L’impact de l’intelligence artificielle générative sur la créativité des développeurs informatiques

Guy Parmentier
Univ. Grenoble Alpes, Grenoble INP, CERAG, 38000 Grenoble France
guy.parmentier@univ-grenoble-alpes.fr

Raphaël Teixeira
Univ. Grenoble Alpes, Grenoble INP, CERAG, 38000 Grenoble France
raphael.teixeira@univ-grenoble-alpes.fr

Ce travail va être publié en 2025 dans la Revue Française de Gestion

Résumé

De premières études indiquent que l’utilisation de l’IA générative a un effet sur la créativité des individus et des équipes dans les organisations. Pour analyser ce phénomène, nous étudions les effets de l’utilisation de l’IA générative sur la créativité des développeurs qui l’ont largement adoptée depuis 2021. Nos résultats indiquent une utilisation de l’IA générative très large pour soutenir les activités de développement qui va au-delà de la génération de code. Nous identifions deux modes relationnels Développeur-IA : l’IA comme assistant-développeur et l’IA comme compagnon du développeur, qui ont des conséquences différentes sur la créativité ainsi qu’en matière d’apprentissage.

Abstract

Initial studies suggest that the use of generative AI has an impact on the creativity of individuals and teams in organisations. To analyse this phenomenon, we examine the impact of the use of generative AI on the creativity of developers who have widely adopted it since 2021. Our results indicate a very broad use of generative AI to support development activities beyond code generation. We identify two modes of developer-AI relationships: AI as developer assistant and AI as developer companion, which have different implications for creativity as well as for learning.

Ce travail a bénéficié d'une aide de l’état gérée par l'Agence Nationale de la Recherche au titre de France 2030 portant la référence ANR-22-EXEN-0004.

L'impact de l'intelligence artificielle générative sur la créativité des développeurs informatiques

INTRODUCTION

L'arrivée récente d'outils d'intelligence artificielle (IA) générative a des conséquences importantes sur le travail en entreprise (Diallo, 2023). Les générations précédentes d'IA étaient conçues pour résoudre des taches spécifiques avec des règles prédéfinies alors que l'intelligence artificielle générative (IAG) est basée sur des modèles de langage entraînés sur de très grandes quantités de données (LLM), L'IAG est ainsi capable de générer de nouvelles données (textes, images, sons et vidéos) qui ne sont pas similaires à celles utilisées pour l'entraînement des LLMs. Ces IA effectuent donc des tâches de recherche et d'association d'informations en langage naturel qui n'étaient possibles jusqu'à ce jour qu'avec des humains, et comme l'accès à l'information par internet précédemment, elle risque de modifier fortement nos manières de rechercher et traiter l'information ainsi que nos processus cognitifs de synthèse et d'analyse. En tant que technologie dite "cognitive" (Davenport et Ronanki, 2021), l'IAG pourrait avoir des effets majeurs sur la créativité de ses utilisateurs (Jia *et al.*, 2024) et sur la créativité organisationnelle (Mikalef et Gupta, 2021 ; Vinchon *et al.*, 2023).

En 1982, Marvin Minsky déclarait que l'IA pourrait être capable de produire des textes aussi géniaux que Shakespeare d'ici 50 à 100 ans (Amabile, 2020). Les sciences informatiques s'intéressent depuis de nombreuses années à cette question de la créativité de l'IA (Lamers, 2023). Les chercheurs ont défini quels types de créativité pourraient être développés par l'IA (Boden, 1998) et des modèles informatiques d'IA créative (Mikalef et Gupta, 2021). Le débat est vif entre ceux qui considèrent l'IA comme "créative" et ceux qui la considèrent une "pseudo-créativité" (Runco, 2023). Cependant, le manifeste de Vinchon *et al.* (2023) appelle à développer des recherches sur la co-création avec l'IA plutôt que d'opposer la créativité supposée de l'IA à la créativité humaine. Des recherches essayent ainsi de déterminer quel serait l'effet de l'utilisation de l'IAG sur les utilisateurs, notamment dans les activités de résolution de problème (Raisch et Fomina, 2023) et la création de contenu (Doshi et Hauser, 2024 ; Ivcevic et Grandinetti, 2024 ; Lim *et al.*, 2023). Les résultats de premières études sont contradictoires, certaines études mènent à plus de créativité (Doshi et Hauser, 2024 ; Jia *et al.*,

2024) alors que d'autres à moins de créativité (Boussioux *et al.*, 2024). Par ailleurs, peu de recherches ont été menées sur l'impact de l'IAG sur la créativité des utilisateurs en milieu professionnel et sur les mécanismes qui induisent ses effets. Cette question est d'autant plus importante que l'IAG ouvre de nouvelles possibilités d'analyse et de création et pourrait avoir des impacts importants dans les fonctions d'entreprises, telles que les opérations clients, le marketing, l'ingénierie logicielle et la recherche et développement (Chui *et al.*, 2023).

Depuis 2021, les développeurs ont accès à des programmes spécialisés d'aide à la conception, ChatGPT et GitHub Copilot qui fournissent une solution de génération de code à partir de la description en langage naturel. GitHub Copilot intègre aussi de l'auto-complétion de codes, il propose la suite du code au fur et à mesure que les développeurs codent. Ces deux logiciels, basés sur des LLMs offrent de nouvelles possibilités d'action aux développeurs qui modifient leur manière de travailler. Une enquête de 2023 effectuée auprès de 500 développeurs sur GitHub révèle que 92% utilisaient déjà des outils basés sur l'IAG (GitHub et Inbal, 2023). Et les conséquences sont importantes, la programmation avec GitHub Copilot est 55,8% plus rapide (Peng *et al.*, 2023), mais peut aboutir à un manque de variété et une faible qualité des propositions de codes (Zhang *et al.*, 2023).

L'industrie informatique est considérée comme une industrie créative[1] dans la mesure où elle mobilise largement la créativité des développeurs pour créer des programmes originaux et résoudre les problèmes de développement de manière originale (Israel-Fishelson et Hershkovitz, 2022). La créativité est donc un des éléments clés de cette industrie et elle pourrait être fortement impactée par l'usage de l'intelligence artificielle générative dans la mesure où elle s'intègre dans les activités de création des développeurs.

Quelles sont les conséquences de l'utilisation de l'IAG sur la créativité des développeurs ? Quels sont les usages de l'IAG qui agissent le plus sur la créativité des développeurs ? La réponse à ces questions nous donne des enseignements sur les conséquences de cette nouvelle technologie sur cette activité de création afin d'anticiper ses effets sur l'industrie informatique. Ainsi à partir du témoignage de 14 développeurs qui utilisent les outils d'IA dans leur travail, nous en déduisons les modes relationnels les plus favorables à leur créativité et les dérives possibles de dépendance technologique et de désapprentissage.

[1] Voir la classification de l'UK DCMS (https://www.gov.uk/government/publications/dcms-sectors-economic-estimates-methodology/dcms-sector-economic-estimates-methodology#definitions).

I - CADRE THÉORIQUE

1. Intelligence artificielle et créativité

La créativité est une activité de production d'idées nouvelles, appropriées, utiles et faisables effectuée par un individu ou un petit groupe d'individus travaillant ensemble (Amabile, 1983). Depuis les années 80, la recherche a exploré le rôle majeur du contexte sur la créativité individuelle et collective (Woodman, Sawyer et Griffin, 1993). Au sein de ce contexte, la facilitation (Wróbel *et al.*, 2021) et les jeux (Agogué *et al.*, 2015) illustrent l'interaction entre humains, objets et méthodes dans le processus créatif, formant ainsi une structure socio-matérielle de la créativité. Il existe actuellement un nombre croissant de travaux sur la socio-matérialité de la créativité (Tanggaard, 2013) montrant par exemple que les objets (Chen, Chandler et Venkatesh, 2020) et les méthodes (Gillier et Bayus, 2022) influencent les processus créatifs. L'approche socio-matérielle de la créativité est ainsi un cadre intéressant pour analyser l'effet de l'IAG sur la créativité.

