SAKIRA

ARTICLE

Règles de la NASA : ce que je demande à mes IA

Saray Hach · 5 novembre 2026

© SAKIRA

sakira.fr
Texte de Saray Hach, © 2026 Saray Hach, publié par SAKIRA SAS
SAKIRA SAS · 6 rue d'Armaille, 75017 Paris · SIREN 108 493 933

Règles de la NASA : ce que je demande à mes IA

Saray Hach, fondateur de SAKIRA

SOMMAIRE

1. Ce que proposent les règles de la NASA
2. Les règles que je pose à mes IA
3. L'IA qui écrit et l'IA qui répond
4. Ce que je cherche avant de confier le travail

Le 19 septembre, en travaillant sur un programme avec mes IA, j'ai rencontré un refus qui prenait 5,8 secondes. La valeur demandée était démesurée. Le programme devait la refuser immédiatement. Mais, pour expliquer ce refus, il calculait lui-même un nombre énorme.

J'avais demandé une limite. Elle existait à l'entrée du programme, puis le message d'erreur la contournait. J'en ai fait une règle supplémentaire : le message de refus doit être borné lui aussi.

C'est ce genre de détail que je regarde quand une intelligence artificielle (IA) écrit du code pour moi. J'ai envie de savoir comment le programme se comporte lorsque quelque chose lui manque, lorsqu'une demande dépasse ses limites ou lorsqu'une personne se trompe. Les règles de la NASA m'intéressent pour cette raison : elles rendent le code plus facile à vérifier. J'ai prolongé cette démarche dans les règles que je donne à mes IA.

Ce que proposent les règles de la NASA

En 2006, Gerard Holzmann, du laboratoire de fiabilité logicielle du Jet Propulsion Laboratory (JPL), à la NASA, l'agence spatiale américaine, publie [The Power of Ten](https://spinroot.com/gerard/pdf/P10.pdf). Ses dix règles visent principalement le langage C et les logiciels critiques, dont une erreur peut avoir des conséquences graves.

Les enchaînements restent simples. La récursion est exclue : une fonction ne se rappelle pas elle-même. Les répétitions ont une limite vérifiable. La mémoire nécessaire est réservée au démarrage. Les fonctions sont courtes.

Des assertions, des vérifications intégrées au code, détectent les anomalies. Les variables, qui conservent les données, sont définies au plus près de leur usage. Les paramètres reçus et les résultats des appels sont contrôlés.

L'emploi du préprocesseur est limité. Les pointeurs sont fortement encadrés. La compilation et l'analyse statique doivent passer sans avertissement.

La compilation traduit le code en programme exécutable. Le préprocesseur transforme le texte avant cette traduction. Les pointeurs désignent des emplacements en mémoire. L'analyse statique examine le code sans l'exécuter.

Le [manuel d'ingénierie logicielle de la NASA](https://swehb.nasa.gov/spaces/SWEHBVD/pages/102695794/9.03%2BCoding%2BStandards) décrit aussi des standards adaptés aux langages et aux projets. Je m'intéresse à ce que cette démarche permet de demander aux

Texte de Saray Hach, © 2026 Saray Hach, publié par SAKIRA SAS · sakira.fr
SAKIRA SAS · 6 rue d'Armaille, 75017 Paris · SIREN 108 493 933

logiciels que mes IA construisent.

Les règles que je pose à mes IA

Mes règles portent sur le déroulement du travail et sur les traces qu'il me laisse. Pour les rendre concrètes, je prends le cas d'un logiciel chargé de préparer des devis. Ce devis sert ici à expliquer mes exigences.

Tout est-il borné ? Je fixe une limite aux attentes, aux quantités traitées et aux tentatives. Pour le devis, l'appel au fournisseur a un délai maximum et un nombre de relances limité. Si le tarif manque, le brouillon le signale et le calcul s'arrête à cet endroit. Le refus lui-même doit rester rapide, comme me l'a appris le programme du 19 septembre.

Puis-je tout reconstruire après coup sans deviner ? Je demande de conserver les éléments qui expliquent un résultat. Pour le devis, il faut retrouver la version du tarif utilisée, sa provenance, le moment où elle a été lue et l'étape où le travail s'est arrêté. La personne qui reprend le dossier dispose alors des faits nécessaires.

Le vital survit-il au non vital ? Je sépare les opérations indispensables des tâches qui peuvent attendre. Pour le devis, la conservation des lignes déjà saisies doit tenir même si la fabrication de son aperçu échoue. Une difficulté de présentation laisse le travail disponible pour la reprise.

Chaque décision laisse-t-elle une trace ? Je veux retrouver la raison d'un choix, en plus du résultat. Pour le devis, la trace explique quel tarif a été retenu, pourquoi une ligne a été refusée et qui a autorisé la suite. Cela permet de comprendre un montant et de corriger la bonne étape.

Chaque action peut-elle se défaire ? Je demande un retour à l'état précédent pour les modifications. Pour le devis, une révision garde le brouillon antérieur. L'envoi au client constitue une étape distincte, soumise à une validation humaine : un document déjà reçu engage autrement qu'un brouillon que je peux encore reprendre.

