

# Construire un MVP grâce à l'IA

De l'idée au prototype en quelques jours



Plus d'information ?  
[www.hesperans.com](http://www.hesperans.com)  
[info@hesperans.com](mailto:info@hesperans.com)

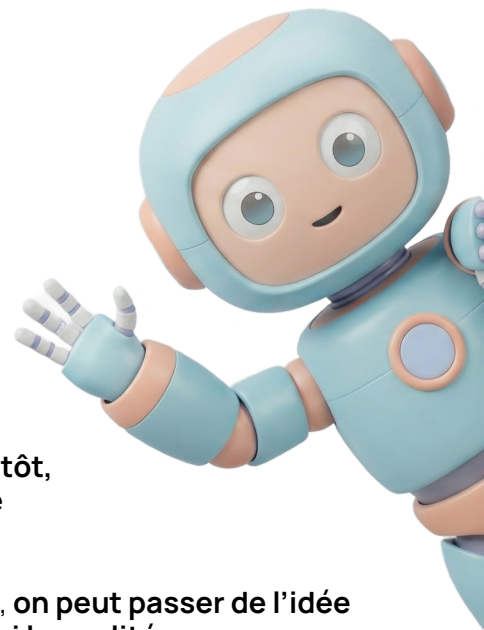
2025

# L'IA bouleverse la façon de lancer un projet digital

En quelques échanges avec une IA, il est possible de **clarifier une idée, imaginer des parcours, générer du code, préparer des maquettes et documenter un projet naissant.**

Cette nouvelle donne change la façon de démarrer, on peut **tester plus tôt, explorer plus largement, apprendre plus vite.** L'IA devient une matière première pour concevoir, plutôt qu'un simple outil de "productivité".

En combinant la vitesse de l'IA avec une approche structurée du produit, **on peut passer de l'idée au prototype en peu de temps, sans sacrifier la clarté, la cohérence ni la qualité.**



## Pourquoi ce livre blanc

Ce document s'adresse à celles et ceux **qui souhaitent lancer un projet digital plus rapidement**, s'appuyer sur l'IA pour gagner en logique tout en gardant la maîtrise de ce qui est conçu et développé.

Voici une méthode **pour passer de l'idée au prototype** en peu de temps, en préparant dès le départ **un produit capable d'évoluer.**

## Ce que vous allez apprendre

1

### Clarifier une idée avec l'IA

Structurer une intuition en problème clair, hypothèses et premiers scénarios d'usage.

2

### Générer un squelette technique

Amorcer une application complète avec des outils IA (front, back, API), exportable vers votre écosystème (GitHub, base de données, hébergement).

3

### Installer une boucle d'itération

Utiliser GitHub, Supabase, Vercel et l'IA pour enrichir le produit avec un workflow de développement lisible, partagé et durable.

Comprendre



Cadrer



Mettre en place



Prototyper et itérer

## 3 changements de posture pour tirer le meilleur de l'IA



### La vision donne le cap, l'IA amplifie.

Plus votre intention est claire, plus l'IA propose des pistes pertinentes et actionnables au plus précis de vos idées



### Le prototype devient un laboratoire.

Une app générée en peu de temps sert à tester, apprendre et ajuster avant de consolider l'architecture et l'expérience.



### L'IA devient un membre de l'équipe.

Elle aide à rédiger, explorer, simuler et documenter, pendant que l'équipe humaine garde le discernement, les choix et les arbitrages.

## Etape 1 : Clarifier le projet avec l'IA

Avant de parler maquettes, stack technique ou roadmap, un projet digital commence par une question simple : **quel besoin avons-nous ou problème voulons-nous résoudre, pour qui et dans quelles situations ?**

## L'IA est particulièrement utile à ce moment-là \_\_\_\_\_

Elle aide à transformer une intuition en formulation claire, à faire émerger les enjeux, à explorer plusieurs angles et à préparer une première matière de travail pour l'équipe : problématiques, scénarios d'usage, questions à poser aux utilisateurs.

**L'objectif n'est pas de "laisser l'IA décider", mais d'utiliser sa capacité à générer, reformuler et structurer pour accélérer la phase de cadrage.**



### De l'idée au problème formulé

Un projet démarre souvent par une phrase courte :  
« On pourrait faire une app pour... »

C'est un bon déclencheur, mais c'est insuffisant pour guider une conception, prioriser des fonctionnalités ou engager un budget.