Les études sur l'effet de l'utilisation de l'IAG sur la créativité des utilisateurs sont encore à ce jour assez rares et contradictoires. Selon le type d'IA, et les compétences cognitives et émotionnelles, la créativité des utilisateurs pourrait être différente (Ameen *et al.*, 2022), indiquant ainsi une possible interaction entre ces variables. Les effets positifs de l'IAG sur la créativité des graphistes dépendraient de la possibilité de développer des relations collaboratives et empathiques avec l'IAG (Lim *et al.*, 2023), suggérant que la créativité des utilisateurs serait dépendante des relations Humain-IA. L'IAG aurait un effet positif sur la créativité que sur les téléconseillers les plus qualifiés (Jia *et al.*, 2024), ce qui suggère qu'un niveau minimum d'expertise est nécessaire pour développer sa créativité avec l'IAG. Néanmoins une étude menée au sein d'une plateforme de crowdsourcing montre que les idées des internautes générées sans l'IAG sont légèrement plus originales que les idées générées avec l'IAG. Cependant, ces dernières sont plus faisables et utiles, avec un potentiel de création de valeur plus important (Boussioux *et al.*, 2024). De plus, les internautes qui utilisent l'IAG avec des prompts successifs dépendants des réponses précédentes sont plus créatifs que les internautes qui font des prompts indépendants. Une autre étude menée auprès d'écrivains amateurs indique que l'utilisation de l'IAG produit des histoires plus originales, plus plaisantes, mais moins diversifiées (Doshi et Hauser, 2024). Toutefois ce sont les écrivains les moins créatifs au départ qui bénéficieraient le plus des suggestions de l'IAG.

Ces études montrent des résultats assez contrastés sur l'effet de l'utilisation de l'IAG sur la créativité. Selon la tâche à réaliser et le contexte, l'utilisation de l'IAG aurait des effets sur

l'originalité et l'utilité des idées. Toutefois cet effet dépendrait de la manière de dialoguer avec l'IAG et des compétences initiales. Il existe de plus peu d'études auprès de professionnels en activité, la majorité des études sont des expérimentations hors contexte professionnel ou avec des étudiants.

2. Les possibilités d'action de l'IAG sur la créativité

Dans une approche socio-matériel de la créativité, nous proposons de mobiliser la théorie des affordances (Gibson, 1977) pour analyser l'effet de l'usage de l'IAG sur la créativité. Le concept d'affordances a été fondé par James Gibson (1977) pour expliquer comment les individus perçoivent les possibilités comportementales d'un objet ou d'un environnement. D'après Jarzabkowski et Kaplan (2015), la matérialité d'un objet façonne et, en même temps contraint, un ensemble d'usages spécifiques. Ce concept a été utilisé pour étudier l'action des fonctions ou les possibilités d'action des outils technologiques. Les affordances d'un objet sont perceptibles quand l'objet est explicitement conçu pour provoquer une action (Gaver, 1991). Par exemple, pour ChatGPT, l'action possible la plus perceptible est celle de poser une question afin d'obtenir une réponse. Les outils technologiques ont aussi des affordances "cachées", des possibilités d'action qui ne sont pas directement perceptibles, mais possibles, même si leur ergonomie ne les suggère pas directement (Gaver, 1991). ChatGPT, étant basée sur un grand modèle de langage, peut être utilisée pour de multiples actions : rechercher une information, synthétiser, traduire, réduire, et corriger un texte par exemple, sans que ces activités soient explicitement suggérées par l'interface de l'outil[2]. Dans les activités créatives, le contexte apporte un cadre favorisant ou limitant les comportements créatifs (Amabile, 1988) dans lequel il y a mobilisation de méthodes ou d'outils pour faciliter la problématisation, l'idéation et la sélection des idées. Les affordances des outils et technologies utilisées en groupe ont une quintuple dimension : fonctionnelle, symbolique, politique (Rampa et Parmentier, 2024), cognitive (Hartson, 2003) et stratégique (Jarzabkowski et Kaplan, 2015). L'IA en tant que technologies cognitives utilisées au niveau individuel, dans des équipes ou déployée dans une organisation est donc susceptible de favoriser ou contraindre les activités créatives avec des affordances fonctionnelle, psychologique, cognitive, et politique.

[2] ChatGPT évoluant rapidement, il n'est pas exclu que l'interface ait changé depuis la réalisation de l'étude qui a eu lieu de février à août 2024.

L'IA générative permet de traiter de grandes masses d'information, d'obtenir des réponses à des questions spécifiques, de synthétiser de l'information ou d'identifier des catégories. Elle a donc en premier lieu une fonction instrumentale ouvrant de nouvelles possibilités d'action au niveau du traitement de l'information pour les individus et les collectifs. La créativité est basée sur la collecte et le traitement d'information pour identifier les problèmes et trouver des solutions (cf. les méthodes Creative Problem Solving et Design Thinking). L'analyse des affordances fonctionnelles de l'IAG consiste à examiner quelles sont les possibilités offertes par l'IAG et mobilisées dans les processus de création, et quelles sont les conséquences sur la créativité des individus et des équipes.

La créativité se base sur des processus cognitifs bien identifiés : la pensée divergente et convergente (Guilford, 1950), et la manière de traiter, combiner et synthétiser l'information avec l'association, l'analogie et la bissociation (Lubart, 2003). L'IAG en tant que technologies cognitives pourrait agir sur ces processus cognitifs et ainsi influencer la créativité. L'analyse de l'affordance cognitive de l'IAG consiste donc à examiner comment les utilisateurs de l'IAG s'en emparent pour modifier ou favoriser des processus cognitifs qui sont le socle de la créativité individuelle.

Le contexte du processus créatif dépend aussi des interprétations individuelles et collectives. La créativité est une expérience à la fois de pensée et de faire qui s'ancre dans l'action (Cummings *et al.*, 2015). Ainsi l'IAG pourrait être utilisé pour donner du sens à un problème ou même diminuer la difficulté perçue d'un problème. L'IAG aurait donc des possibilités d'actions sur l'état mental, les perceptions et les conceptions de l'utilisateur, des affordances psychologiques. Par exemple, la motivation intrinsèque dépend en grande partie du contexte dans lequel l'individu agit (Deci et Ryan, 1985) et elle a un fort impact sur la créativité (Amabile, 1988). La mise à disposition d'outil d'IAG pourrait donc agir sur la motivation à passer à l'action, car ils fournissent des outils avec de nombreuses possibilités pour créer du contenu. De même, l'utilisation de ces outils pourrait renforcer la confiance créative (Tierney et Farmer, 2002) en donnant l'impression que les possibilités offertes ouvrent le champ des possibles. À l'inverse, travailler avec une IAG capable de générer beaucoup de contenus peut faire douter les utilisateurs de leur capacité à créer. Les conséquences sur la confiance créative, élément majeur de la créativité individuel, peuvent donc être très importantes et avoir des effets positifs ou négatifs sur la créativité individuelle. Finalement, l'IAG pourrait aussi avoir aussi un effet sur la capacité à prendre des risques intellectuels, puisqu'elle fournit un outil qui permet d'explorer rapidement un champ de connaissance. La propension à

prendre des risques intellectuels est une variable médiatrice du lien entre motivation intrinsèque et créativité des employés (Dewett, 2007).

