résumé, tu risques de croire qu'il faut nettoyer le code, dit-il. Mais avant de nettoyer, il faut comprendre ce qu'on s'apprête à enlever.
— Tu pourrais mettre ça dans le contexte de l'équipe ?
— C'est ce que je fais.
Thomas partagea la nouvelle note. Il y avait ajouté la source, l'explication et deux cas de test.
Il n'avait toujours pas confié la production de ses modifications techniques aux agents comme Lucas le faisait. Son scepticisme n'avait pas disparu. Pourtant, il utilisait volontiers les nouveaux espaces de documentation et

travaillait davantage avec les autres pour transmettre son expérience.

Vivien, qui avait suivi l'intégration du nouveau développeur, appela Thomas en visioconférence un peu plus tard.

— Comment ça se passe ?

— Bien. Il est curieux. Il pose des questions avant de supprimer des choses.

— C'est rassurant.

— Et j'ai enfin pris le temps d'écrire pourquoi certains contrôles existent.

Vivien sourit.

— Tu as donc changé d'avis sur l'IA ?

— Non. J'ai changé d'avis sur le fait de garder ces explications pour moi.

Il marqua une pause.

— En revanche, il faut du temps pour ça. Si on me demande d'accompagner les autres et qu'on remplit quand même toutes mes journées avec des développements, ça ne marchera pas.

— J'en ai parlé à Hubert. On a réservé du temps d'accompagnement dans la planification. Ce n'est pas une activité qu'on fera seulement quand tout le reste sera terminé.

Vivien savait de quoi il parlait. Pendant des années, il avait vu des équipes annoncer qu'elles formeraient les nouveaux « dès qu'elles auraient un peu de temps ». Ce moment n'arrivait jamais.

Ils discutèrent encore des besoins du développeur, puis Vivien ferma sa visioconférence.
Il n'avait pas ouvert le tableau des livraisons une seule fois.

À l'heure du déjeuner, Hubert demanda à Vivien de le rejoindre en appel.
— J'ai une question, commença-t-il. Combien de temps passes-tu aujourd'hui à faire ce que tu faisais l'an dernier ?
Vivien réfléchit.
— Pas assez pour justifier mon ancien poste à temps plein, si c'est ce que tu veux savoir.
Hubert ne répondit pas immédiatement.
Vivien avait longtemps été le point de passage de deux équipes techniques. Il gérait le delivery, les dépendances, une partie des urgences et les discussions entre développeurs et responsables de métier. Quand l'organisation avait été moins autonome, son rôle occupait largement ses journées.
Ce n'était plus le cas.
Les sujets avaient un responsable clairement identifié. Les développeurs prenaient une grande partie des décisions locales. Les informations circulaient sans qu'il les retransmette lui-même. Les outils préparaient les suivis et signalaient les dépendances, même s'il fallait encore vérifier ce qu'ils affirmaient.
Les problèmes humains, les conflits et l'accompagnement n'avaient pas disparu. Vivien aidait encore à les traiter. Mais

ils n'avaient plus besoin d'être concentrés dans un poste dédié à la supervision quotidienne du delivery.

— Ça ne me dérange pas d'avoir moins de réunions inutiles, reprit-il. Au contraire. Mais je n'ai pas passé toutes ces années à apprendre à gérer des équipes pour devenir quelqu'un qui vérifie si un agent a bien rempli un tableau.

— Je ne te le demanderai pas, répondit Hubert.

Ils avaient préparé une transition. Les décisions techniques importantes revenaient aux Garants concernés. Les référents d'accompagnement disposaient de temps identifié. Hubert gardait la responsabilité des choix de carrière et des difficultés humaines qui ne pouvaient pas être laissées à un simple document. Vivien assurerait le passage de relais sur les derniers sujets de coordination.

Tout cela n'était pas parfait. Ils avaient dû reprendre une situation où deux personnes pensaient que l'autre suivait un collaborateur en difficulté. Une équipe avait aussi attendu trop longtemps un arbitrage parce que personne n'avait désigné son responsable.

