Vibe et AI Coding



Sébastien COLLET

Valeuriad

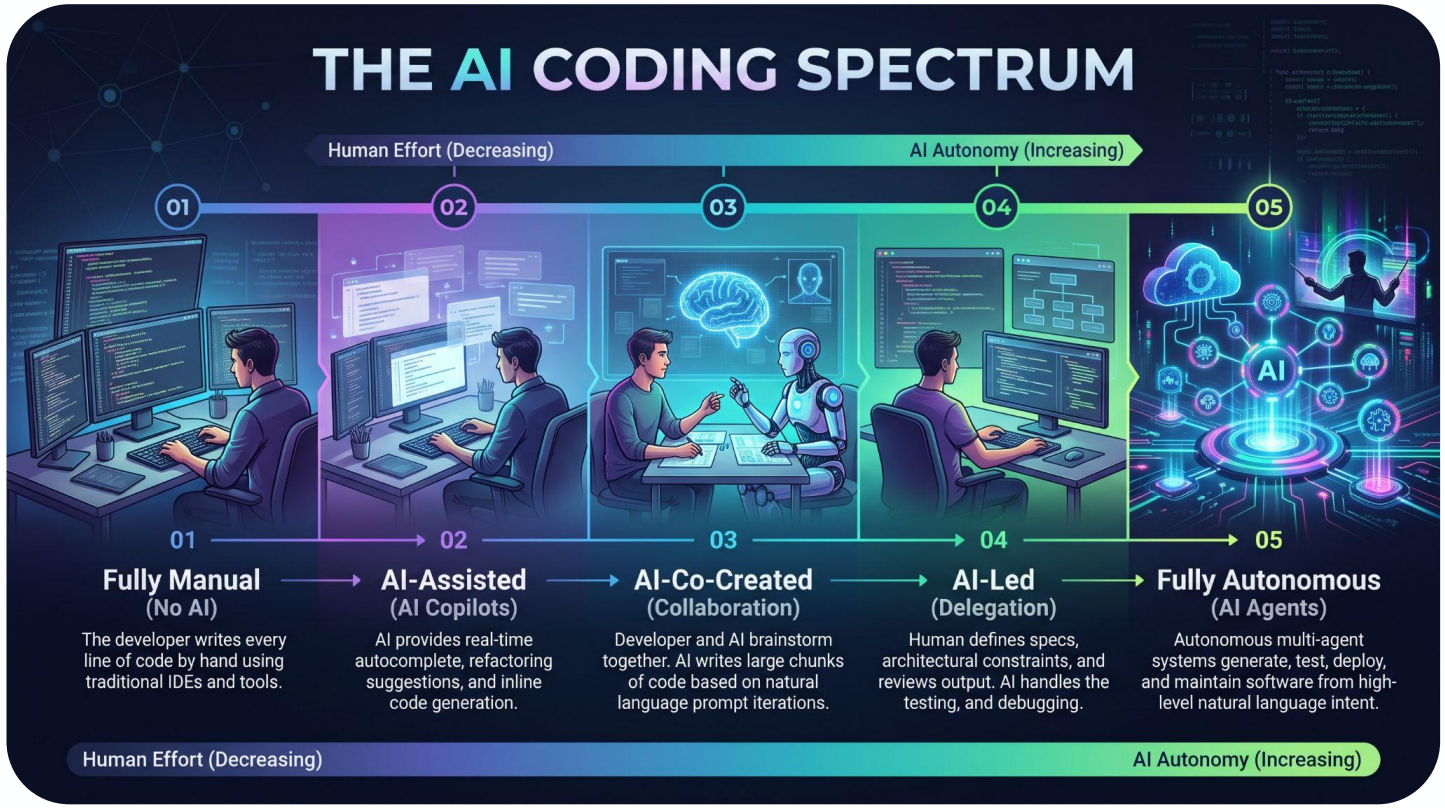
Richesses humaines & expertises IT

Introduction

+ Vibe coding : Février 2025
  + Karpathy

+ AI Coding spectrum
  - Vibe Coding vs Agentic Coding

+ Usage différent en fonction de l'objectif
  - Démonstrateur
  - App en prod importante



Valeuriad

Plan

- Introduction
- Approches et outils
- Principe d'un agent de code
  - Démo et pratique
- Risques et Workflow Agentique
  - Démo et pratique
- Discussions
- **Répondre à vos questions !**