La littérature montre que les outils et technologies peuvent être utilisés pour servir les intérêts de personnes ou de groupes dans les organisations (Spee et Jarzabkowski, 2009). Ils permettent de réaligner les ressources organisationnelles aux profits de ces parties prenantes. Ils sont ainsi des objets de pouvoir et leur utilisation prend une dimension politique. Par exemple, l'étude du processus collectif de feuilles de route technologique montre qu'elles peuvent servir à promouvoir les intérêts de parties prenantes, à légitimer de nouvelles activités, ou à consolider la juridiction sur un territoire d'action (Rampa et Parmentier, 2024). Les nouveaux outils d'IAG pourraient, en équipant les individus en outils de création, remettre en cause le statut des créatifs, modifiant ainsi les relations sociales dans l'organisation. L'analyse des affordances politiques de l'IAG consiste à examiner comment les différentes parties prenantes dans une organisation s'en emparent pour promouvoir leur créativité et leurs intérêts, et quelles sont les modifications des relations sociales suite à l'adoption de cette technologique.

Le tableau 1 présente les conséquences possibles sur le processus créatif en fonction du type d'affordance mobilisé par les utilisateurs de l'IAG.

Tableau 1. Affordances et processus créatif

Type d'affordances	**Conséquences sur le processus créatif**
Affordances fonctionnelles	Recherche d'information Recherche de solution Génération de codes, textes et images
Affordances cognitives	Pensée divergente Association d'idées et connaissances Transfert d'un principe de fonctionnement à d'autres sujets (analogie) Activation de l'imagination Pensée convergente
Affordances psychologiques	Confiance créative Motivation Prise de risque Perception de la créativité d'une IA Perception de la créativité d'un humain
Affordances politiques	Mise en valeur de sa créativité et sa performance Modification des relations avec les autres

3. Les effets de l'IAG sur la créativité des développeurs

Les métiers de développeur ont évolué en fonction de l'arrivée de nouveaux langages, des nouvelles technologies et d'outils de développement. Les outils d'aide au développement ont permis de faciliter le travail et d'augmenter la productivité (Lanubile *et al.*, 2010), et de rendre possible le développement distribué (Domingos Alves *et al.*, 2023). L'utilisation de l'IAG dans le développement améliore considérablement la productivité et semble réduire les erreurs (Peng *et al.*, 2023). Toutefois une grande partie du travail des développeurs informatiques mobilise des activités créatives de résolution de problème (Crawford *et al.*, 2013) et la créativité facilite la résolution de problèmes algorithmiques (Israel-Fishelson et Hershkovitz, 2022). De plus, un développement théorique récent suggère que l'hybridation humain-IAG pourrait avoir des effets significatifs sur l'accès à de nouvelles connaissances dans les activités de résolution de problèmes (Raisch et Fomina, 2023), menant ainsi à des solutions plus originales. Les nombreuses affordances mobilisables de l'IAG dans les activités de développement pourraient donc agir sur la créativité individuelle, mais aussi la créativité professionnelle en enrichissant les manières de penser, et les connaissances du développeur en situation de travail. Nous proposons donc d'examiner les effets de l'IAG sur les activités des développeurs et plus précisément sur leur créativité en analysant les affordances mobilisées dans le processus créatif, notamment sur les capacités des développeurs à définir les problèmes, et à chercher de nouvelles idées et solutions pour résoudre les problèmes de manière créative.

Méthodologie

Pour réaliser cette recherche, nous avons mené une étude qualitative exploratoire basée sur des entretiens semi-directifs dans l'industrie informatique. Les développeurs dans cette industrie utilisent depuis 2021 des outils d'aide au développement basée sur l'IAG. Les développeurs qui se sont approprié ces outils et qui utilisent régulièrement leur créativité pour trouver de nouvelles solutions sont donc une population intéressante pour étudier les effets de l'IAG sur leur créativité.

Nous avons enquêté auprès de programmeurs pour étudier comment ils s'approprient les technologies d'IAG et quelles sont les conséquences sur leurs pratiques et leur créativité. Nous avons recruté les développeurs dans une grande entreprise de produits et services de gestion d'énergie et à partir de prise de contact sur LinkedIn. Nous avons retenu que les développeurs qui utilisaient des outils d'IAG dans leur travail avec une diversité d'expériences dans le métier et de types de fonction de développement. Au final notre échantillon comporte 14 développeurs qui ont entre 25 à 40 ans, et qui ont entre 2 à 15 d'expériences professionnelles dans le métier (voir tableau 2). Le guide d'entretien a

été construit sur la base des 4 types d'affordances possibles avec les outils d'IAG. Nous avons estimé l'effet sur la créativité en repérant les effets sur leur manière d'identifier le problème, de rechercher de l'information sur le problème, de générer de nouvelles idées et d'explorer de nouvelles pistes, et de sélectionner les idées et solutions pour résoudre les problèmes.

Les interviews réalisées ont duré en moyenne 45 min au moins. Nous avons utilisé une technologie d'intelligence artificielle pour retranscrire les échanges avec la solution de retranscription automatique TADDAM développés par notre université. Nous avons utilisé la méthode par codage thématique sans IA avec une montée progressive en généralité pour identifier les affordances, les phases de développement, les mécanismes et effets de l'IAG sur la créativité des individus et des équipes.

Tableau 2. Caractéristiques de l'échantillon

Nom de code	Entreprise	Fonction	Niveau de formation	Année d'expérience de développement
Dev1	Schneider Electric	Ingénieur sénior	Bac + 5	10
Dev2	Schneider Electric	Data Scientist	Bac + 5	3
Dev3	Schneider Electric	Ingénieur logiciel	Bac + 5	10
Dev4	Expleo	Ingénieur logiciel	Bac + 5	5
Dev5	Alten	Data Scientist	Bac + 5	3
Dev6	Trimble Map	Ingénieur commercial	Bac +3	5
Dev7	Infomaniak	Développeur web	Bac + 3	5
Dev8	Groupe AEC	Ingénieur informatique	Bac + 5	3
Dev9	Sardel Conseil	Développeur	Bac + 5	1
Dev10	Ixiade	Développeur Web et logiciel	Bac + 2	3
Dev11	Leano	Responsable sécurité informatique et qualité	Bac + 8	15
Dev 12	Lab	Développeur	Bac + 2	2,5
Dev 13	Association Production Bas Carbone	Chef de projet	Bac + 5	5
Dev 14	Jetpulp	Développeur	Bac + 3	2

II - RÉSULTATS

1. Une utilisation très large dans toutes les tâches liées au développement

Le développeur effectue des tâches variées : analyse du besoin, recherche de la solution optimum, génération de codes, optimisation du code, test du code et documentation du code. Ces tâches peuvent se faire en partie directement dans le même logiciel avec GitHub Copilot alors qu'il faut sortir de l'environnement de développement pour utiliser ChatGTP. Les développeurs interrogés identifient plusieurs phases dans leur travail. Une première phase, qualifiée de « phase d'exploration » (Dev 2) ou « phase d'analyse » (Dev 9) renvoie à l'étape de recherche pour comprendre le besoin et apporter des solutions. Dev 14 la décrit comme la « phase de compréhension du cahier des charges ». La deuxième étape correspond à l'activité de codage. Certains développeurs font mention d'une étape de « répartition des tâches » (Dev 14), afin d'assigner les rôles et organiser le "workflow". La troisième étape concerne la correction de bugs et les tests nécessaires à la validation du code. Enfin, une dernière étape consiste au déploiement et à la maintenance du produit dans le temps. Les développeurs utilisant l'IAG décrivent donc un processus de développement en quatre étapes : analyse, codage, test et déploiement et maintenance, qui est assez classique dans les systèmes d'information (voir Reix *et al.,* 2011, page 337).

À l'exception de l'analyse du besoin, les développeurs utilisent les outils d'IAG dans toutes ces phases. Au-delà de la fonctionnalité de questionnement, ChatGPT permet de nombreuses possibilités d'actions non explicitées dans l'interface. Ces nombreuses possibilités d'actions sont bien mobilisées par les développeurs pour les aider dans les activités de développement. ChatGPT est très largement utilisée : 10 développeurs de notre échantillon affirment utiliser ChatGPT souvent ou de façon modérée, alors que l'utilisation de GitHub Copilot est plus faible et se limite essentiellement à la complétion automatique de code ; seulement 3 développeurs de notre échantillon affirment l'utiliser en permanence, souvent ou de façon modérée.

