



Design delivery

**La delivery à l'ère de l'IA:
ce qui se déplace et ce qui
demeure.**

Jean Mauvais - Sr Product Designer, Doctolib
Raphaël Lorand - Head of Design, Doctolib

Introduction

1. Synthèse exécutive
2. Périmètre du rapport
3. Panel d'experts consultés

Insights clés

1. L'accélération déplace la friction, elle ne l'efface pas
2. L'autonomie du designer: une responsabilité à clarifier
3. Le design system devient un système d'exécution
4. Bifurcation du rôle: une spécialisation qui s'installe
5. Une transformation des outils mais aussi culturelle

Quatre patterns d'implémentation observés

1. La correction design en autonomie (Design QA direct)
2. La spécification et le merge depuis Slack
3. La maintenance du design system par les designers
4. Les agents métier pour la delivery

Leviers et recommandations

1. Réduire la barrière d'entrée
2. Renforcer la gouvernance qualité
3. Investir dans l'intégration design system
4. Organiser l'apprentissage et les career paths
5. Piloter le changement et la charge mentale
6. Recentrer l'effort du designer sur ce qui compte vraiment

Perspectives à 12–18 mois

Plan d'action suggéré

Conclusion

Introduction

Introduction

Synthèse exécutive et description de la recherche menée.

Périmètre du rapport

Ce rapport synthétise les retours de plus de 18 professionnels du design, du produit et de l'ingénierie sur l'évolution de la delivery tech à l'ère de l'IA. Les entretiens ont été conduits sous forme de conversations semi-structurées d'une heure environ, entre janvier et mars 2026.

Ce document analyse l'impact organisationnel et stratégique de l'IA sur la delivery produit. Il ne constitue ni un comparatif d'outils, ni une évaluation de conformité sécurité. Il vise à éclairer les décisions de structuration, de gouvernance et d'évolution des rôles dans les équipes produit.

Synthèse exécutive

Ce que cette étude révèle n'est pas que l'IA change les outils, c'est qu'elle déplace la question de la responsabilité.

Quand n'importe qui peut shipper, la vraie question devient: qui détient la qualité?

Cette étude apporte des réponses concrètes, issues d'organisations qui ont déjà commencé à y répondre.

Les signaux convergent: les designers gagnent en capacité d'action jusqu'à contribuer directement en production via différents nouveaux outils: des IDE augmentés, des agents conversationnels et des outils de génération assistée.

Cette transformation réduit des frictions historiques (handoffs, itérations lentes) et augmente le rythme d'expérimentation. **Mais elle fait apparaître de nouveaux problèmes: gouvernance, qualité, responsabilité, montée en compétences, et dans les environnements régulés, sécurité et conformité.**

L'IA augmente la vitesse de delivery mais la performance durable dépend de la capacité de l'organisation à cadrer les scopes des équipes, contrôler la qualité et s'assurer que les fondations, le design system et la documentation tiennent à l'échelle.

Quatre grands patterns d'implémentation ont émergé sur le terrain, allant de la correction design autonome à l'usage d'agents métier pour la delivery complète. Ils permettent aux organisations de calibrer leur niveau d'autonomie selon leur maturité.

Panel d'experts consultés

Nous avons interrogé plus de 18 professionnels aux profils et contextes variés, issus du design, du produit et de l'ingénierie.

Le panel couvre des environnements de maturité et de contraintes très différents: conseil, startups, scale-ups, grandes organisations technologiques, ainsi que des secteurs régulés comme la santé, la finance.

La sélection a privilégié la diversité des contextes organisationnels sur la représentativité statistique. Les verbatims cités dans ce rapport sont représentatifs des tensions récurrentes identifiées, et non de cas isolés.

Plateformes & grandes organisations



Guillaume Perchais
Senior PM @Spotify



Caroline Caltagirone
Lead AI Design Ops & Design Manager @Malt

Secteurs réglementés



Laure Boutmy
Product Designer @Alan



Tiphaine Mazzetto
Product Designer @HarfangLab



Lisa Genovese
Product Designer @Doctrine



Louis-Eric Maucout et son équipe
Head of Design @Doctrine



Brice Burty
Product Manager @STid



Laurent Moulouquet
Research & Design lead @ Dedalus

Fintech & finance durable



Marion Echevard
Product Design Lead - Platform @Qonto



Rhys Wallace
Sr Product Designer @Qonto



Matthieu Lerat
Lead Product Designer, ex @Qonto



Sally Azar
UI/UX Designer @Weefin



Théo Lavaux
Sr Front-end Developer @WeeFin

Startups



Maxime Podgorski
Design Engineer & co-founder @Lyse



Benjamin Kaminski
Product Designer & Founder @Ryvka

Industrie & Data



Alena Torgonskaya
Product Designer @Metroscope



Camille Diconstanzo
Head of Product Experience @Dawex

Conseil



Julien Mosciatti
Senior Service Designer @PWC

Insights clés

Les points suivants ne sont pas des prises de position. Ils ressortent comme des tensions récurrentes dans les retours terrain.