Chaque fois, ils avaient corrigé les responsabilités et la manière de les rendre visibles.

— Il y a une chose que je ne veux pas, dit Hubert. C'est qu'on fasse disparaître ton poste et qu'on prétende que le travail que tu faisais n'avait aucune valeur.

— Moi non plus.

— Alors il faut qu'on trouve où cette expérience peut nous servir maintenant.
Vivien regarda sa tasse.
— Je crois que j'ai une idée.

Quelques jours plus tôt, sa femme lui avait reparlé de l'Allemagne.
Elle était allemande. Depuis quelque temps, ils discutaient de la possibilité de s'y installer. Pas dans l'urgence : ils avaient leur vie en France, leurs habitudes, leurs amis. Mais cette perspective revenait régulièrement, avec de plus en plus de détails.
Vivien avait toujours repoussé la question.
Il aimait trop son travail chez Malorean. Il connaissait les équipes, leur histoire, le logiciel et les clients. Partir lui aurait donné l'impression d'abandonner quelque chose qu'il avait contribué à construire.
Pourtant, il aimait aussi les défis. À ce stade de sa carrière, il se surprenait à avoir envie d'en relever un nouveau.
Le samedi précédent, ils avaient regardé des annonces de logements en Allemagne sans vraiment se l'avouer. Puis sa femme avait refermé l'ordinateur.
— Tu sais, dit-elle, on n'est pas obligés d'en parler pendant encore dix ans.
Vivien avait ri.
Mais le lundi, il y pensait toujours.

— L'Allemagne ? répéta Hubert.

Vivien acquiesça.

— On en parle depuis des années dans les réunions stratégiques. On a quelques clients là-bas, mais ce sont surtout des filiales de groupes français. On n'a pas encore vraiment appris à vendre notre ERP à des entreprises allemandes.

— C'est vrai.

— Et tout le monde se dit qu'on pourra y aller quand le produit sera prêt. Sauf qu'on ne saura jamais ce qui manque tant qu'on n'aura pas parlé sérieusement aux entreprises sur place.

Hubert l'écoutait avec attention.

— Tu voudrais t'occuper du développement du marché ?

— Oui. Pas en prétendant que je suis devenu expert du marché allemand pendant la nuit. Je connais notre produit, nos contraintes, nos équipes et la façon dont les clients le déploient. Je sais réunir des personnes autour d'un problème compliqué. Il faudra apprendre le reste.

— Tu as déjà fait du commerce ?

— Pas comme Julie ou Philippe. Et je n'ai pas l'intention de leur expliquer leur métier. Je voudrais travailler avec eux, avec le déploiement et avec nos spécialistes métier pour comprendre les besoins allemands, puis voir si on est capables d'y répondre.

Vivien hésita.

— Ma femme est allemande. On parle depuis quelque temps d'aller vivre là-bas. Ce serait une occasion de faire quelque chose dont j'ai vraiment envie, sans quitter Malorean.

Hubert resta silencieux un moment.

— Tu sais que ce n'est pas un simple projet de traduction ?

— Évidemment. L'ERP touche à des règles et à des processus qui varient selon les pays. Traduire les écrans ne suffit pas. Il faut savoir ce qu'on peut réellement promettre et ce qu'on devra développer, adapter ou refuser.

— Et on peut se tromper.

— On se trompera probablement sur certaines choses. Je préfère le découvrir avec des clients qu'après avoir investi un an dans des fonctionnalités dont personne ne veut.

Hubert regarda sa liste de projets.

Pendant longtemps, ils avaient parlé de l'Allemagne comme d'une ambition pour plus tard. Il fallait d'abord finir les sujets en retard, recruter davantage de développeurs, libérer du temps pour adapter l'ERP, préparer les démonstrations et former les équipes.

Certaines de ces difficultés n'avaient pas disparu. Mais plusieurs avaient changé d'échelle.