La recherche de solution pour atteindre un objectif ou résoudre un problème bien ou mal défini est une tâche centrale dans le développement informatique, car il existe une multitude de manières de coder. Pour atteindre un objectif ou résoudre un problème mal défini, les outils d'IAG accélèrent la recherche de solution, la réponse est rapide et peut être agrémentée d'exemples de code. Avant l'arrivée de ces outils, le développeur explorait les posts des forums de développement (un des plus connus étant Stack Overflow) comme l'exprime Dev 5 : *« Au lieu de parcourir 2 ou 3 forums, là, c'est vrai que souvent, en une réponse, c'est souvent réglé, quoi ».* Dans ces forums, les développeurs posent des questions du type : « comment faire pour… » ou « j'ai tel problème, quelles seraient les solutions pour le résoudre », et obtiennent quelquefois plusieurs réponses avec l'établissement d'un dialogue pour préciser la question ou demander des précisions sur la réponse. Cette recherche peut être assez longue avec la consultation de dizaines de pages, car les problèmes décrits peuvent être proches de ceux

rencontrés par le développeur sans pour autant correspondre totalement à sa situation. Avec l'utilisation des outils d'IAG, la réponse est immédiate et souvent plus en lien, et plus applicable aux objectifs et problèmes rencontrés. Les développeurs considèrent souvent ChatGPT comme un moteur de recherche "intelligent" plus précis qui interprètent mieux les demandes qu'une recherche sur un moteur de recherche classique. Toutefois, lorsque le problème est bien défini, quelques développeurs affirment encore utiliser les forums ou la recherche dans la documentation.

La génération de codes est utilisée à la fois avec le système de complétion automatique de GitHub Copilot et avec des demandes en langage naturel avec ChatGPT. Il est possible aussi de demander du code à partir d'une question sur GitHub Copilot, mais cette fonctionnalité est rarement utilisée, un seul développeur de notre échantillon affirme l'utiliser de façon modérée. Il s'agit essentiellement de génération de code assez simple que les développeurs considèrent comme inintéressant ou laborieux à développer. Ils cherchent à gagner du temps en déléguant à la machine des tâches ennuyeuses ou répétitives ou ils souhaitent obtenir du code dans un langage dont ils ont peu de connaissances.

Les outils d'IAG sont aussi utiles pour optimiser le code ou expliquer des erreurs. La recherche d'erreurs est une tâche assez longue qui nécessite quelquefois une longue pérégrination dans les forums ou dans une documentation assez dense et difficile à lire. L'IAG est capable d'apporter une réponse plus rapidement et souvent pertinente d'après les développeurs. L'optimisation du code est aussi une tâche confiée à l'IAG, car sauf si le développeur en fait un challenge, elle est considérée comme une tache peu intéressante. Ce cas d'usage est assez proche d'une opération de révision de textes (correction de grammaire et d'orthographe, amélioration du style) dans laquelle les IAGs sont assez performantes. Les développeurs utilisent d'ailleurs aussi l'IAG pour générer des scripts de tests afin de tester le bon fonctionnement de leur code et l'atteinte des objectifs, car cette tache est longue et répétitive.

Les possibilités de recherche et de synthèses textuelles des IAG sont aussi mises à profit par les développeurs pour extraire et structurer de l'information dans des bases de données ou de longs fichiers de logs, et même pour extraire une information ciblée dans les documentations de langage de développement : *« je vais me demander s'il la connaît et s'il peut me l'expliquer et me résumer pour m'éviter d'aller lire les 50 ou 100 pages de documentation de la librairie »* (Dev6). Dans ce cas, il s'agit d'automatiser une tâche qui demanderait de traiter un grand volume d'information ou de trouver une information extrêmement précise dans ce volume. Quelques développeurs utilisent la capacité rédactionnelle des IA génératives pour documenter le code, voire même pour générer des notices d'utilisation.

Les développeurs expliquent qu'ils utilisent l'IAG pour gagner du temps et être plus productifs. Ils confient des tâches à l'IAG soit parce qu'ils considèrent qu'elle est plus efficace pour l'exécuter, soit parce qu'ils trouvent que la tâche a peu de valeur ajoutée. Ils préfèrent utiliser leur temps et leur

capacité à mieux définir le problème, à travailler sur la structure logique de leur code, ou à explorer de nouvelles pistes. Pour la grande majorité, l'IAG apporte un gain de performance plus ou moins important en fonction de leur expérience de développement. Ce gain est lié à la génération et l'optimisation de code chez les jeunes développeurs (en termes d'expérience) alors qu'elle est plus liée à la recherche et la synthèse d'information chez les développeurs aguerris.

Certains développeurs indiquent que l'IAG agit sur leur motivation à passer à l'action. Comme ils savent qu'ils peuvent obtenir des réponses rapidement des outils d'IAG, ils s'engagent plus facilement dans des projets qui peuvent être compliqués, comme le précise Dev1 : *« …cette espèce de syndrome de la page blanche il n'existe pas du tout avec une IA ».* D'autres signalent que l'IAG les rend plus fainéant, car ils ne font plus l'effort de rechercher par eux-mêmes l'information ou délèguent des tâches de codage basique : *« Comme je l'ai dit, je deviens un peu plus fainéant sur certaines choses » (Dev6*).

Les développeurs constatent quasiment tous que l'IAG peut quelquefois indiquer des réponses fausses, du code non fonctionnel, et même inventer des documentations. Ce phénomène est dû au fait que les IAG sont basées sur des modèles de langage qui ne comprennent pas la logique procédurale du code, la prévision textuelle statistique peut donc aboutir à du code illogique, ne fonctionnant pas ou ne respectant pas les règles de syntaxes. De plus, l'IAG ne fait pas la distinction entre une fonction réellement implémentée d'une fonction discutée de manière spéculative dans les forums. Les erreurs peuvent aussi provenir d'une évolution rapide des bibliothèques de codes qui n'est pas prise en compte, comme l'exprime Dev4 : « *…dans des cas très particuliers, l'IA va être datée, elle va être un peu trop vieille* ». Cependant, comme le code peut rapidement être testé en situation, il semble que ces errements de l'IAG ne sont pas un problème en soi. Une majorité de développeurs mixent d'ailleurs l'utilisation de l'IAG et les moteurs de recherche, dans un sens ou dans l'autre.

L'utilisation de l'IAG semble avoir peu de conséquences sur les relations sociales au travail. Les développeurs évoquent tout de même le fait qu'ils demandent conseil à l'IA avant de demander à leurs collègues, ce qui pourrait amener à diminuer les relations sociales entre développeurs : *« Et donc, je vais essayer de trouver le moyen avec Google et ChatGPT… Et là, si vraiment ça ne fonctionne pas… je vais demander à mon responsable »* (Dev8).

En définitive, les développeurs affirment que, utiliser l'IAG est un facteur de performances et, que ne pas l'utiliser peut mener à être dévalorisé dans leur entreprise, puisqu'ils deviendraient moins performants que les autres. Elle est donc bien utilisée à des fins politiques pour maintenir son statut, voire même à l'améliorer. Il existe donc une pression, liée à la recherche de performance imposée par les entreprises, qu'elle soit quantitative ou qualitative, qui pousse à l'utilisation des outils d'IAG. Il serait presque impossible de s'en passer. Cette situation risque de mener à une dépendance des sociétés informatiques vis-à-vis des prestataires de services d'IA qui pourront alors imposer leurs

solutions, leur vision de la manière de développer et leur tarif, au risque de provoquer une normalisation excessive des tâches de développement et de capter une part croissante de la valeur ajoutée des entreprises utilisant des technologies numériques. Toutes ces activités sont synthétisées ci-dessous dans le tableau 3 en fonction des phases de développement des logiciels.