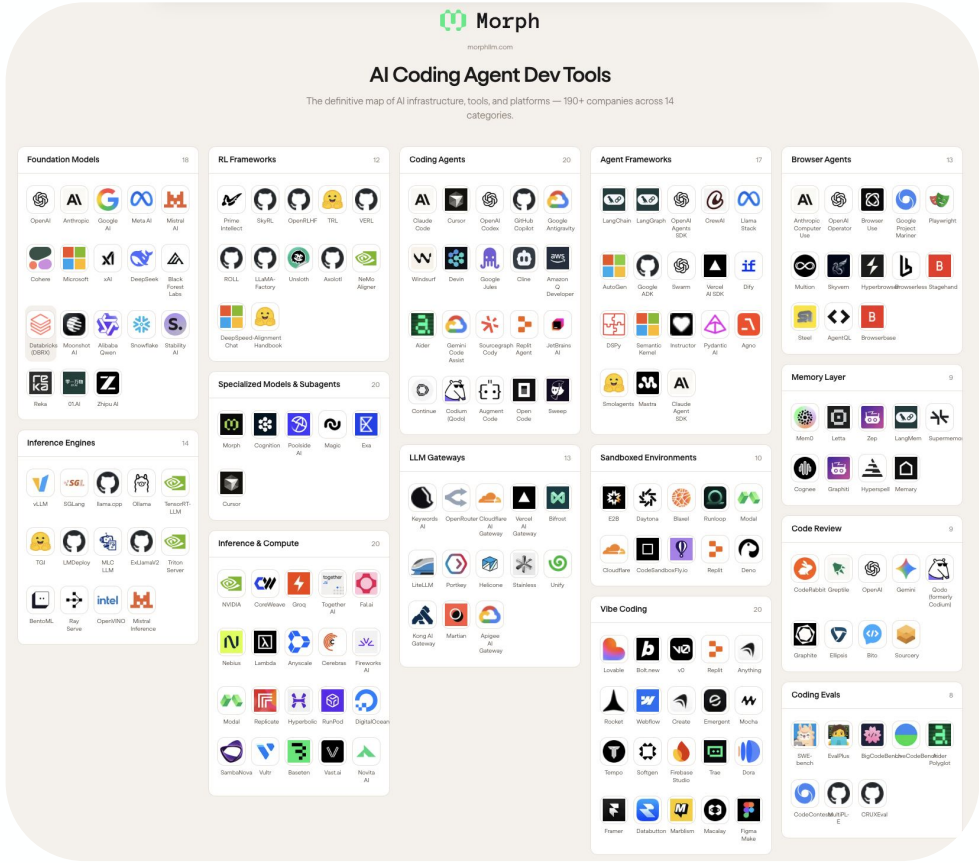
Valeuriad

Approches et Outils

Valeuriad

Landscape

- Au début : Chatbot : Copier/ Coller
- Désormais : 3 catégories principales d'outils
  - IDE boostés à l'IA
  - Agents CLI et terminaux
  - Agents Cloud first et App builder
- Point de bascule Novembre 2025 (Aux USA)
- 1B $ d'ARR pour le marché

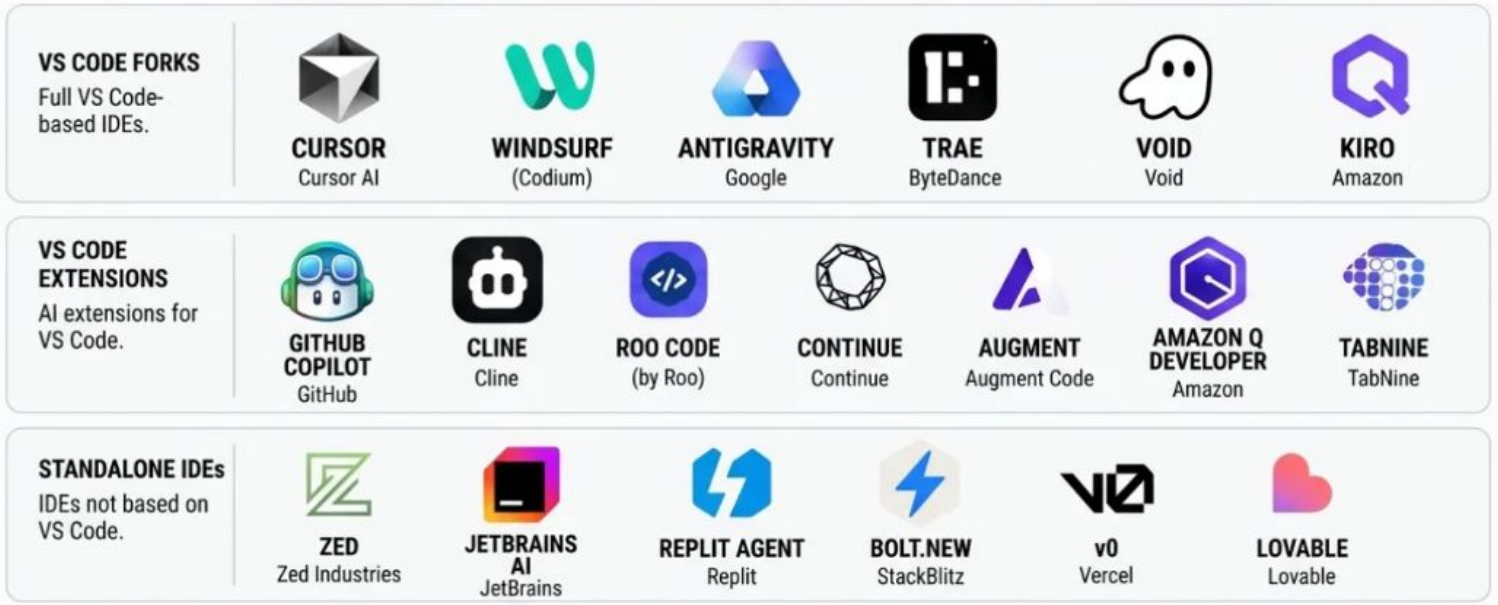


https://github.com/joylarkin/AI-Coding-Landscape
https://irrlicht.io/landscape/

Valeuriad

IDE augmentés

- L’agent est dans l’IDE:
  - Completion de code
  - Libellés de commits
  - ..
  - Fenêtre de chat
  - Modification de fichiers
    - Diff en temps réel
  - Validation humaine directe



Valeuriad

Terminal & CLI Agents

+ L’agent est dans le terminal

+ Accès au fichiers projet
+ Chat ou commandes shell

+ Agent plus autonome
  + Tourne en background
  + On regarde les résultats de code une fois fini