La préparation d'une démonstration demandait moins d'efforts. Les variantes du produit pouvaient être testées sans mobiliser systématiquement des développeurs. Les équipes support et avant-vente produisaient plus facilement leurs documents et partageaient leurs connaissances. Les

développeurs ne manquaient plus autant de capacité pour explorer une piste limitée.
Cela ne signifiait pas qu'ils n'avaient plus de travail.
Cela signifiait qu'ils pouvaient choisir de consacrer une partie de leur énergie à autre chose qu'à absorber la file des demandes françaises.
— Prépare-moi une proposition de première étape, dit Hubert. Pas une promesse d'ouverture du marché. Des rencontres, les questions auxquelles on veut répondre, les compétences dont tu auras besoin et une manière de décider si on continue.
Vivien sourit.
— Un Mini PRD pour l'Allemagne ?
— Ne te moque pas. Si on arrive à cadrer un sujet technique de cette façon, on peut au moins essayer de formuler les hypothèses d'un projet de marché.
— D'accord. Mais je veux garder le droit de découvrir qu'on s'est trompés.
— Je compte bien là-dessus.

La semaine suivante, Hubert réunit Alice, Bob, Vivien, Julie et Philippe.
Il leur présenta la proposition sans commencer par demander combien de fonctionnalités il faudrait développer.
— On a maintenant une capacité que nous n'avions pas il y a deux mois. Je voudrais qu'on en consacre une partie à

comprendre ce que représenterait une vraie présence sur le marché allemand.

Julie fut la première à réagir.

— J'ai déjà des démonstrations en allemand que je peux préparer. On pourrait aussi contacter des prospects assez vite. Avec l'IA, on peut adapter les supports en quelques heures.

— Les supports, peut-être, dit Philippe. Mais un dirigeant allemand qui achète un ERP ne choisira pas simplement parce que notre présentation est dans sa langue.

Julie croisa les bras.

— Je n'ai pas dit ça.

— Je sais. Je dis juste qu'on devra comprendre comment ils choisissent leurs fournisseurs, qui ils consultent et de quelle confiance ils ont besoin. On ne connaît pas encore assez bien ce marché.

Vivien intervint avant que la discussion ne devienne un débat sur les outils.

— C'est justement ce que je voudrais apprendre. Julie, on aura besoin de ta capacité à préparer rapidement des pistes et des démonstrations. Philippe, on aura besoin de ton expérience des grands comptes et des discussions qu'on ne peut pas résoudre avec un joli écran. Et les personnes du déploiement connaissent les questions que les clients découvrent parfois trop tard.

Alice prit la parole.

— De notre côté, il faudra être très clairs sur les limites Produit. On peut explorer vite, mais pas promettre des fonctions réglementaires qu'on n'a pas validées.

Bob acquiesça.

— On va aussi devoir choisir les entreprises qu'on veut comprendre en premier. L'Allemagne n'est pas un type de client. Si on essaie de répondre à tout le monde, on recommencera les erreurs qu'on a appris à éviter ici.

Hubert regarda les visages sur son écran.

Il avait demandé à Vivien de proposer une démarche. Personne ne lui avait demandé de constituer immédiatement une équipe permanente, de réserver tout le développement ou de promettre une date d'ouverture commerciale.

Pourtant, plusieurs personnes avaient déjà commencé à identifier ce qu'elles pouvaient apporter.

C'était devenu une habitude chez Malorean.

— Très bien, dit Hubert. Alors voilà ce que je décide : on finance une première phase d'exploration. Vivien la porte. Elle devra nous dire quels problèmes allemands nous savons déjà résoudre, lesquels demandent une adaptation et lesquels nous ne voulons pas prendre en charge. Ensuite, on choisira si nous investissons davantage.

Il ajouta :

— Et personne ne se retrouve affecté à ce projet simplement parce qu'un outil lui a fait gagner du temps. On va regarder ce

que chacun peut réellement dégager sans abandonner les clients que nous servons aujourd'hui.
Vivien nota le dernier point. Il avait passé suffisamment d'années à gérer des plannings pour savoir qu'une capacité libérée sur le papier n'était pas toujours disponible dans la réalité.