Tableau 3. Usages de l'IAG selon les phases de développement

Phases	**Analyse**	**Codage**	**Test**	**Déploiement et maintenance**
Activités	Identification des problèmes et recherche de solutions	Développement	Test, débug	Amélioration continue
Usage de l'IA générative	Recherche d'information sur un problème global Synthétisation de la documentation par ChatGPT. Utilisation des pre-prompts pour contextualiser la génération de code Élaboration d'idées en dialogue avec Chat GPT Apprentissage de langages ou technologies non maîtrisés Exploration de direction possible avec du code « jetable » généré par ChatGPT.	Auto-complétion avec GitHub Copilot Génération de code par Chat GPT (variables, tables...) Recherche d'information sur un problème de code Formulation des problèmes rencontrés et appropriation des réponses données par Chat GPT	Itération essai-erreur en implémentant le code généré par Chat GPT. Explication des bugs par Chat GPT. Optimisation du code Génération de scripts de tests	Aide à la documentation du code par Chat GPT. Aide à l'écriture de la documentation d'utilisation

2. Les conséquences sur la créativité des développeurs

L'analyse des affordances fonctionnelles, cognitives et psychologiques c'est-à-dire la manière dont est utilisée l'IAG, la représentation de ce que sont les outils d'IAG pour les développeurs et la manière d'entrer en relation avec l'IAG, fait ressortit deux modes relationnels Développeur-IAG : l'IAG comme assistant-développeur et l'IAG comme compagnon de développement. Ces deux modes peuvent être mobilisées par un même développeur avec une dominante pour l'un ou l'autre mode avec des conséquences différentes sur la créativité et l'apprentissage.

L'IA comme assistant de développement

Dans ce mode relationnel, comme l'exprime Dev4 : « *Et du coup, pour ma part, je l'utilise vraiment un petit peu comme un assistant* », l'IAG est considérée comme un assistant auprès duquel le développeur délègue toutes les tâches "inintéressantes, répétitives et rébarbatives", comme le précise Dev12 : *« Il y a des tâches qui peuvent nécessiter une heure d'automatisation, des choses qui sont*

vraiment ennuyeuses et répétitives ». Il s'agit alors de passer le moins de temps possible sur ces taches, voire même de les automatiser. Dans ce cas, il demande à l'assistant IAG de générer et commenter du code, de traiter des données, de résumer de longues documentations et de générer des tests. Le développeur sait assez précisément ce qu'il veut faire et où il veut aboutir : *« Donc, moi, je m'en sers beaucoup, des fois, pour des trucs un peu rapides, comme ça, où j'ai un peu la flemme d'écrire, parce que si je cherche et je sais l'écrire, il n'y a pas de problème »* (Dev3). Il élabore alors un prompt[3] qui décrit précisément la situation à l'aide d'une instruction directive avec comme objectif d'obtenir une réponse utilisable dès la première proposition. Dans le cas de l'utilisation de GitHub Copilot, l'utilisation avec la complétion automatique est d'emblée considérée comme un assistant et le développeur attend des propositions qui sont complètement alignées avec le travail en cours. Quand ils utilisent ChatGPT, ils copient-collent le code sans forcément essayer de l'analyser ou de le comprendre, sauf si celui-ci dysfonctionne : *« on se dit peut-être que quand je vais taper mon truc, il va me faire tout mon job et moi, j'aurai juste à copier-coller »* (Dev3).

Dans ce type de relation avec l'IA, les développeurs expriment une crainte de perte de compétence puisque ce type de délégation peut mener à oublier la réalisation de code basique. Il réduit aussi la possibilité d'apprentissage puisque les développeurs n'essayent pas de comprendre comment fonctionne le code quand les propositions sortent de leurs habitudes ou domaines d'expertises. Les développeurs craignent aussi de devenir fainéants et de déléguer de plus en plus de tâches à une machine. La recherche de productivité, en laissant peu de temps à l'analyse des propositions de l'IAG, pourrait donc mener à une diminution des apprentissages, voire même à des désapprentissages, comme l'exprime Dev 10 : *« Ce que je vois, c'est qu'il y a un vrai risque de désapprentissage quelque part »*. Dev13 rapporte privilégier sa productivité au détriment de sa réflexion : *« Il y a une erreur, je l'envoie dans ChatGPT, il me renvoie la réponse, et du coup, je ne réfléchis pas, je ne cherche pas. Je gagne plus de temps au quotidien parce que je ne me prends pas la tête. »*

Ce mode relationnel est peu propice à la créativité des développeurs puisqu'ils ne raffinent pas leurs prompts en reformulant la question posée, et ne poursuivent pas la conversation pour trouver de nouvelles idées ou de nouvelles solutions à un problème rencontré. *« On a l'impression d'être un peu passif »,* relate ainsi le Dev5. L'objectif est essentiellement une recherche d'efficacité et une délégation des tâches les plus ennuyeuses. L'usage de l'IAG réside avant tout dans ses avantages productifs et ne stimule donc pas beaucoup le potentiel créatif. Dans le premier cas de figure, l'IAG impacte donc peu la créativité des développeurs.

[3] Un prompt est une question rédigée dans l'interface de l'outil d'IA.

L'IA comme compagnon de développement

À l'inverse du mode précédent, l'IA est considérée ici comme un compagnon de travail qui aurait des compétences et du temps pour aider les développeurs à réaliser des tâches plus complexes : *« ChatGPT, ce serait plus... un collègue qui a du temps libre »* (dev10), *« Il est devenu une sorte d'ami. Je peux discuter avec eux de tout ce qui concerne mon projet »* (Dev12). Dans ce cas, il s'agit pour les développeurs de rechercher de nouvelles idées pour atteindre un objectif ou résoudre un problème. Il entre en dialogue avec l'IAG avec un raffinage du prompt au fur et à mesure des réponses. Certains développeurs ont une idée précise de la structure du prompt dès le départ, alors que d'autres le structurent au fur et à mesure des réponses. En fonction des réponses, le développeur demande alors à l'IAG des précisions, des explications ou reformule le prompt car la réponse n'est pas en rapport avec sa conception de la demande. Dans ce mode relationnel, pour trouver la solution, comprendre son fonctionnement et tester sa viabilité, le développeur effectue de nombreuses requêtes à l'IAG et, dans ce cas, il ne cherche pas à gagner du temps, mais à trouver "la solution". Deux développeurs débutants de notre panel expriment toutefois qu'il n'est pas facile de formuler un bon prompt. Le fait que ChatGTP n'ait pas le contexte du code (à la différence de GitHub Copilot) oblige aussi les développeurs à faire des prompts beaucoup plus longs pour présenter le contexte.

Nos entretiens ont mis en évidence différentes méthodes pour dialoguer avec les interfaces d'IAG. Trois étapes clés se dégagent pour maximiser la collaboration avec l'IAG. Premièrement, le cadrage à travers le *préprompt* sert à contextualiser la problématique. Ainsi, le développeur s'assure que l'IAG possède tous les éléments nécessaires à la résolution du problème. Deuxièmement, la phase de dialogue permet de peaufiner l'échange et d'apporter des précisions supplémentaires. Cette opération de « *fine-tuning »* à travers un dialogue prend la forme d'une *« espèce de ping-pong là où le contexte est prédéfini »,* souligne Dev1, et il rajoute que *« c'est une vraie aide à la créativité ».* Enfin, une troisième étape permet de confirmer ou d'infirmer la proposition de l'IAG en testant le code généré. Si le résultat n'est pas concluant, les développeurs réitèrent l'étape 2. Le mode relationnel de type compagnon est en définitive le plus susceptible de pousser la créativité des individus dans leur retranchement.

Dans ce mode, les développeurs demandent ainsi de la génération de code pour pouvoir l'analyser et le comprendre ou pour le tester rapidement afin de vérifier si une idée est bien adaptée à la situation. L'IAG sert alors à faire du prototypage rapide avec un processus d'essai-erreur.