1

L'accélération déplace la friction, elle ne l'efface pas.

La réduction du temps entre itération et production est nette, notamment sur la QA design et les petits ajustements UI. Des tâches qui durent des semaines passent à quelques heures. Moins de dépendances, plus de continuité.

J'ai réalisé qu'on avait accumulé un backlog énorme de petites améliorations UI, des tickets que personne n'avait touchés, certains là depuis plus d'un an. On a liquidé 35 tickets en quelques jours avec Cursor.



Alena Torgonskaya
Product Designer @Metroscope

Mais la friction se déplace. Là où le ralentissement se situait dans l'implémentation, il se situe désormais dans la revue et la gouvernance. Les PR augmentent en volume, et sans automatisation adaptée, les gains de vitesse s'annulent. Les équipes les plus lucides anticipent un glissement du rôle engineering vers davantage de contrôle qualité et d'orchestration.

Chez Spotify, tout le monde a des KPI IA et lance ses propres initiatives. Chacun pousse ses trucs dans sa bulle. Maintenant, on doit tout re-centraliser et rationaliser sans créer de conflits



Guillaume Perchais
Senior PM @Spotify

2

L'autonomie du designer: une responsabilité à clarifier

Quand des non-ingénieurs poussent du code, la question n'est pas seulement de savoir si c'est possible, mais ce qui est acceptable et sous quelles conditions. Les organisations avancées encadrent l'autonomie avec des périmètres progressifs, des règles explicites et une responsabilité clairement définie en cas d'incident.

Quand je travaille sur une fonctionnalité, j'utilise plusieurs agents et outils pour planifier, rédiger des user stories, cadrer le périmètre, identifier les impacts potentiels sur d'autres parties du produit. Si un changement est trop important, je ne le fais pas.



Alena Torgonskaya
Product Designer @Metroscope

Sans cadre, le risque principal est l'ambiguïté.

Trois questions restent sans réponse dans les organisations non préparées:

- **Périmètre d'intervention:** quelles contributions sont autorisées sans revue humaine ?
- **Charge de review:** avec plus de contributeurs, le backlog de PR peut exploser.
- **Responsabilité en cas d'incident:** en environnement régulé, l'absence d'ownership clair est un risque de conformité.

3

Le design system devient un système d'exécution

L'IA amplifie un problème connu: le décalage entre design et code. Lorsque des prototypes sont générés rapidement, la dérive apparaît facilement: composants non alignés, tokens mal mappés, règles implicites oubliées. Dans les produits complexes, la logique métier n'est pas toujours capturée dans les outils de design, ce qui rend certaines sorties rapides difficilement exploitables et fait émerger une dette invisible.

La vitesse de génération n'a de valeur que si le design system tient.

Dès qu'un composant manque de spécification précise (UX writing patterns, cas d'usage, do & don'ts, règle métier associée, pattern d'utilisation) l'IA comble le vide par une approximation plausible mais incorrecte. Dans ce cadre, un design system mal documenté devient un générateur de dette, et la correction post-merge coûte plus cher que l'itération manuelle initiale.

“Un agent peut gérer l'exécution technique à notre place. L'intention reste du côté du designer. Résultat ? On gagne du temps pour se concentrer sur ce qui compte vraiment: la qualité et la vision produit.”



Maxime Podgorski
Design Engineer @Lyse

4

Bifurcation du rôle de designer: une spécialisation qui s'installe

Deux trajectoires se distinguent: des profils orientés delivery et technique, capables d'implémenter et d'optimiser les workflows, et des profils orientés recherche, vision et structuration de problèmes complexes. Tout le monde ne suivra pas la même pente ; les organisations qui réussissent reconnaissent cette diversité et construisent des parcours cohérents plutôt que d'imposer un modèle unique.

“Je me pose vraiment cette question: quelle sera l'expertise fondamentale qui restera ? Qu'est-ce qui subsiste quand tout est automatisé ? En design, le delight n'est pas délivré par l'IA qui elle ne fait que reproduire les patterns standardisés.”



