



YVON GOUTAL,
avocat et professeur associé
en droit public (université
de Reims), cabinet Goutal,
Alibert et associés



THÉO SIMON,
juriste et consultant,
cabinet Goutal, Alibert
et associés

Définition

Le «vibe coding» consiste à décrire à une IA ce que l'on attend d'un logiciel et la laisser produire le code correspondant : la programmation pour les littéraires.

Encadrement

Le code «vibe codé» est rapide à écrire ; les questions qu'il pose sont plus longues à instruire : propriété intellectuelle, données personnelles, système d'intelligence artificielle...

Organisation

Le «vibe coding» fait éclore, dans les services, des compétences que la collectivité ignorait avoir. Reste à en organiser l'exercice et à en tirer parti.

Accueillir le «vibe coding» des agents publics

01 Comprendre ce qu'est le «vibe coding»

Ces dernières semaines, l'expression «vibe coding» est sur toutes les lèvres (enfin, sur celles des férus d'intelligence artificielle, IA). Mais, finalement, de quoi parle-t-on ? Il s'agit, littéralement, d'une méthode de «programmation au ressenti». Elle désigne la production de logiciels et, plus largement, de code informatique grâce à un dialogue, en langage courant. L'humain décrit ce qu'il veut et l'intelligence artificielle générative (IAG) transforme le souhait en code.

Les agents trouvent ainsi des réponses techniques à leurs problèmes et leur créativité fait éclore des logiciels «métiers» d'un nouveau genre : tableau de bord budgétaire, outil de suivi des autorisations d'urbanisme, logiciel de pilotage des contrats. Autant d'applications sur mesure qui peuvent rivaliser avec des logiciels propriétaires dont les abonnements sont souvent coûteux.

Les outils d'IAG grand public ont abaissé la barrière d'entrée à presque rien : d'abord réservée aux initiés, la pratique du codage se répand dans toutes les directions métiers : directions des systèmes d'information, mais aussi finances, ressources humaines, urbanisme et même juridique...

L'effet «waouh» est immédiat, mais, comme bien souvent, assez trompeur. Une application fonctionnelle apparaît en seulement quelques minutes, ou jours, au lieu de semaines dans un développement traditionnel. Mais l'image est souvent «flatteuse» et le «peu» qu'il reste à accomplir prend souvent beaucoup plus de temps que la création initiale. Pour l'exprimer autrement, les 10 à 20% qui manquent au prototype concentrent une bonne part de la difficulté et n'apparaissent qu'à l'usage. De la coupe aux lèvres, de la maquette à l'application fonctionnelle, il y a parfois un long chemin...

02 Se positionner (et le faire savoir)

La tentation est forte, en présence d'un phénomène nouveau et non maîtrisé, de prononcer une interdiction pure et simple. Il n'est pas certain que la solution soit la plus efficace : il est peu probable qu'une note de service suffise à contrecarrer la facilité d'usage d'un outil que les agents utilisent parfois déjà chez eux. Plutôt que d'interdire ou de laisser faire, la collectivité

a tout intérêt à adopter une position à la fois nuancée et explicite. Le bon véhicule pour formaliser cette doctrine d'usage peut être la «charte IA», mise en cohérence avec la charte informatique de la collectivité (si elle existe) et son règlement intérieur.

On conseillera, notamment, de distinguer les usages internes non sensibles, qui peuvent être assez largement ouverts, de tout ce qui touche au système d'information ou à l'utilisateur. Dans ce dernier cas, le principe doit être celui d'une supervision par un ou plusieurs agents dotés des compétences juridiques et informatiques.

La question des IAG mobilisées par les applications devra également être surveillée, notamment pour vérifier les clauses contractuelles et les garanties offertes. Par ailleurs, le recensement des applications produites est vivement recommandé, de même que la conservation des versions du code, accompagnées d'une trace des choix de conception : à quelle contrainte technique répond chaque décision, quelle alternative a été écartée et pourquoi, quel prompt a engendré quel composant...

Cette mémoire «structurée» permettra, le jour venu, de ne pas faire reposer la maintenance du logiciel sur la seule mémoire des collègues de l'agent ayant quitté son poste.

03 Maîtriser la répartition des droits de propriété intellectuelle

Le «vibe coding» nous renvoie à des problématiques déjà explorées, dont celle des éventuels droits d'auteur des agents. Par principe, un logiciel n'est protégé par le droit d'auteur que s'il présente un caractère «original». L'originalité, essentielle, se déduit de deux éléments : un apport intellectuel propre de son auteur, permettant de distinguer sa création d'autres logiciels existants (1) et de véritables choix créatifs dans le développement du logiciel (2).

Dans le cas du «vibe coding», l'apport intellectuel et les choix créatifs sont a priori davantage ceux du modèle d'IA que ceux de l'agent, ce qui, toujours a priori, prive le code de protection. Mais la nuance s'impose : si le «vibe coding» a été assorti d'une forte



On conseillera notamment de distinguer les usages internes non sensibles, qui peuvent être assez largement ouverts, de tout ce qui touche au système d'information ou à l'utilisateur.

intervention humaine, de multiples itérations, voire d'une réécriture partielle, l'originalité – et donc la protection – peut être revendiquée. Afin de régler cette question, le degré d'intervention humaine devra être démontré et donc documenté à mesure que le code est produit.