Ce travail d'exploration des possibles agit sur la créativité des développeurs. Ils sont dans un premier temps dans une phase de clarification avec un processus itératif de formulation et reformulation du problème jusqu'à identifier la bonne formulation : *« Le fait d'utiliser ChatGPT, ça nous force d'une certaine façon à décrire le problème si on espère un résultat »* (dev4). Dans un second temps, avec le prototypage et les tests, ils sélectionnent la bonne solution. Bien que les idées de solution proviennent

de l'IAG, les utilisateurs les combinent fréquemment avec leurs propres connaissances. Ce sont eux qui sélectionnent et affinent l'idée finale, l'IA ne fournissant alors qu'une partie de la solution. Ce mode est aussi beaucoup plus favorable à l'apprentissage puisque les développeurs découvrent de nouvelles pistes, de nouvelles manières de faire, de nouvelles fonctions, voire de nouveaux langages : *« La recherche de solutions, d'idées même. Voilà. Et là, on voit qu'il y a plutôt une attitude d'apprentissage. C'est une piste que je n'aurais pas explorée. Je vais voir et je vais l'appliquer sur d'autres codes. Avec vraiment une volonté d'ouverture de chercher de nouvelles directions* » (Dev10). Ils ne se contentent pas de copier-coller la solution, mais ils font l'effort cognitif de comprendre et s'approprier la solution. La nouvelle connaissance sera donc au final beaucoup ancrée qu'avec une utilisation rapide sans effort pour se l'approprier. Les effets sur l'apprentissage impliquent plusieurs types d'affordances. Par exemple, dev14 témoigne que ChatGPT *« décrivait les étapes et [lui] permettait d'avoir une meilleure construction dans [sa] tête »*, et rajoute qu'elle *« lui disait bonjour avec smiley »* avant d'engager la conversation. L'IAG modifie donc le processus cognitif du développeur, mais également sa manière d'entrer en relation avec l'interface. La personnification de l'interface (comme un « compagnon », un « collègue » ou un « tuteur ») par le développeur fait intervenir l'affordance psychologique. Nous remarquons la nécessité de placer le dialogue avec Chat GPT dans une relation affective située par l'humain. Le tableau 4 ci-dessous synthétise les différentes affordances mobilisées dans le processus créatif.

Tableau 4. Impact des affordances mobilisées dans le processus créatif

	Identification du problème	**Recherche d'information**	**Génération d'idées**	**Validation et sélection des idées**
Affordances fonctionnelles	Demander à ChatGPT de structurer les étapes en vue de résoudre un problème (par ex pour intégrer une fonctionnalité ou organiser des tables). Demander à ChatGPT d'expliquer un bug pour trouver de nouvelles idées de solutions Donner le contexte d'un problème à ChatGPT en formulant un preprompt. Demander à ChatGPT ou GitHub Copilot du code rapide pour tester son idée de direction et redéfinir le problème	Demander à ChatGPT de synthétiser de la documentation pour estimer la pertinence du choix d'une librairie, fonction ou langage. Demander à ChatGPT d'expliquer des technologies (SQL, langages...) pour comprendre et faire des choix. Demander à ChatGPT ou GitHub Copilot du code rapide pour tester son idée	Demander à ChatGPT de générer des pistes pour organiser son code, trouver des noms de variables... Examiner le code proposé par GitHub Copilot ou Chat GPT pour évaluer de nouvelles orientations du code. Demander à ChatGPT de donner les étapes pour résoudre un bug ou de résoudre un bug Le code proposé par ChatGPT ou GitHub Copilot donne des idées sur d'autres parties du code	Demander à ChatGPT de tester son code pour valider ses idées. Demander à ChatGPT de générer des protocoles de tests pour valider ses idées.

Affordances cognitives	L'effort de clarification du développeur pour que ChatGPT comprenne le problème mobilise la pensée convergente (préciser son problème). Le dialogue entre ChatGPT et le développeur pour raffiner la requête nécessite de traduire des problèmes de code en prompts utilisant le langage naturel.	Le développeur associe les informations et explications générées par ChatGPT ou GitHub Copilot à ses connaissances à un projet de code existant. Le développeur restitue les connaissances de ChatGPT dans son projet de code.	Les idées générées par ChatGPT pour organiser et structurer son code peuvent infléchir de nouvelles pistes (pensée divergente). Par exemple, pour créer et intégrer des composants dans son code. Des idées de fonctionnalités proposées par ChatGPT peuvent stimuler l'imagination des développeurs.	Lorsque les contenus de ChatGPT ou GitHub Copilot fonctionnent dans le code du développeur, cela précise et valide une manière de faire (pensée convergente).
Affordances psychologiques	La perception de ChatGPT ou GitHub Copilot comme un « collègue » ou « ami » peut motiver le développeur à poser des questions, en vue de résoudre un problème.	Le large volume d'informations que possède ChatGPT incite le développeur à se tourner vers lui pour obtenir des connaissances. La rapidité et la portée des réponses de ChatGPT motivent l'interaction et peuvent initier le passage à l'action.	Le code généré par ChatGPT ou GitHub Copilot permet la montée en confiance du développeur et une plus grande prise de risque intellectuelle. L'augmentation des performances et de la productivité du développeur grâce à ChatGPT ou GitHub Copilot lui permet d'allouer plus de temps aux tâches créatives.	Au terme d'un dialogue ayant permis de valider un morceau de code, la perception de ChatGPT par le développeur sera améliorée et renforcera sa confiance à son égard. Renforcement du lien affectif entre les actants et pérennisation de l'usage de ChatGPT ou GitHub Copilot.
Affordances politiques	L'intégration de ChatGPT dans les phases d'identification de problèmes modifie les dynamiques de groupe au sein des équipes. Par exemple, demander de l'aide à ChatGPT est devenu un prérequis avant de solliciter un collègue.	L'emploi de ChatGPT pour trouver des informations devient normalisé et socialement accepté chez les développeurs. Son usage tend à dépasser celui de Stackoverflow, témoignant de son acceptation sociale.	Possibilité de valoriser de nouvelles pistes issues d'une conversation avec ChatGPT auprès des collègues. Possibilité de valoriser sa performance et témoigner de l'usage ingénieux de ChatGPT (prompt engineering).	

III - DISCUSSION

Les développeurs se sont largement emparés des outils d'IAG pour les aider dans leurs activités de développement. Ils n'utilisent pas seulement l'IAG pour générer du code, mais aussi pour, par exemple, rechercher de l'information afin de résoudre des problèmes de développements, synthétiser ou remettre en forme de grande masse d'informations ou tester et corriger du code. Les outils d'IAG induisent aussi des affordances cognitives liées à la créativité : identification du problème, divergence avec de la génération d'idées et convergences avec le test de ces idées. L'utilisation de l'IAG

provoque une augmentation de prise de risque intellectuelle et de la motivation à passer à l'action. Nous constatons peu de mobilisation des outils d'IAG pour des objectifs politiques, l'IAG est en effet peu utilisée pour obtenir une reconnaissance, du pouvoir ou des ressources supplémentaires. Toutefois l'augmentation de la performance des développeurs par l'IAG rend quasiment indispensable son utilisation, ce qui en fait d'emblée un outil politique pour maintenir sa position dans l'organisation.

Notre étude révèle en outre que le dialogue *humain-machine* catalyse le potentiel créatif des développeurs en fonction de leur mode de relation avec l'IAG. Quand ils considèrent l'IAG comme un assistant, leur objectif principal est de gagner du temps et les actions effectuées avec l'IAG ont peu de conséquences sur le processus créatif. Quand l'IAG est pour eux un compagnon, ils entrent en dialogue pour clarifier une situation, explorer des possibles et trouver de nouvelles idées de solutions, se servant de l'IAG pour favoriser les processus cognitifs de la créativité. Dans ce deuxième cas de figure, l'IAG se rapproche du scénario de « Co-Cre-AI-tion » (Vinchon *et al.*, 2023) dans lequel la relation humain-IA engendre des outputs qu'aucun des agents n'aurait pu produire seul.