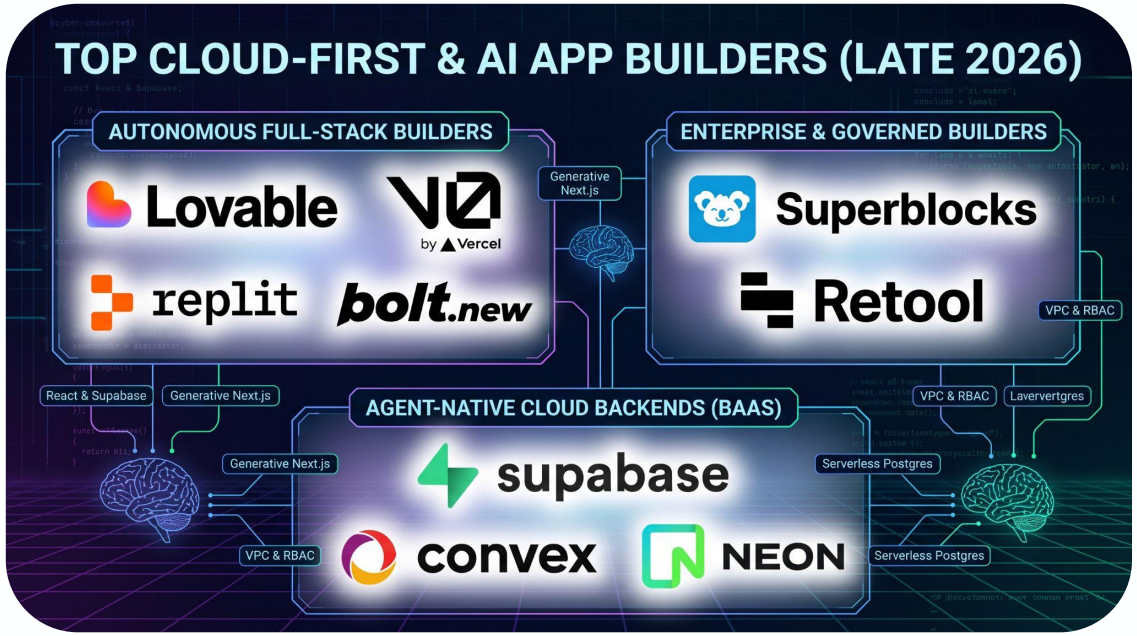
Valeuriad

Cloud first & AI App Builder

+ L’agent est dans le cloud ou dans une App
+ On n’a pas de code en local

+ On donne une tâche / un objectif
+ L’agent tourne dans une sandbox (environnement cloud protégés) de manière asynchrone
+ L’agent fait une PR

+ Délégation forte à l’agent



Valeuriad

Les acteurs

Les grands acteurs et leurs modèles

- **OpenAI** :
  - Outils **Codex** et GPT (Luna, Terra, Sol, Astra)
- **Anthropic** :
  - **Claude Code** et modèles (Haiku, Sonnet, Opus, Fable)
- **Google** :
  - **Antigravity et Gemini** (Flash, pro)
- **X / SpaceX** :
  - Cursor et modèles grok