Dans les jours suivants, il fit le tour de l'équipe avant-vente et déploiement.
Une partie de leur travail avait changé depuis que les démonstrations pouvaient être préparées à partir de configurations réutilisables et que les synthèses des échanges clients étaient plus faciles à produire. Deux personnes lui expliquèrent qu'elles passaient désormais moins de temps à reconstruire les mêmes supports et qu'elles aimeraient participer aux premiers entretiens en Allemagne.
— Je peux t'aider à préparer les questions sur le déploiement, proposa l'une d'elles. Et accompagner quelques rendez-vous à distance.
— Et tes clients français ? demanda Vivien.
— Je ne peux pas tout lâcher. Mais je peux dégager du temps si on l'inscrit dans notre planning.
Une autre personne, sollicitée sur plusieurs mises en service délicates, lui dit au contraire qu'elle n'avait aucune disponibilité.
Vivien n'insista pas.

Ils préparèrent une répartition provisoire : quelques créneaux réservés pour comprendre les attentes des entreprises allemandes, des rendez-vous avec les filiales déjà clientes lorsque cela serait utile, et aucune promesse de mobilisation permanente avant d'avoir mesuré l'effort réel.
Le gain de temps ne s'était pas transformé tout seul en croissance. Il fallait encore décider d'en faire quelque chose et accepter de ne pas remplir immédiatement chaque heure retrouvée avec les demandes d'hier.

Le soir, Alice reçut un message de Lucas.
J'ai vu passer le projet Allemagne. On a déjà un mécanisme de configuration qui pourrait aider à tester certaines différences de processus. Je peux préparer un environnement exploratoire sans toucher au produit standard. Ça vous intéresse ?
Elle relut le message.
Deux mois plus tôt, une proposition comparable aurait probablement déclenché une discussion sur le fait qu'un développeur devait attendre qu'on lui confie un sujet.
Aujourd'hui, la première question était plus simple : était-ce utile à la stratégie et dans quel cadre pouvait-il avancer ?
Alice ouvrit le contexte de la première phase allemande.
Vivien l'avait déjà rempli avec les hypothèses, les responsables, les limites et les questions ouvertes.
Elle répondit :

Oui, si tu le cadres avec Vivien et si on ne transforme pas une hypothèse métier en comportement standard sans validation. Commence par ce qu'on veut apprendre, pas par ce qu'on sait construire.

Lucas répondit presque immédiatement.

Reçu. Je prépare un Mini PRD.

Alice sourit.

Elle se souvint du vendredi où il leur avait montré un module de congés développé en deux jours. Tout le monde avait alors essayé de comprendre comment il avait pu produire autant, aussi vite.

Ce soir, personne ne s'étonnait plus qu'il puisse construire une démonstration rapidement.

Le sujet était de savoir pourquoi ils en auraient besoin.

Vivien termina sa journée en partageant avec sa femme les dates des premières rencontres prévues avec des entreprises en Allemagne.

Il n'avait encore signé aucun contrat. Malorean n'avait pas ouvert de bureau, ni adapté son ERP à toutes les exigences du marché. Il faudrait découvrir les usages locaux, vérifier les contraintes juridiques et réglementaires, apprendre à convaincre des entreprises qui ne les connaissaient pas.

Il y avait beaucoup plus d'inconnues que de certitudes.

— Ça te fait peur ? demanda sa femme.

— Un peu.

— Tu avais l'air tellement sûr de toi quand tu coordonnais tes équipes.

Vivien sourit.

— J'avais passé des années à apprendre ce métier.

— Et maintenant ?

Il regarda le document qu'il avait commencé à rédiger pour sa première visite.

Des questions. Des hypothèses. Des noms de personnes à rencontrer. Une première liste de choses qu'il ne savait pas.

— Maintenant, répondit-il, j'ai de nouveau quelque chose à apprendre.

Elle s'approcha pour regarder l'écran.

— Et nous, dans tout ça ?