Guillaume Perchais
Senior PM @Spotify

Mais entre ces deux pôles, un troisième profil s'impose progressivement celui qui réduit le gap plutôt que de choisir un camp: le design engineer. Il opère dans les deux langages à la fois. Il traduit l'intention design en implémentation, et ramène les contraintes techniques dans la conversation de design avant que les décisions soient figées.

Son terrain naturel est le design system. Il ne se contente pas de consommer les composants: il les maintient, les fait évoluer, les documente pour qu'ils soient exploitables aussi bien dans Figma que par un agent IA générant du code. Dans un contexte où l'IA produit de plus en plus de surface UI, le design system devient l'interface de contrôle qualité et le design engineer en est le gardien opérationnel.

Ce rôle exige un bilinguisme professionnel rare. L'IA l'amplifie, mais ne le remplace pas: le jugement qui rend ce profil stratégique reste humain.

“Et si UX et UI laissaient place à deux profils distincts: le design engineer, ancré dans l'exécution, et le design researcher, connecteur entre équipes produit ?”



Maxime Podgorski
Design Engineer @Lyse

5

Une transformation des outils mais aussi culturelle

Les outils évoluent vite. Mais la différence entre les équipes qui réussissent et celles qui stagnent se joue ailleurs: dans la capacité à construire un cadre culturel: documentation partagée, apprentissage collectif, gestion du changement. L'adoption technique n'est que la partie visible. Deux modèles d'organisation se distinguent sur le terrain.

Chez Alan, le programme « Everyone Can Build » illustre ce que signifie une transformation culturelle assumée: designers et PMs contribuent directement en production, encadrés par une équipe enablement dédiée qui définit les garde-fous, les standards de revue et les règles de contribution. L'objectif n'est pas de rendre tout le monde développeur, c'est de rendre l'autonomie safe à l'échelle.

Chez Qonto, une équipe Platform dédiée a été constituée pour définir les building blocks qui permettent aux designers de shipper avec l'IA. L'approche est centrée sur la réduction de la friction d'entrée (en particulier l'aversion au terminal) et la standardisation des workflows.

Notre outil principal, c'est Cursor. C'est un outil visuel, un vrai IDE, pas un terminal complexe. C'est l'outil avec le moins de friction, on évite la peur d'utiliser le terminal.



Rhys Wallace
Sr Product Designer @Qonto

À l'inverse, quand le cadre culturel fait défaut, la transformation se fragmente: chaque équipe développe ses propres pratiques dans sa bulle, l'apprentissage ne circule pas, et les initiatives IA multiplient les outils sans capitaliser sur les expériences des uns et des autres. La vitesse individuelle augmente, mais la cohérence collective se dégrade.

Ce qui distingue les organisations qui réussissent cette transition n'est pas la sophistication de leur stack, c'est leur capacité à traiter l'apprentissage comme un actif collectif plutôt qu'une compétence individuelle.

Patterns d'implémentation observés

Au-delà des tendances générales, quatre modes concrets d'action émergent sur le terrain. Ils ne sont pas exclusifs, les équipes les plus avancées les combinent.

Les patterns 1 et 2 sont accessibles dès les premières semaines, avec peu de prérequis techniques. Les patterns 3 et 4 supposent des fondations plus solides: un design system structuré, une gouvernance de review en place, et au moins un profil avec une aisance technique dans l'équipe.

Un bon point d'entrée est de commencer par le pattern qui correspond au problème le plus douloureux aujourd'hui (backlog de QA, dette design system, ou complexité d'implémentation) et de monter en complexité à mesure que la confiance s'installe.

Patterns observés

Une autonomie sur la design debt et la design QA

Les designers prennent possession de la dette design et de la qualité visuelle. Des corrections qui prenaient des semaines se règlent en quelques heures, avec une forte réduction des dépendances sur les développeurs.

Point de vigilance: les PRs doivent atteindre un très haut niveau de qualité, sous peine d'exploser le temps de review.

2

La spécification et le merge depuis Slack

Très faible barrière technique, interface familière, impact direct et contrôlé. Permet de vider une dette design significative sans onboarding lourd.

Point de vigilance: charge de review accrue pour les développeurs si le volume de PRs monte rapidement.

Exemple: Spotify a développé son propre pipeline "Slack to PR" pour éviter toute complexité d'installation locale.

3

La maintenance du design system par les designers

Les designers deviennent propriétaires de la livraison et de la QA du design system, dans une logique test-and-learn.