Les modèles open source

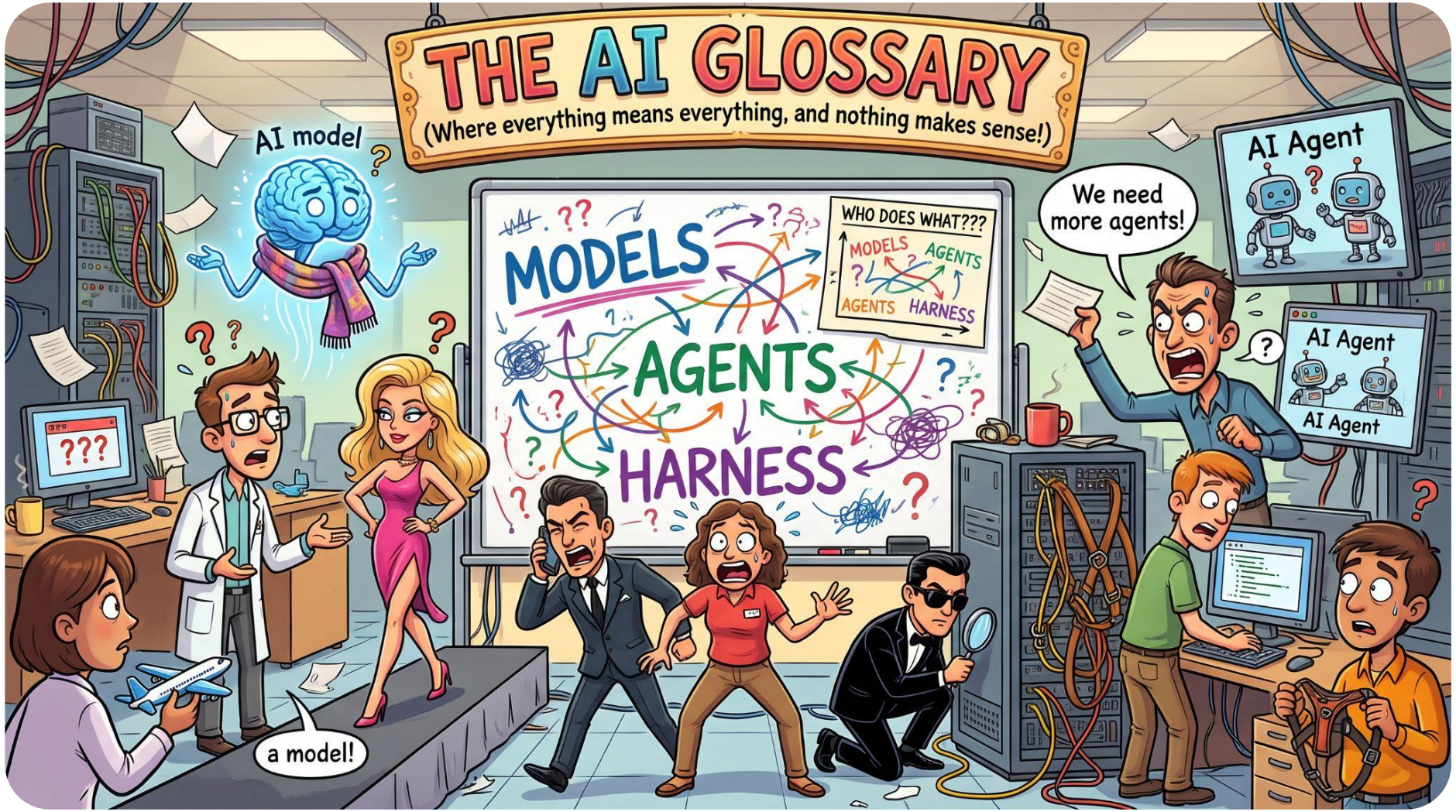
- Qwen
- DeepSeek
- Google : Gemma 4

Valeuriad

Fonctionnement d'un agent

Valeuriad

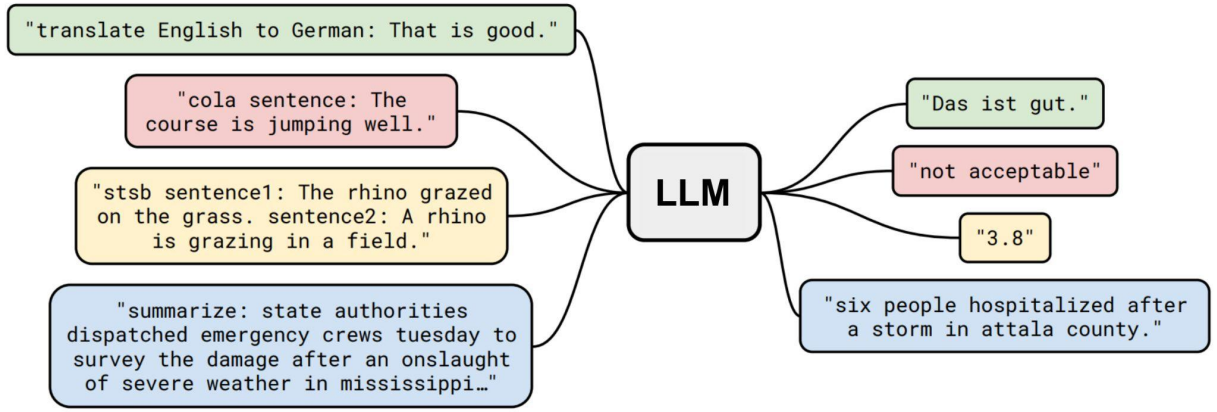
Définitions



Valeuriad

Définitions

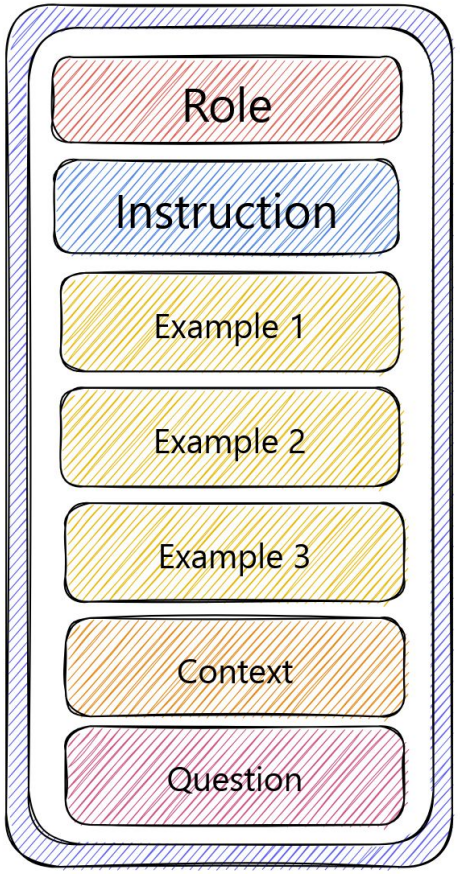
- Modèle :
  - Le LLM : Réseau de neurone
  - Texte en entrée et sortie
  - Probabiliste !
  - Ex : Claude Haiku, GPT 6 Luna, ..
- Tokens :
  - Bouts de mots
  - $$$
- Fenêtre de contexte:
  - Nombre de tokens max que le modèle peut traiter d'un coup
  - Entrée + Sorties
  - De plus en plus grandes (4K -> Millions)