Nos travaux confirment avec des professionnels les résultats de Boussioux *et al.* (2024) qui montrent que l'utilisation de prompts successifs reliés entre eux favorise la créativité des utilisateurs. Néanmoins nous allons plus loin en identifiant ici un tandem relationnel entre l'IA et l'humain. Cette relation avec la machine est cognitive, mais aussi affective. Le mode de relation « relational artifact » est notamment défini dans les travaux de Lim *et al.* (2023). Ils insistent sur la nécessité de développer une interaction avec l'interface au travers des dialogues. À la différence des technologies précédentes qui agissaient sur l'organisation des tâches de développement (Domingos Alves, *et al.*, 2023 ; Lanubile *et al.*, 2010), cette technologie est aussi conversationnelle, l'interaction n'est pas seulement une action avec un retour reproductible, l'interaction se fait à travers une conversation avec des aller-retour multiples. Le développeur qui utilise l'IAG est amené à donner du sens à cette relation, qui, dans notre cas, est une relation de délégation de tâches à un assistant ou de co-construction des tâches avec un "quasi collaborateur". Le cadre d'analyse des affordances quand il est étendu aux possibilités d'actions psychologiques, cognitives et politiques est donc intéressant pour mieux comprendre cette relation. Les résultats contrastés des études antérieures sur l'effet de l'utilisation de l'IAG sur la créativité pourraient être liés aux effets psychologiques et cognitifs qui ne sont pas pris en compte dans les enquêtes quantitatives. S'il est difficile d'affirmer que l'IAG rend créatif en soi, nous pouvons toutefois constater que l'exploration des possibles avec l'IAG agit sur le processus créatif. Nous faisons l'hypothèse que c'est aussi le cas pour toutes les professions créatives.

Nous constatons également que les organisations ont inégalement investi les technologies de l'intelligence artificielle. Pour les industries technologiques, le virage IA s'est naturellement imposé, facilitant ainsi l'acculturation des développeurs à l'IAG. Par exemple, Schneider Electric inscrit l'utilisation de l'IAG dans sa stratégie avec une collaboration assumée entre l'IAG et les individus. D'autres secteurs, moins portés sur ces technologies, peinent encore à intégrer l'intérêt de l'IAG pour

les développeurs. Ces derniers se saisissent donc parfois individuellement de l'IAG et ne bénéficient pas forcément d'un soutien managérial. L'enquête effectuée par GitHub révèle que, pour un tiers des développeurs, l'évaluation de leur performance se fait en fonction du volume de code qu'ils produisent (GitHub et Inbal, 2023). Toutefois, des recherches indiquent que le volume, la qualité, la vitesse et la satisfaction du client sont des dimensions qui se complètent, mais qui nécessitent des mesures spécifiques, à la fois quantitative et qualitative, qui doivent plutôt s'élaborer au sein de l'équipe (Ko, 2019). Notre étude confirme que l'IAG améliore la productivité en écriture de code, et qu'en mode relationnel compagnon, elle aide à résoudre des problèmes complexes et à explorer de nouvelles solutions. Il conviendrait donc pour les entreprises de l'industrie informatique d'introduire plus de critères d'évaluation qualitative, décidé plus localement dans les équipes, pour évaluer la performance des développeurs. En mettant au cœur l'augmentation de la créativité avec l'IAG chez les employées (Jia *et al.*, 2024), les entreprises disposeraient ainsi de ressources supplémentaires pour être plus performantes et plus innovantes.

Nos recommandations portent aussi sur les politiques d'intégration et d'appropriation de l'IAG par les développeurs. Force est de constater que ChatGPT et GitHub Copilot sont aujourd'hui largement utilisés, mais leur mobilisation souffre d'un manque d'accompagnement et de formation à l'écriture de prompts de la part des entreprises. L'emploi de ces technologies est encore en phase exploratoire et dépend des contextes de travail et des modes relationnels. Les premières études d'utilisation de ces outils indiquent que s'il y a bien des gains de productivité, les développeurs rencontrent des difficultés d'intégration du code et que son utilisation nécessite une évaluation critique (Grewal et al. 2024 ; Zhang et al. 2024). Nous préconisons donc de conduire des tests pour chaque étape de développement, afin de dégager des évaluations de l'utilité du code généré par l'IAG, des performances des développeurs et de la qualité du code final. Ces évaluations permettraient d'établir les bonnes pratiques selon les spécificités de chaque entreprise. Il serait aussi pertinent d'envisager une collaboration humain-IAG pour certaines tâches effectuées entre humains. Par exemple, la relecture et la validation de codes entre deux développeurs sont des pratiques courantes pour les entreprises aux processus matures et éprouvés. Les plus petites structures sont quant à elles moins susceptibles de systématiser ces pratiques par manque de ressources. L'usage de l'IAG comme "re-lecteur" pourrait ici pallier cette contrainte, et la proposition des pistes d'amélioration augmenter la qualité du code final.

Reste à interroger la nature des modes relationnels entre l'IAG et l'humain pour nourrir la créativité des individus et optimiser la production d'idées. Dans le cadre de l'apprentissage assisté par l'IA générative, Mike Sharples (2023) introduit 6 rôles pour décrire les modalités d'échanges entre ChatGPT et des étudiants : générateur de possibilités, opposant dans l'argumentation, co-concepteur, outil exploratoire et collaborateur dans l'écriture créative. Chacun des rôles décrit la manière dont ses étudiants envisagent le dialogue. Par exemple, le rôle d'opposant dans l'argumentation (*socratic*

opponent) privilégiera un échange conflictuel sur la base d'un débat entre l'humain et l'interface. Le rôle co-concepteur désigne quant à lui une collaboration vers un objectif partagé. Ces illustrations montrent les différentes applications possibles pour engager le dialogue dans le cadre d'une relation de collaboration Humain-IAG, mais cette relation est aussi dépendante de la conception que se fait le créatif de l'IAG et de sa relation avec l'IAG.

Notre recherche comporte toutefois des limites liées à la méthodologie qualitative employée, adaptée pour une étude exploratoire, mais qui appelle à des études confirmatoires. L'échantillon est limité, au seul développeur, il n'est donc pas possible d'étendre ces résultats à l'ensemble des professions créatives. De plus notre panel intègre peu d'utilisateurs réguliers de GitHub Copilot, ce qui limite la compréhension des effets de l'utilisation de la complétion de code. Nous éclairons toutefois un phénomène qui doit être exploré en détail dans plusieurs industries créatives pour mieux comprendre ce qui se joue dans la relation humain-IA avec cette technologie conversationnelle.

Nous appelons donc à développer les recherches expérimentales sur la relation Humain-IAG dans de multiples activités de création pour identifier à quels moments, avec quels états d'esprit, quel profil d'utilisateur, et sur quelles activités l'IA apporterait le plus de valeur ajoutée en termes de performance et de créativité. Il s'agira de travailler avec les développeurs d'IA pour créer des IA éthiques qui augmentent les capacités créatives des humains au lieu de les concurrencer dans cette activité fondamentalement liée à notre humanité.

BIBLIOGRAPHIE

Agogué M., Levillain K., Hooge S. (2015). « Gamification of Creativity: Exploring the Usefulness of Serious Games for Ideation », *Creativity & Innovation Management*, vol. 24, n° 3, p. 415-429.

Amabile T.M. (1983). *The social psychology of creativity*, Springer-Verlag, New York.

Amabile T.M. (1988). « A model of creativity and innovation in organizations », *Research in Organizational Behavior*, vol. 10, p. 123-167.

Amabile T.M. (2020). « Creativity, Artificial Intelligence, and a World of Surprises », *Academy of Management Discoveries*, vol. 6, n° 3, p. 351-354.

Ameen N., Sharma G.D., Tarba S., Rao A., Chopra R. (2022). « Toward advancing theory on creativity in marketing and artificial intelligence », *Psychology & Marketing*, vol. 39, n° 9, p. 1802-1825.