Point de vigilance: fonctionne uniquement avec une gouvernance et des processus de review clairement définis en amont.

Exemple: l'équipe Platform de Qonto a défini les building blocks pour que les designers contribuent directement au DS.



Les agents métier pour la delivery

Les agents gèrent les spécificités techniques d'implémentation, libérant le designer pour l'intention et la qualité. De-risking en temps réel via linters automatiques.

Point de vigilance: nécessite une configuration initiale par des profils techniques, et une consommation de tokens à anticiper.

Exemple: chez Lyse, le design s'opère directement dans Cursor avec des spécifications structurées vérifiées par des agents.

Je travaille maintenant directement sur une branche de production, avec des agents IA qui jouent le rôle de CTO et challengent mes idées.



Camille Diconstanzo

Head of Product Experience @Dawex

Leviers et recommandations

Les insights précédents posent le diagnostic. Les six leviers ci-dessous traduisent ces tensions en décisions concrètes, sur l'organisation, les processus et les priorités d'investissement.

Trois d'entre eux sont prioritaires parce qu'ils conditionnent tous les autres: sans onboarding fonctionnel, personne ne contribue ; sans gouvernance qualité, la vitesse devient un risque ; sans design system solide, l'IA génère de la dette plus vite qu'on ne peut la corriger.

Les leviers 4, 5 et 6 opèrent en parallèle et créent les conditions culturelles de la durabilité.

Réduire la barrière d'entrée

Rendre l'onboarding simple et reproductible: scripts, environnements préconfigurés, guides clairs. L'objectif n'est pas la performance technique c'est que chaque designer puisse contribuer sans dépendre d'un intermédiaire. L'onboarding réussi se mesure à l'autonomie, pas au temps passé en setup.

Envisager des environnements cloud/distants pour les équipes distribuées ou les grandes organisations, ils suppriment la complexité de setup local et réduisent la charge de support.

2

Renforcer la gouvernance qualité

Mettre la qualité par défaut grâce aux contrôles automatiques: linters, CI, boucles de review multi-étages où l'IA corrige jusqu'à conformité avant revue humaine.

La gouvernance qualité ne se résume pas à des outils, elle se résume à des responsabilités claires. Qui a le droit de merger? Qui corrige quand ça casse? Ces questions doivent avoir une réponse avant le premier incident, pas après.

Définir des SLA de review explicites (délais d'approbation, critères d'acceptation) pour maintenir la prévisibilité du pipeline même quand le volume de PRs augmente.

3

Investir dans l'intégration design system

Renforcer le lien design et code, formaliser les règles métier, rendre le drift visible en continu. Quand les règles sont absentes ou implicites, l'IA comble les lacunes par des approximations. La vitesse gagnée en amont se perd en corrections post-merge.

Chaque niveau atomique doit être spécifié composants, tokens, spacing avec des Do & Don'ts explicites, des use cases et des patterns contextualisés. C'est ce qui transforme le design system en rails: l'IA s'y appuie pour suggérer, vérifier, corriger.

Plus les règles sont précises, plus l'accompagnement est efficace.



Organiser l'apprentissage et les career paths

Former, documenter et reconnaître les trajectoires hybrides. Ne pas imposer un modèle unique: certains profils iront vers la delivery technique, d'autres vers la recherche et la vision. Les deux ont de la valeur à condition que l'organisation le reconnaisse explicitement.

Mais la montée en compétences ne se décrète pas: elle se construit dans la pratique collective. Les équipes les plus avancées structurent des espaces d'échange dédiés channels Slack thématiques, sessions de partage régulières, bibliothèques de prompts et de workflows en commun. Moins pour former de manière top-down que pour capitaliser sur ce que chacun découvre, et éviter que l'apprentissage reste isolé.

La résistance au changement vient de l'anxiété: « Qu'est-ce que je vais devenir ? Est-ce que je vais m'en sortir ? ». Mais avec les bons outils et le bon accompagnement, ces peurs disparaissent.



Laure Boutmy
Product Designer @Alan



Piloter le changement et la charge mentale

Limiter la fragmentation outillage, clarifier la source de vérité, traiter les enjeux administratifs sérieusement (accès, offboarding, continuité). Quand l'information se disperse dans des environnements multi-outils, le contexte se perd et l'ownership devient flou.

L'IA ne remplace pas les designers, elle nous augmente. On étend notre périmètre tout en gardant ce qui nous est essentiel. Notre expertise nous maintient aux commandes de l'IA, pas l'inverse.