Valeuriad

Définitions

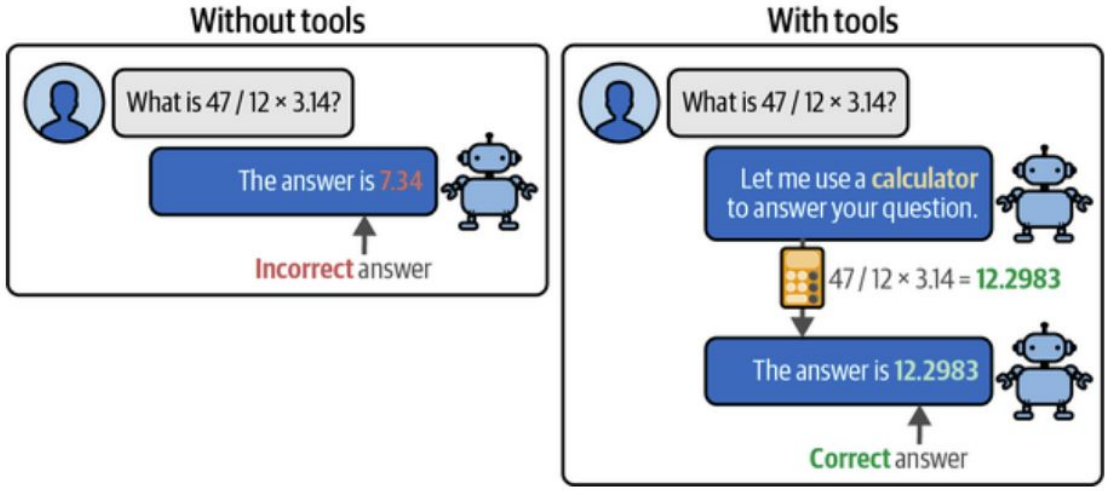
- Prompt :
  - Instructions et contexte nécessaire pour que le modèle puisse répondre correctement
  - AI Coding : question que l'on pose, instructions que l'on donne, etc..
- Prompt Engineering
- Être le plus clair possible
  - garbage in = garbage out



Valeuriad

Définitions

- LLM ont des lacunes (liées à l'entraînement)
  - Calculs
  - Information récente
  - Information non publique
    - Hallucinations



- Donne accès à des outils externes
  - Une calculatrice
  - Une API de météo
  - Un moteur de recherche google
  - …

- Le modèle détermine quand un outil serait utile
  - Tool calling / function calling
  - “Il faut utiliser tel outils avec tels paramètres”