Boden M.A. (1998). « Creativity and artificial intelligence », *Artificial Intelligence*, vol. 103, n° 1, p. 347-356.

Boussioux L., Lane J.N., Zhang M., Jacimovic V., Lakhani K.R. (2024). « The Crowdless Future? Generative AI and Creative Problem-Solving », *Organization Science*, vol. 35, n° 5, p. 1589-1607.

Chen S., Chandler J., Venkatesh A. (2020). « The influence of objects on creativity », *Creativity & Innovation Management*, vol. 29, n° 3, p. 481-494.

Chui M., Hazan E., Roberts R., Sigla A., Smaje K., Sukharevsky A., Yee L., Zemmel R. (2023). *Economic potential of generative AI, The next productivity frontier*, McKinsey.

Crawford B., Barra C., Soto R., Misra S., Monfroy E. (2013). « Knowledge management and creativity in software engineering the foundations of agility », *Computational Science and Its Applications*, ICCSA.

Cummings S., Bilton C., Ogilvie D. (2015). « Toward a New Understanding of Creative Dynamics: From One-Size-Fits-All Models to Multiple and Dynamic Forms of Creativity », *Technology Innovation Management Review*, vol. 5, n° 7, p. 14-24.

Davenport T.H., Ronanki R. (2021). « Artificial Intelligence for the Real World », *Harvard Business Review*, p. 44-53.

Deci E.L., Ryan R.M. (1985). *Intrinsic motivation and self-determination in human behavior*, Plenum, New York.

Dewett T. (2007). « Linking intrinsic motivation, risk taking, and employee creativity in an R&D environment », *R&D Management*, vol. 37, n° 3, p. 197-208.

Diallo M.F. (2023). « Ce que ChatGPT fait à l'enseignement, à la recherche et aux organisations », *Revue française de gestion*, vol. 312, n° 5, p. 9-14.

Domingos Alves D., Souza Matos E. de, Chavez C. von F.G. (2023). « Interaction design in distributed software development: a systematic mapping study », *Behaviour & Information Technology*, p. 1-37.

Doshi A.R., Hauser O.P. (2024). « Generative AI enhances individual creativity but reduces the collective diversity of novel content », *SciencesAdvances*, vol. 10, n° 28.

Gaver W.W. (1991). « Technology affordances », *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 79-84.

Gibson J. (1977). «The theory of affordances », dans *Perceiving, acting, and knowing: Toward an ecological psychology*, Hillsdale, Erlbaum, p. 67-82.

Gillier T., Bayus B.L. (2022). « Group creativity in the wild: When building on ideas enhances the generation and selection of creative ideas », *Creativity and Innovation Management*, vol. 31, n°3, p. 430-446

GitHub S., Inbal S. (2023). « Survey reveals AI's impact on the developer experience », *The GitHub Blog*.

Grewal B., Lu W., Nadi S., Bezemer, C-P. (2024). « Analyzing Developer Use of ChatGPT Generated Code in Open Source GitHub Projects », *Proceedings of the 21st International Conference on Mining Software Repositories*, p. 157-161

Guilford J.P. (1950). « Creativity », *American Psychologist*, vol. 5, p. 444-454.

Hartson H.R. (2003). « Cognitive, physical, sensory, and functional affordances in interaction design », *Behaviour & Information Technology*, vol. 22, n° 5, p. 315.

Israel-Fishelson R., Hershkovitz A. (2022). « Studying interrelations of computational thinking and creativity: A scoping review (2011–2020) », *Computers & Education*, vol. 176, p. 104353.

Ivcevic Z., Grandinetti M. (2024). « Artificial intelligence as a tool for creativity », *Journal of Creativity*, vol. 34, n° 2, p. 100079.

Jarzabkowski P., Kaplan S. (2015). « Strategy tools-in-use: A framework for understanding "technologies of rationality" in practice », *Strategic Management Journal*, vol. 36, n° 4, p. 537-558.

Jia N., Luo X., Fang Z., Liao C. (2024). « When and How Artificial Intelligence Augments Employee Creativity », *Academy of Management Journal*, vol. 67, n° 1, p. 1-1.

Ko A.J. (2019). « Why We Should Not Measure Productivity », dans SADOWSKI C., ZIMMERMANN T. (dirs.), *Rethinking Productivity in Software Engineering*, Apress, Berkeley, CA, p. 21-26.

Lamers M.H. (2023). « A Computer Scientist's View on Machine Learning and Creativity » ANUTARIYA C., BONSANGUE M.M. (dirs.), *Data Science and Artificial Intelligence*, p. 325-335.

Lanubile F., Ebert C., Prikladnicki R., Vizcaíno A. (2010). « Collaboration Tools for Global Software Engineering », *IEEE Software*, vol. 27, n° 2, p. 52-55.

Lim J., Leinonen T., Lipponen L., Lee H., DeVita J., Murray D. (2023). « Artificial intelligence as relational artifacts in creative learning », *Digital Creativity*, vol. 34, n°3, p. 192-210.

Lubart, T. (2003). *Psychologie de la créativité*, Armand Colin, Paris

Mikalef P., Gupta M. (2021). « Artificial intelligence capability: Conceptualization, measurement calibration, and empirical study on its impact on organizational creativity and firm performance », *Information & Management*, vol. 58, n° 3, p. 103434.

Peng S., Kalliamvakou E., Cihon P., Demirer M. (2023). « The Impact of AI on Developer Productivity: Evidence from GitHub Copilot ».

Raisch S., Fomina K. (2023). « Combining Human and Artificial Intelligence: Hybrid Problem-Solving in Organizations. », *Academy of Management Review*, p. 1-24.

Rampa R., Parmentier G. (2024). « The affordances of technology and strategic roadmapping: An exploration of its instrumental, symbolic, and political functions », *Journal of Engineering and Technology Management*, vol. 71, p. 101778.

Reix R., Fallery B., Kalika M. (2011). *Systèmes d'information et management des organisations*, 6e édition, Vuibert, Paris.

Runco M.A. (2023). « AI can only produce artificial creativity », *Journal of Creativity*, vol. 33, n° 3,

p. 100063.

Sharples M. (2023). « Towards social generative AI for education: theory, practices and ethics », *Learning: Research and Practice*, vol. 9, n° 2, p. 159-167.

Spee A.P., Jarzabkowski P. (2009). « Strategy tools as boundary objects », *Strategic Organization*, vol. 7, n° 2, p. 223-232.

Tanggaard L. (2013). « The sociomateriality of creativity in everyday life », *Culture & Psychology*, vol. 19, n° 1, p. 20-32.

Tierney P., Farmer S.M. (2002). « Creative self-efficacity: its potential antecedents and relationship to creative performance », *Academy of Management Journal*, vol. 45, n° 6, p. 1137-1148.

Vinchon F., Lubart T., Bartolotta S., Gironnay V., Botella M., Bourgeois-Bougrine S., Burkhardt J.-M., Bonnardel N., Corazza G.E., Glăveanu V., Hanchett Hanson M., Ivcevic Z., Karwowski M., Kaufman J.C., Okada T., Reiter-Palmon R., Gaggioli A. (2023). « Artificial Intelligence & Creativity: A Manifesto for Collaboration », *The Journal of Creative Behavior*, vol. 57, n° 4, p. 472-484.

Woodman R.W., Sawyer J.E., Griffin R.W. (1993). « Toward a theory of organizational creativity », *Academy of Management Review*, vol. 18, n° 2, p. 293-321.

Wróbel A.E., Johansen M.K., Jørgensen M.S., Cash P. (2021). « Facilitating creativity: Shaping team processes », *Creativity & Innovation Management*, vol. 30, n° 4, p. 742-762.

Zhang B., Liang P., Zhou X., Ahmad A., Waseem M. (2023). « Practices and Challenges of Using GitHub Copilot: An Empirical Study », *Arxiv*, p. 124-129.