— On va pouvoir parler sérieusement de notre déménagement.

Elle posa la main sur son épaule.

Vivien repensa à son ancien tableau des livraisons, à ses réunions du lundi, à tout ce travail qui lui avait permis de tenir les projets ensemble. Il l'avait aimé. Il ne regrettait pas ces années.

Il ne regrettait pas non plus d'avoir accepté que les équipes n'aient plus besoin de lui de la même manière.

Sur son écran, le premier rendez-vous allemand était confirmé.

Pour Malorean, ce n'était encore ni une conquête ni un succès.

C'était un départ.

12 — Conclusion

Des métiers moins étanches, des responsabilités plus visibles

Les métiers ne délimitent plus exactement ce que chacun peut faire. Un développeur peut explorer un besoin, un PM peut construire, un commercial peut prototyper. L'IA ouvre des passages auparavant réservés à d'autres spécialités. Elle ne rend pas pour autant leurs expertises interchangeables.

Nous ne sommes plus responsables seulement de ce que nous exécutons nous-mêmes. Nous pouvons porter un sujet qui traverse plusieurs fonctions, tandis que les Garants veillent à la cohérence de leurs domaines. La capacité d'agir circule ; la responsabilité ne doit pas se dissoudre.

Le risque est double : continuer à dire « ce n'est pas mon métier » ou supposer « puisque je peux le faire, je peux décider seul ». Ni la défense des territoires ni leur disparition ne suffisent à organiser le travail.

L'autonomie repose sur un contexte partagé

Plus de liberté n'a de sens que si chacun dispose de ce qui lui permet de juger. Sans contexte partagé, une décision peut sembler juste localement et contredire un besoin, un choix ou une contrainte plus large.

Documenter n'est donc pas empiler des comptes rendus. C'est rendre accessibles le problème à résoudre, l'objectif, les

hypothèses, les raisons des décisions, les limites et les règles d'escalade. Formaliser ces éléments permet d'agir sans devoir retrouver à chaque étape la personne qui détient l'information. Le contexte devient un outil de décision, pour les personnes comme pour les agents.

C'est là que les Garants changent de posture. Ils n'inspectent pas toutes les initiatives : ils rendent le cadre compréhensible, veillent à sa cohérence et interviennent quand une décision le dépasse. Sans garde-fous, la liberté peut tourner au chaos ; si tout remonte pour validation, elle n'est qu'une promesse.

Le management évolue lui aussi. Une partie du suivi et de la coordination peut être prise en charge autrement. Mais il reste des personnes à accompagner, de l'expérience à transmettre, des conflits à résoudre et des arbitrages à assumer. Ces responsabilités ne disparaissent pas avec les tâches qui leur donnaient autrefois leur forme.

Faire davantage de ce qui compte

La tentation sera grande de mesurer cette transformation au nombre de fonctionnalités produites ou d'heures économisées. L'IA peut effectivement nous faire produire davantage. Mais remplir immédiatement le temps gagné avec de nouvelles tâches reviendrait à accélérer l'ancien fonctionnement sans en questionner la finalité.

Nous pouvons aussi consacrer cette capacité à mieux comprendre les besoins, vérifier nos hypothèses, améliorer ce

qui existe et renoncer à ce qui ne crée pas de valeur. Rien de cela ne se produit automatiquement. Il faut choisir où investir l'attention et les moyens libérés.

Vivien, chez Malorean Corp., ne découvre pas seulement que son ancien travail occupe moins de place. Il découvre qu'il peut mettre son expérience au service d'un autre problème, sans garantie de réussite. Le progrès ne tient pas seulement à ce qui a été automatisé, mais à ce que cette transformation rend possible.

Au début du récit, tout le monde s'étonnait de voir Lucas construire si vite. À la fin, la question est devenue : pourquoi construire, pour qui et avec quelles conséquences ?

L'enjeu de l'IA n'est pas de nous faire faire toujours plus. C'est de nous donner les moyens de faire davantage de ce qui compte vraiment.