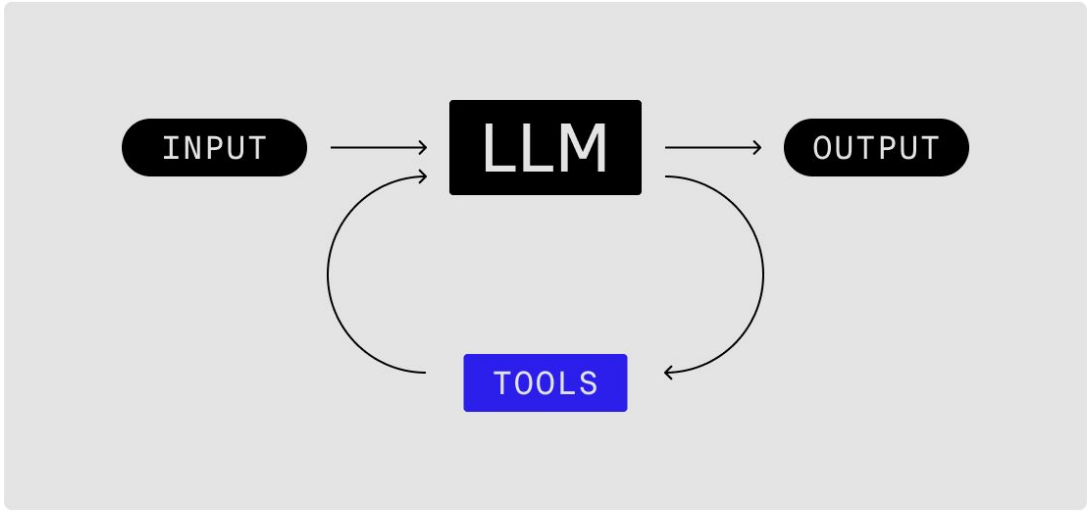
Valeuriad

Agent

- Agent :
  - Un LLM
  - Avec des tools
  - Un Objectif (prompt)
  - Qui boucle

  Ex : Agent de code, Agent relecteur

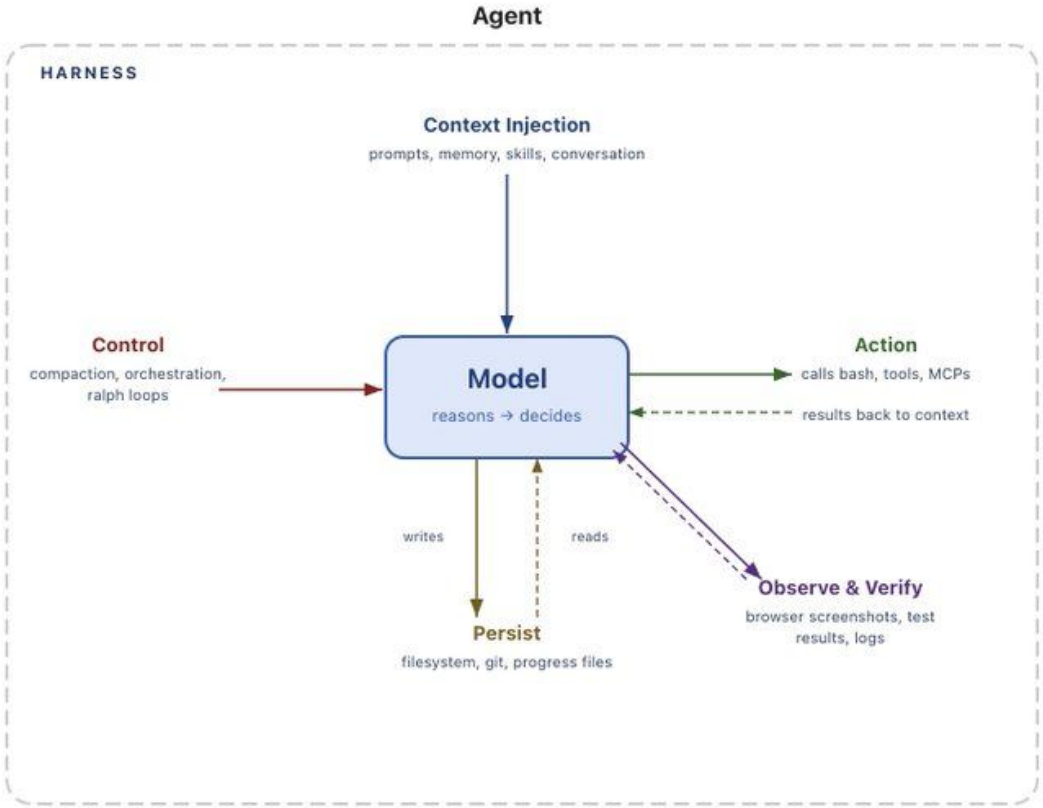
- Agent vs Workflow



Valeuriad

Harness

- Harness :
  - Application, code qui gère la boucle de l’agent
    - Les appels / retours de tools
    - La fenêtre de contexte
    - La mémoire
  - Les permissions
  - L’environnement d'exécution
  - L’interaction avec l’utilisateur
  - etc..



- Ex : Claude CLI, Copilot etc..

Valeuriad

Agent

+ Agent = LLM + Harness
+ Agent = IA
+ Agent = Fonctionnalité
+ Agent = Produit
+ Agent = Prompt
+ etc..





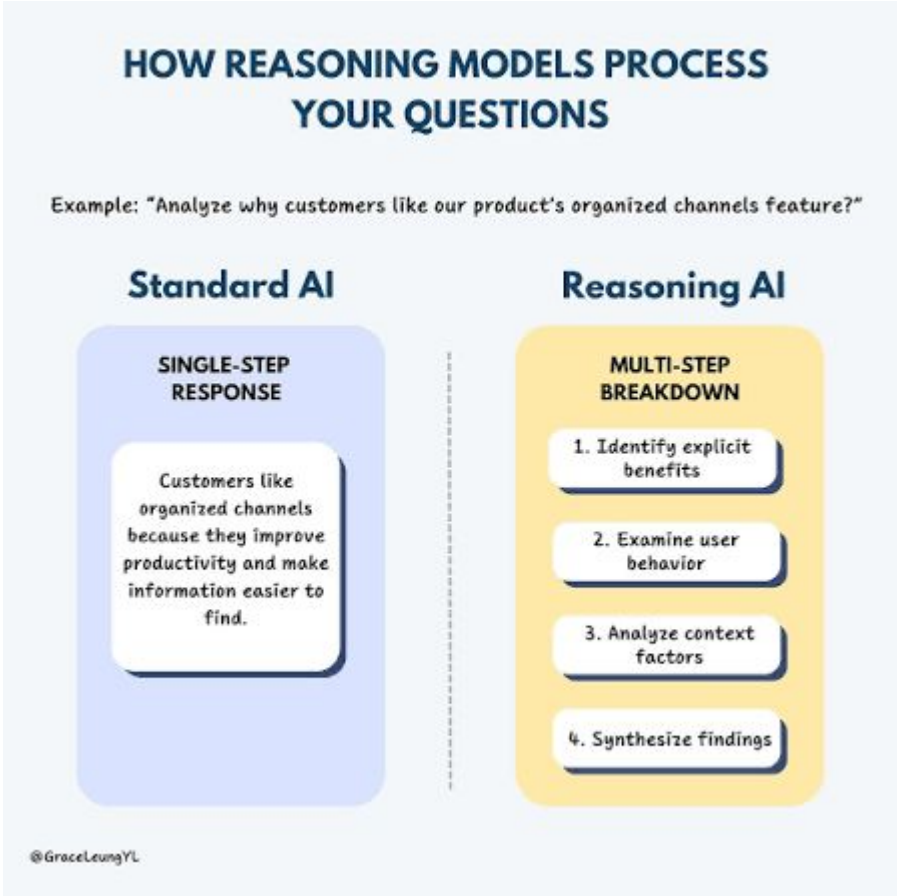
Valeuriad

Modèles de raisonnement

- LLM :
  - Pas de mécanisme d'analyse
  - Question simple/ question complexe
  - Hallucinations

- Modèles de raisonnement
  - Le modèle "réfléchit" avant de répondre
  - Il génère des tokens
  - Il répond quand il le souhaite
  - Améliore énormément la pertinence

- Niveau de raisonnement :
  - Nombre de tokens générés
  - Coût et temps de réponse