Le système limite-t-il les erreurs humaines ? Je cherche ce que le logiciel peut empêcher, plutôt que ce qu'une personne devra se rappeler. Pour le devis, un tarif absent bloque la finalisation. Je demande ensuite un essai avec le tarif présent et un autre avec le tarif absent. Le second doit montrer le refus attendu et la conservation du travail.

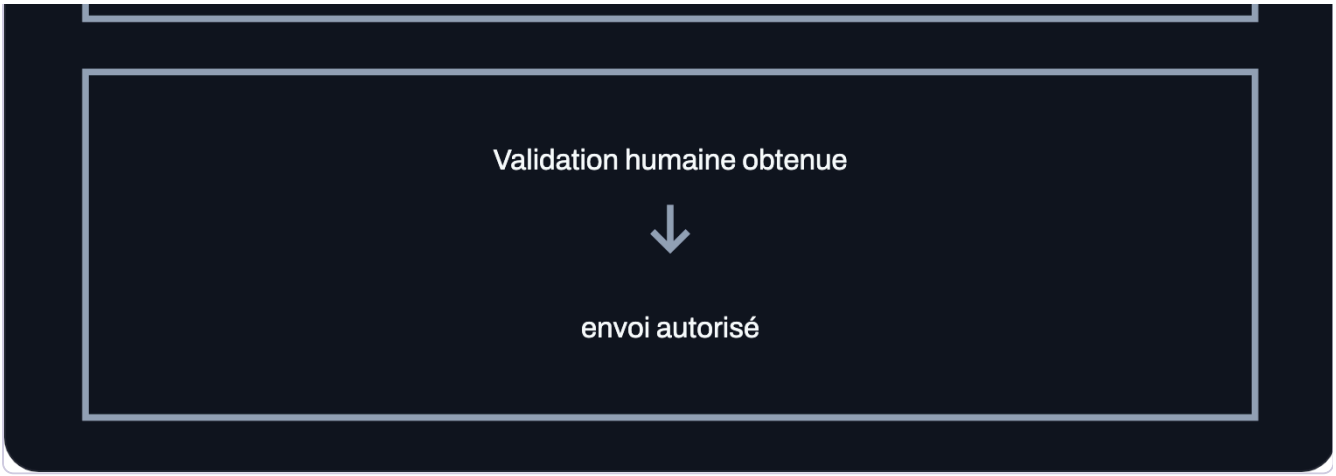
Le devis reste en brouillon tant que le tarif ou la validation manque.

© SAKIRA

Texte de Saray Hach, © 2026 Saray Hach, publié par SAKIRA SAS · sakira.fr
SAKIRA SAS · 6 rue d'Armaille, 75017 Paris · SIREN 108 493 933



Texte de Saray Hach, © 2026 Saray Hach, publié par SAKIRA SAS · sakira.fr
SAKIRA SAS · 6 rue d'Armaille, 75017 Paris · SIREN 108 493 933



Le devis reste en brouillon tant que le tarif ou la validation manque.

L'IA qui écrit et l'IA qui répond

Je distingue ensuite deux usages. Une IA peut écrire un programme qui suivra des instructions fixes. Elle peut aussi rester dans l'application et proposer une réponse à chaque demande. Dans ce second cas, je dois regarder le contenu de cette réponse en plus du comportement du programme.

Pour le devis, le total se calcule avec une formule. Si l'IA propose un tarif, je demande qu'il soit retrouvé dans la source autorisée. Un montant plausible reste à confronter au tarif réel. C'est le problème des réponses inventées par les IA.

Je choisis donc où l'IA intervient. Comprendre la demande exprimée avec les mots du client peut justifier son emploi. Additionner les lignes suit une règle fixe. Cette répartition prolonge ma réflexion sur le choix entre programme et IA.

Dans Ne fais pas ce que j'ai dit. Fais ce que je veux., je parlais du harnais : les règles, les permissions et les contrôles qui encadrent le travail de l'IA. Pour le devis, cela signifie que l'autorisation d'envoyer doit être contrôlée par le logiciel, même si l'IA propose de passer à la suite.

Ce que je cherche avant de confier le travail

Préparer ces contrôles demande du temps. Il faut définir les limites, organiser la reprise et entretenir les essais lorsque le programme change. Je regarde les conséquences d'une erreur pour choisir ce travail : refaire un brouillon et envoyer un montant faux à un client demandent des protections différentes.

Le cas du refus qui prenait 5,8 secondes m'a fait préciser une exigence que je croyais déjà posée. C'est ainsi que mes règles s'enrichissent : un comportement observé donne une règle précise et un contrôle à refaire. Quel premier travail d'une entreprise peut-on examiner de cette façon, en définissant aussi sa conduite en cas d'échec ?

Texte de Saray Hach, © 2026 Saray Hach, publié par SAKIRA SAS · sakira.fr
SAKIRA SAS · 6 rue d'Armaille, 75017 Paris · SIREN 108 493 933