En quelques allers-retours, on obtient des 1ères spécifications que l'équipe peut partager, challenger et enrichir, au lieu de rester dans des intentions vagues.

## Une méthode simple en 3 temps

- ✓ **Poser le contexte**  
Décrire votre organisation, vos enjeux du moment, les contraintes principales (temps, budget, réglementation, systèmes existants).
- ✓ **Explorer les usages**  
Demander à l'IA de proposer des situations-types, des profils d'utilisateurs et des scénarios d'usage. Cela permet de visualiser rapidement à quoi pourrait ressembler la vie "avec" le produit.
- ✓ **Formuler les hypothèses clés**  
Lister, avec l'aide de l'IA, ce qui doit être vrai pour que le projet ait de l'impact tel que les comportements attendus, bénéfices, signaux de succès.



**"J'ai une idée..."**  
(Vague, émotionnel)



**Le Filtre IA**  
(Challenge, structure)



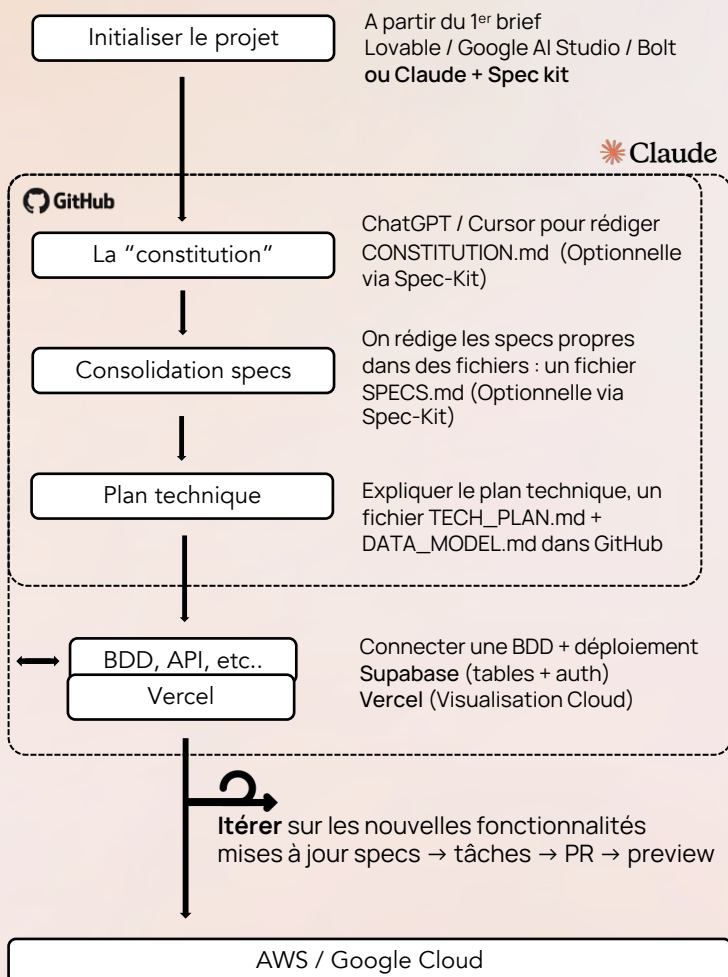
**Hypothèses claires**  
(Actionnable, testable)



## Etape 2 : Générer le squelette du projet avec l'IA

Une fois le projet clarifié, l'enjeu c'est de poser une base exécutable. Un squelette sert à rendre visibles les décisions qui vont structurer le produit : **l'architecture, le modèle de données, les contrats, les conventions, et le chemin de déploiement.**

### Le développement piloté par les spécifications



## Structurer ses spécifications

### Constitution (les règles du jeu)

Les principes non négociables : qualité, compatibilité, sécurité, performance, style, contraintes produit.

### Specs fonctionnelles

Les fonctionnalités, scénarios utilisateurs, critères de succès, cas limites.

*Ici, on évite volontairement la techno : on décrit la valeur.*

### Plan technique

Choix d'architecture, stack, découpage, décisions clés, modèle de données, intégrations.

### Design system « minimal »

Typo, couleurs, spacing, radius, boutons, champs, cards.