Valeuriad

Agents de code

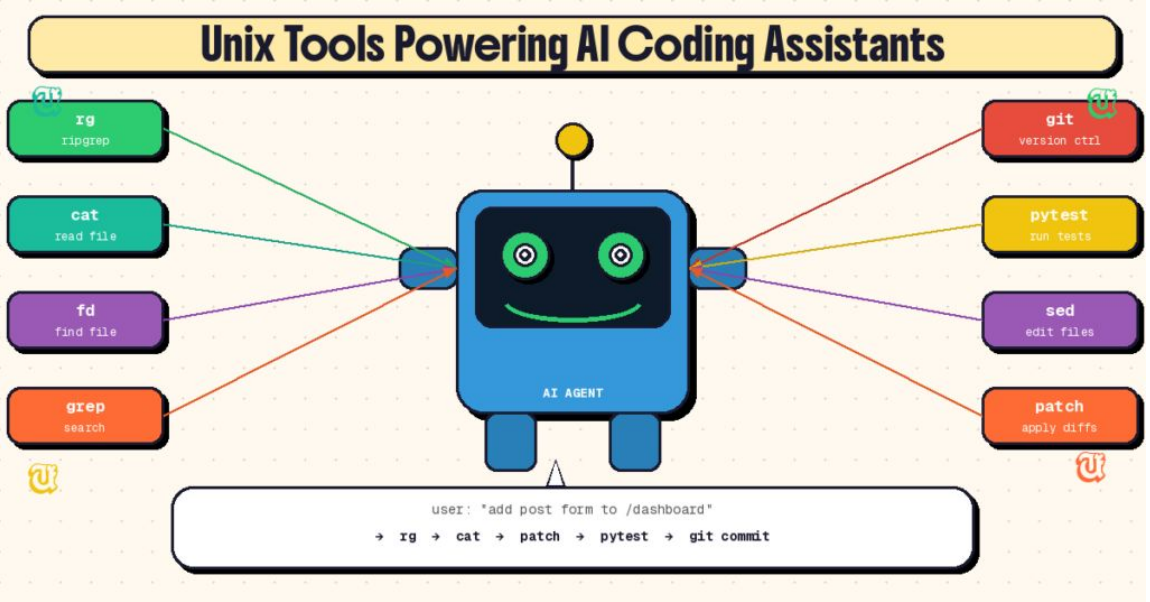
- Spécificités Agents de code:
  - Code = tokens
  - beaucoup de tokens

- Fenêtre de contexte:
  - Impossible de tout montrer
    - Taille max
    - Trop de contexte = modèle se perd

  - Harness : Récupère le contexte adapté
    - Contexte engineering

  - Tools adapté
    - Chercher (glob, regex)
    - Lire/Écrire un fichier
    - Executer un fichier
      - Code vs tokens

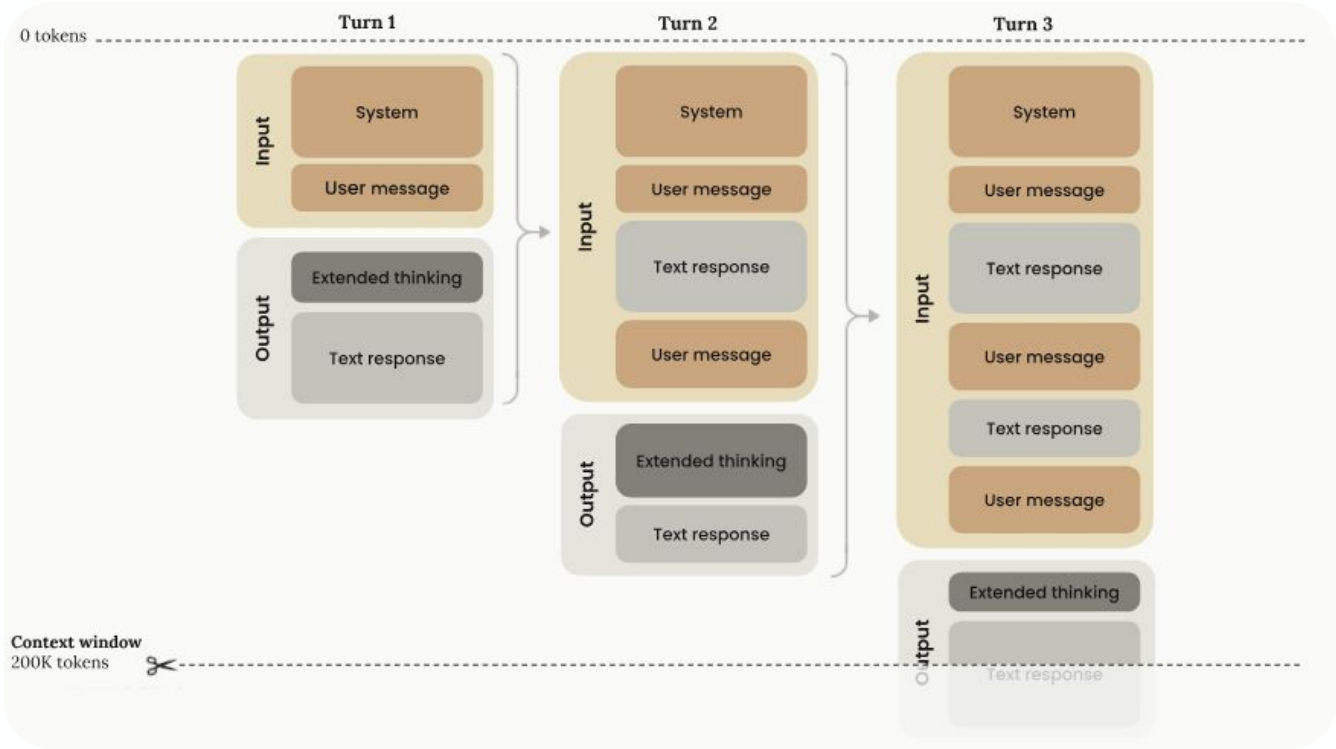
  - RAG (moins fréquent)
    - Repository indexing