Matthieu Lerat
Lead Product Designer, ex @Qonto



Recentrer l'effort du designer sur ce qui compte vraiment

L'IA prend en charge les tâches répétitives et pose les premiers jets d'itération. Ce n'est pas une menace, c'est une opportunité rare: celle de récupérer du temps et du focus pour ce qui fait la valeur irremplaçable du designer.

Les détails qui font une expérience unique. Le goût. La perspective d'expertise qui transforme une interface correcte en quelque chose de mémorable. La capacité à sentir ce qui manque, ce qui sonne faux, ce qui peut être poussé plus loin.

Concrètement, cela suppose un choix organisationnel délibéré: réserver des plages de travail explicitement protégées de l'IA: revues de qualité perçue, sessions de critique de l'expérience globale, travail sur le langage visuel, où le designer exerce son jugement sans filet.

Sans ce cadre, le temps libéré par l'IA se remplit mécaniquement d'autres tâches de production, et la promesse de recentrage ne se matérialise jamais. Les équipes qui réussissent cette transition ne sont pas celles qui utilisent le plus d'outils, ce sont celles qui ont décidé explicitement à quoi sert le temps gagné.

On passe de créateurs à reviewers, et c'est en réalité plus puissant. On va évaluer ce que l'IA génère et avoir l'autorité de dire non, d'exiger mieux.



Louis-Eric Maucout, Head of Design @Doctrine

Perspectives à 12–18 mois

Cette section ne constitue pas un ensemble de prédictions certaines.

Elle synthétise les signaux faibles et les trajectoires probables qui émergent des entretiens, croisés avec les dynamiques d'adoption observées sur le terrain.

Signal 1: La review automatisée devient standard

Dans les 12 prochains mois, les multi-stage review loops, où un agent IA itère sur le code jusqu'à conformité avant intervention humaine devraient passer du statut d'expérimentation à celui de pratique courante. Le nouveau goulot ne sera plus la review elle-même, mais la gouvernance multi-agents: qui orchestre les agents intervenant à différentes étapes ? Qui a la responsabilité finale en cas de conflit ? La formalisation explicite des règles métier devient le prérequis de cette automatisation.

2

Signal 2: Les environnements cloud préconfigurés s'imposent

Le modèle "Slack to PR" de Spotify et d'autres préfigure la direction que prend l'outillage: zéro installation locale, contribution depuis une interface connue.

Pour les grandes organisations, les environnements distants préconfigurés vont devenir le standard pour onboarder des contributeurs non-techniques à grande échelle. Cette évolution soulève en contrepartie de nouvelles questions de sécurité et de contrôle des accès que les environnements régulés devront anticiper.

3

Signal 3: La bifurcation des rôles s'accélère

Les deux trajectoires (design engineer et design researcher/stratège) vont se cristalliser en carrières distinctes. Les organisations qui planifient cette bifurcation maintenant auront un avantage structurel. Le signal le plus radical vient de Qonto: Marion Echevard anticipe un futur à deux ans sans Figma, tout en code. Ce n'est pas une projection universelle, mais une invitation à questionner les hypothèses sur le rôle des outils de design actuels.

“Dans deux ans ? On n'utilisera peut-être plus Figma du tout. Tout en code. C'est un pari, mais fini de maintenir deux versions du design system.”



Marion Echevard, Product Design Lead - Platform @Qonto



Signal 4: Les design systems AI-natifs émergent

Certaines équipes ne documentent plus leur design system uniquement pour les designers et développeurs, elles le conçoivent pour qu'il soit consommé par des agents. Chaque composant, chaque token, chaque règle est documenté de façon interprétable par un modèle. Les équipes qui investissent aujourd'hui dans la rigueur de leur design system se donnent un avantage compétitif direct dans l'ère agentique qui se profile.



Signal 5: La vigilance sur la dette invisible

Avec l'accélération de la génération, un risque croissant émerge: l'accumulation de surface UI qui semble correcte techniquement mais qui dérive progressivement de la vision produit et de la logique métier. Cette dette est difficile à quantifier, elle n'apparaît pas dans les linters, elle se manifeste dans l'expérience utilisateur. Le drift monitoring et la mesure de la qualité perçue vont devenir des disciplines à part entière.