*Une UI cohérente dès la première verticale.*

### Tâches

Une séquence d'exécution, structurée, parallélisable, qui permet de suivre l'avancement et de déléguer proprement.

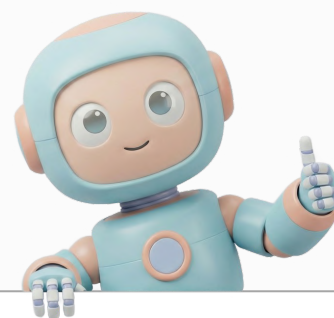


### Votre boîte à outil

<https://github.com/github/spec-kit>

### À la fin de cette étape, vous disposez :

- ✓ d'une application minimale qui tourne,
- ✓ connectée à une base de données et à un dépôt de code,
- ✓ une structure propre pour servir de base aux prochaines itérations.



## À retenir

Un bon cadre ne ralentit pas un projet.  
Il donne à l'IA une mémoire, un langage commun et une trajectoire.

C'est ce qui permet d'aller vite **sans perdre la cohérence.**

### Etape 3 : Installer une boucle d'itération

La boucle d'itération est un cadre simple qui relie **ce qu'on décide**, **ce qu'on construit**, **ce qu'on observe**, et **ce qu'on améliore**.



**Avec l'IA, cette boucle peut aller plus vite** mais à condition de rester pilotée, versionnée, testée, et lisible par l'équipe.



## Claude Code le coéquipier qui travaille pour votre projet

Il lit les fichiers de votre projet, modifie le code, crée des composants, met à jour la documentation. Vous le pilotez avec des consignes simples, et il produit une version concrète, exécutable.

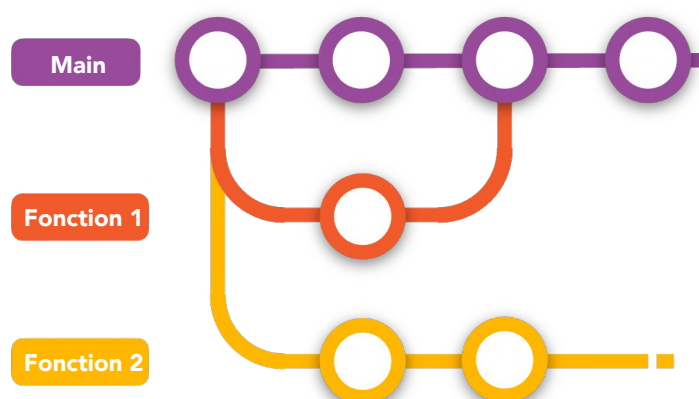
Cette façon de travailler change tout, au lieu d'une conversation, vous construisez un projet qui progresse par petites améliorations cumulatives.

## Les branches Une zone de brouillon sécurisée

Une branche, c'est une copie de travail. La version principale du produit reste stable, pendant qu'on teste une nouvelle fonctionnalité « à côté ».

Chaque branche devient un endroit où tester une hypothèse (d'UX, de parcours ou de fonctionnalité).

Quand l'essai est satisfaisant, on l'intègre dans la version stable. Et on recommence, fonctionnalité après fonctionnalité, dans une cadence maîtrisée.



### Essayer sans risque

On construit la nouvelle fonctionnalité dans un espace d'essai. La base reste stable pendant toute l'exploration.



### Voir avant de décider

Chaque essai doit pouvoir être visualisé simplement, via un lien de prévisualisation, avant d'être intégré à la version stable.



### Intégrer proprement

L'amélioration est intégrée à « main ». La cohérence est protégée par quelques règles simples, dont le respect du design system.

## Le prototype devient un terrain d'observation

Le prototype n'est plus seulement une démo. C'est un petit laboratoire qui montre comment les utilisateurs se comportent vraiment.

### Comment les gens utilisent ce que l'on construit ?

**Laquelle de nos idées fonctionne le mieux pour les utilisateurs ?** On veut juste savoir si les personnes vont au bout du parcours, où elles s'arrêtent, et quelles actions reviennent le plus souvent.

**On peut tester plusieurs hypothèses d'UX ou plusieurs variantes de fonctionnalités**, chacune dans sa propre branche, comme autant d'essais en parallèle.