Quoi qu'il en soit, si le «vibe coding» intervient dans l'exercice des fonctions de l'agent ou d'après les instructions de son employeur, les droits patrimoniaux sur les logiciels sont dévolus automatiquement à la personne publique employeur (code de la propriété intellectuelle, art. L.113-9). On notera que la jurisprudence retient une interprétation extensive du «logiciel créé en service»: est par exemple considéré comme tel le logiciel réalisé en dehors des heures de travail, mais avec l'aide du matériel de l'employeur.

En cas d'externalisation, même partielle, deux points méritent une vigilance particulière. La transparence, d'abord: déclaration obligatoire de l'usage d'IAG par le titulaire du marché. La garantie d'éviction, ensuite: si le code livré reproduit des fragments protégés du corpus d'entraînement du modèle d'IA utilisé, la collectivité doit pouvoir se retourner contre son cocontractant. Cela suppose que ce dernier dispose effectivement des droits qu'il prétend céder – et donc qu'il n'utilise lui-même l'IAG que dans des conditions contractuelles solides vis-à-vis de l'éditeur du modèle.

04 Mesurer ce que le code embarque

Une application «vibe codée» transporte potentiellement trois passagers, que son auteur ne soupçonne pas: du droit d'auteur (la propriété intellectuelle des corpus d'entraînement), des données personnelles (celles que l'agent a glissées dans ses prompts et/ou celles que l'application traitera ensuite), et parfois un modèle d'IA. Trois motifs de vigilance.

Les grands modèles de langage sont entraînés sur des corpus de données dont la licéité n'est pas toujours établie. Le code n'est pas créé ex nihilo et l'IA recombine, parfois littéralement, des fragments de ses données d'entraînement. La frontière entre l'inspiration et la reproduction est poreuse. Le risque, pour la collectivité qui déploie un code vibe codé, est celui d'une

revendication par un titulaire de droits sur les corpus d'entraînement. La parade existe: il s'agit d'une clause d'indemnisation par l'IAG, voire d'une «garantie d'éviction», déjà évoquée, que la collectivité doit chercher à obtenir du fournisseur, ce qui impose au minimum d'examiner les contrats et d'arbitrer entre les formules d'abonnement.

Le risque face au règlement général sur la protection des données (RGPD) est plus quotidien et plus diffus. Durant la production du code, l'agent peut communiquer des données personnelles au fournisseur du modèle, qui les conservera selon ses propres conditions, et qui pourra, sauf stipulation contraire, les réutiliser pour entraîner ses modèles. Dans le même esprit, une fois créée, l'application peut être appelée à traiter des données personnelles pendant son utilisation: le RGPD s'applique pleinement (base légale, finalité explicite, analyse d'impact, etc.). Précisons également que le produit final peut comporter des défauts de sécurité que l'agent n'imagine pas (par exemple, un mot de passe en clair dans le code source). Or la sécurité des données collectées est une obligation (de moyens) à la charge des collectivités.

Enfin, l'application produite peut, sans que l'agent ne s'en rende compte, revêtir la qualification de «système d'IA», parfois classé à «haut risque», au sens du règlement européen sur l'IA (RIA): outil de priorisation dans l'accès aux services publics essentiels, logiciel d'évaluation de l'éligibilité à des prestations, outil de gestion du personnel (annexe III du RIA). Rien d'insurmontable, mais à surveiller...

05 Savoir profiter de l'aubaine

On l'a compris, tout l'enjeu est d'organiser une pratique qui existe déjà, plutôt que de tenter – en vain – de l'interdire ou de la laisser se développer sans contrôle. Ce qui suppose, en premier lieu, de cerner ce que le «vibe coding» révèle. Dans de nombreux services, certains agents construisent aujourd'hui, sur leur temps de travail, des outils parfois comparables à ce que la collectivité aurait, hier, acheté sur étagère à des prix souvent élevés. Mieux, les outils créés peuvent être formidablement pertinents: les agents comprennent en effet mieux que quiconque les contraintes de

leur métier et, même s'ils n'ont pas été formés à la programmation, l'IA leur permet de dépasser cette carence. En somme, le «vibe coding» est aussi un superbe gisement de nouvelles compétences qui mérite d'être valorisé, encouragé, tout en érigeant quelques garde-fous.

Le premier est d'inscrire le «vibe coding» dans des environnements dédiés («bac à sable», licence collective présentant des garanties suffisantes...) qui rendent la pratique plus sûre. Le deuxième est de former les agents au «prompt engineering» et à la relecture critique du code généré. Le troisième est l'organisation d'une gouvernance et d'une documentation: registre des applications développées en interne via IAG, recensement de leur périmètre, de leurs utilisateurs, des données qu'elles traitent et de leur statut au regard de la réglementation applicable (RGPD, RIA...).

Une suggestion en forme de sourire pour conclure: le «vibe coding» pourrait, aussi, être l'occasion de doter la collectivité d'un véritable outil de gestion de son parc applicatif – celui que l'on aurait dû construire depuis bien longtemps... ●

(1) Cour de cass., 2 septembre 2011, pourvoi n°09-71.337.

(2) Cour de cass., 16 avril 1991, pourvoi n°89-21.071.



la Gazette.fr

Retrouvez nos fiches juridiques

www.lagazette.fr/rubriques/fiches-de-droit-pratique