Quelques métriques de pilotage recommandées:

- Lead time moyen entre conception et mise en production
- Volume de rework post-merge
- Backlog de review et temps moyen d'approbation
- Taux d'adoption par profil (% de designers actifs)
- Drift rate du design system (% de composants déviants détectés lors des audits)

Plan d'action suggéré pour implémentation

Le plan ci-dessous organise la transformation en quatre phases progressives. Chaque phase est portée par un owner clairement identifié. L'objectif n'est pas la vitesse maximale, c'est la solidité des fondations à chaque étape.

Prérequis

Prérequis: identifier le sponsor de la transformation

Avant de lancer le premier audit, une question doit avoir une réponse: qui porte ce chantier ? L'expérience des équipes les plus avancées converge sur un point: les initiatives qui échouent ne manquent pas d'outils, elles manquent d'un owner avec suffisamment de légitimité pour trancher sur les tensions entre vitesse, qualité et périmètre. Ce sponsor n'est pas nécessairement le plus senior. C'est celui qui opère à l'intersection du design, de l'engineering et du produit, et qui a la confiance des trois.

Dans la pratique, trois profils endossent ce rôle selon les contextes:

- **Le Head of Design**, quand la transformation est portée par une vision de l'évolution du métier et de la qualité produit.
- **Le Design Ops Lead**, quand l'enjeu principal est l'outillage, l'onboarding et la standardisation des workflows.
- **Un Engineering Manager allié**, quand les questions de gouvernance qualité et de périmètre de contribution dominant.

Ce qui ne fonctionne pas: une initiative pilotée par enthousiasme individuel sans mandat explicite, ou distribuée entre plusieurs équipes sans arbitre clair. La transformation IA en design est un chantier organisationnel autant que technique, il a besoin d'un pilote nommé, pas d'une communauté d'intérêt.

Phase 0: Diagnostic

📅 Semaines 1–4

👤 Owner: Design Ops + Engineering Lead

- Audit des workflows actuels: identifier les blocages techniques, juridiques, humains.
- Cartographie des compétences et des gaps par profil.
- Revue de la maturité du design system: tokens, documentation, connexion Figma-code.
- Identification des quick wins et des risques de conformité (accès, sécurité, offboarding).

Phase 1: Pilote contrôlé

📅 Mois 2-3

👤 Owner: Design Ops + Platform/Eng

- Onboarding simplifié sur un périmètre restreint (2–3 designers volontaires).
- Mise en place des premières automatisations: linter, CI, boucle de review.
- Test d'un premier pattern d'implémentation (Design QA direct ou Slack to PR).
- Documentation continue des learnings et blocages rencontrés.

Phase 2: Consolidation des fondations

📅 Mois 3-6

👤 Owner: Design System Lead + Security

- Renforcement du design system: formalisation des règles métier, connexion Figma-code.
- Définition des SLA de review et des périmètres d'intervention par profil.
- Sécurisation des environnements et clarification des processus d'offboarding.
- Mise en place de la gouvernance qualité et des responsabilités en cas d'incident.

Phase 3: Déploiement et montée en puissance

📅 Mois 6-18

👤 Owner: Tous les leads

- Extension progressive à l'ensemble des équipes selon les learnings du pilote.
- Structuration des career paths: design engineer vs. researcher/stratège.
- Activation des métriques de pilotage.
- Points d'apprentissage réguliers, bibliothèques de prompts et workflows partagés.

Conclusion

L'IA rapproche intention et implémentation. Elle rend visibles les faiblesses structurelles autant qu'elle accélère l'exécution. Ce qu'elle déplace est réel: les tâches répétitives, les premiers jets, une partie de l'implémentation. Ce qu'elle ne remplace pas l'est tout autant: le jugement, le goût, la capacité à sentir ce qui manque.

Ce moment est rare. Chaque révolution technologique redéfinit ce que les métiers font, et surtout ce qu'ils arrêtent de faire. Pour le design, c'est une opportunité de revenir à l'essentiel: poser les bonnes questions avant de chercher les bonnes réponses.

Les organisations qui en tireront le meilleur parti ne seront pas celles qui accumulent les outils. Ce seront celles qui auront compris ce que l'IA a déplacé et utilisé cet espace pour révéler ce que leurs designers font de mieux.

Et peut-être que comme certains de nos intervenants le disent: dans deux ans, plus de Figma. Tout en code. Ce n'est pas une certitude, c'est une invitation à décider, maintenant, ce que vous voulez que vos designers fassent de leur temps quand les outils d'aujourd'hui n'existent plus.

Auteurs

Jean Mauvais

Raphaël Lorand

Merci à tous les participants ayant accepté de participer aux interviews

Merci aux communautés The Design Crew & Discovery Discipline pour leur participation à cette étude.