Le but est d'ajouter un outil léger qui observe l'usage réel, et qui s'intègre naturellement au reste du processus.

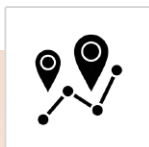


#### On ne mesure pas "tout ce qui bouge".

On choisit ce qu'on veut observer en fonction des questions qu'on se pose.

- ✓ **Parfois, ce sont des signaux très globaux** combien de personnes vont au bout d'un parcours, combien abandonnent en cours de route.
- ✓ **Où plus fin, dans une logique d'UX research** où les personnes hésitent, quel bouton elles ignorent, quelle variante fonctionne mieux.

Dans tous les cas, l'idée est la même lier chaque hypothèse (de parcours, d'interface, de fonctionnalité) à un petit nombre de signaux concrets, que l'outil de mesure pourra suivre dans le temps.



#### Ok et comment on le fait ?

Pour garder les choses simples, on s'appuie sur un seul outil de mesure d'usage.

Par exemple :  **PostHog**

Un outil d'analytics produit qui montre comment les gens utilisent votre application

On le "branche" une fois sur l'application, dans la pratique, cela revient à ajouter une clé fournie par l'outil et quelques lignes de code. C'est typiquement le style de mise en place que l'on peut confier à Claude code en lui expliquant ce que vous voulez suivre.

À partir de là, chaque fois qu'une personne réalise une action importante, l'application envoie un signal à PostHog, et vous pouvez observer l'usage réel de vos hypothèses, branche par branche.



**Mesurer l'usage, c'est donner au prototype une voix.**



# De l'idée au produit qui apprend

Au fil des pages, on a vu comment l'IA peut s'intégrer dans un projet digital, **non pas comme une baguette magique, mais comme un accélérateur bien cadré.**

**On part d'une idée à clarifier, on lui donne un squelette technique, on installe une boucle d'itération, puis on mesure l'usage pour apprendre de ce qui se passe réellement côté utilisateurs.**

## Étape 1 – Clarifier le projet avec l'IA

*L'IA aide à challenger, reformuler, structurer.*



On part d'une idée brute et on la transforme en problème formulé, avec des contextes d'usage, des profils d'utilisateurs et des hypothèses claires.

## Étape 2 – La stack technique

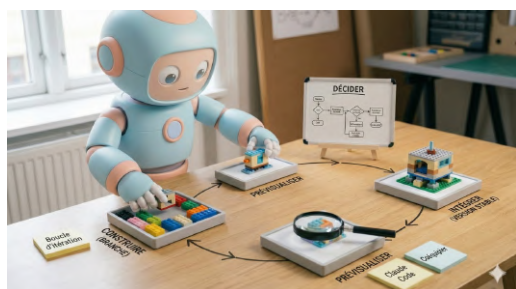
*Il s'agit de donner un premier corps au projet.*



Le squelette relie l'architecture, les données et les premières règles du jeu pour passer de l'intention au produit qui tourne, même de façon minimale.

## Étape 3 – la boucle d'itération

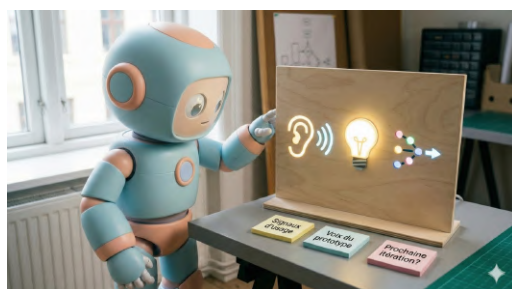
*Tester dans une branche, prévisualiser, intégrer.*



Claude Code et Github deviennent des coéquipiers pour implémenter rapidement de nouvelles fonctionnalités sans casser la version stable.

## Étape 4 – Mesurer l'usage

*Les signaux d'usage pour comparer des hypothèses.*



Les signaux d'usage suffisants pour comparer des hypothèses d'UX ou de fonctionnalités, et orienter les prochaines itérations.



# HE.SPERANS

Invest in innovation, build the future



Vous avez une **vision à concrétiser**,  
une **organisation à aligner**  
ou un **produit à optimiser** ?

**Parlons-en.**

**Vincent LAPPARTIENT**



[www.hesperans.com](http://www.hesperans.com)



[info@hesperans.com](mailto:info@hesperans.com)