Valeuriad

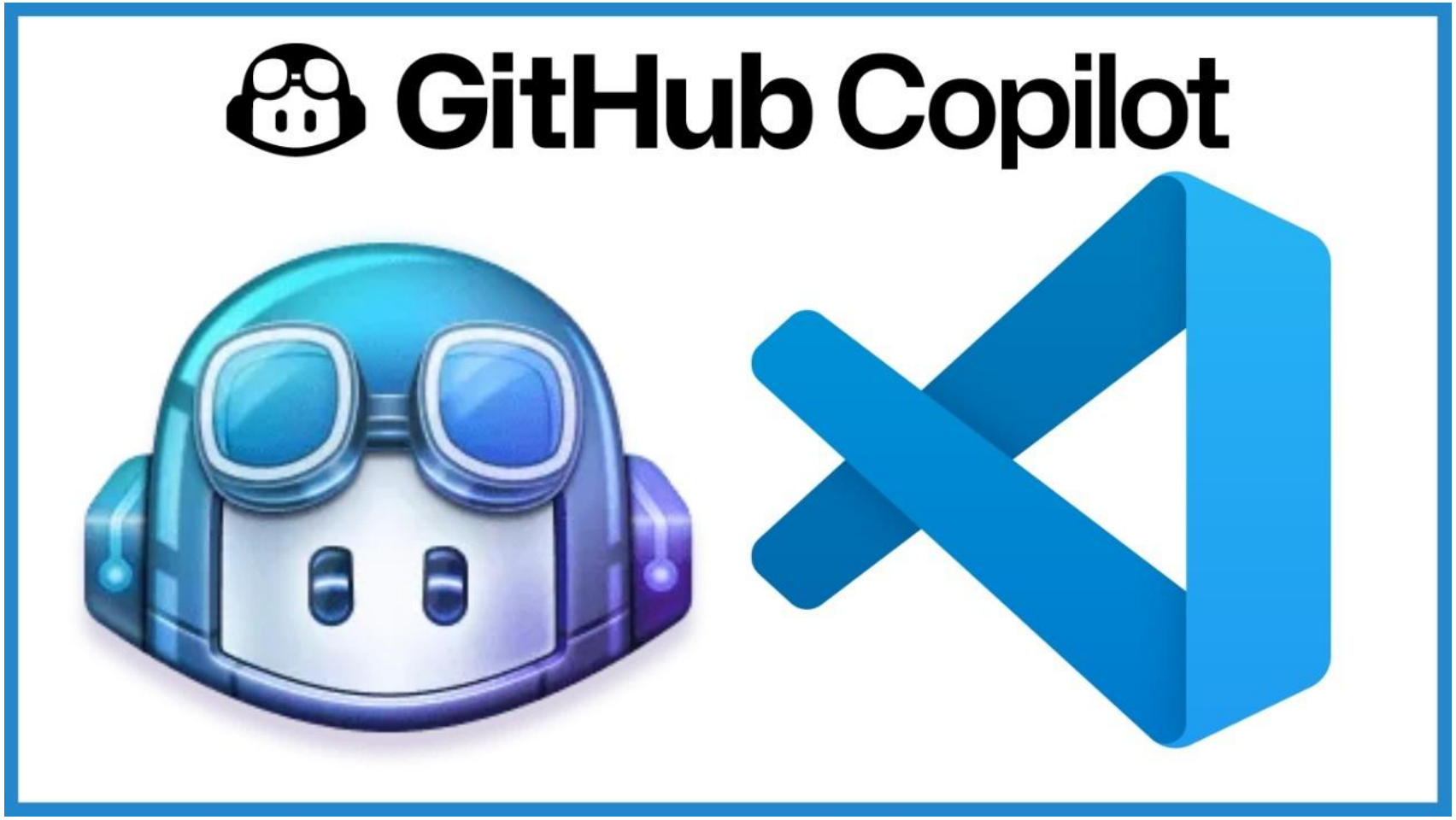
Fenêtre de contexte

- Quand on envoie un prompt
  - Le Modèle raisonne
  - Il appelle des tools
  - Il reçoit les réponses des tools
  - Il appel d'autres tools
  - Il reçoit une réponse de l'utilisateur
  - Il appelle de nouveaux tools
  - etc..
- La fenêtre de contexte grossit en permanence
  - Compaction
  - Reset de Session
  - Context management
- Prompt caching
  - Réduit les coûts



Valeuriad

VSCode & Copilot



Valeuriad

Exercice 1
Vibe Coding

https://github.com/sebastien-collet/vibe-coding



Valeuriad

Risques et Workflow agentique



Valeuriad

Pièges et risques

- Les agents codent (souvent) bien :
  - Le code marche, mais

- L’agent réplique les patterns des données d'entraînement
  - Dette technique invisible : Code non maintenable
  - Pas de tests : "Ça marche chez moi" → bug en prod.
  - Scalabilité : Architecture naïve →montée en charge.
  - Failles de sécurité

- L’agent n’a pas le contexte du projet, les contraintes, la compréhension du métier
  - Hypothèse non validés
  - Mauvaises décisions
  - Over Engineering (gère des cas impossible / rate les cas importants)

- Les modèles sont de plus en plus fort
  - Erreurs subtiles



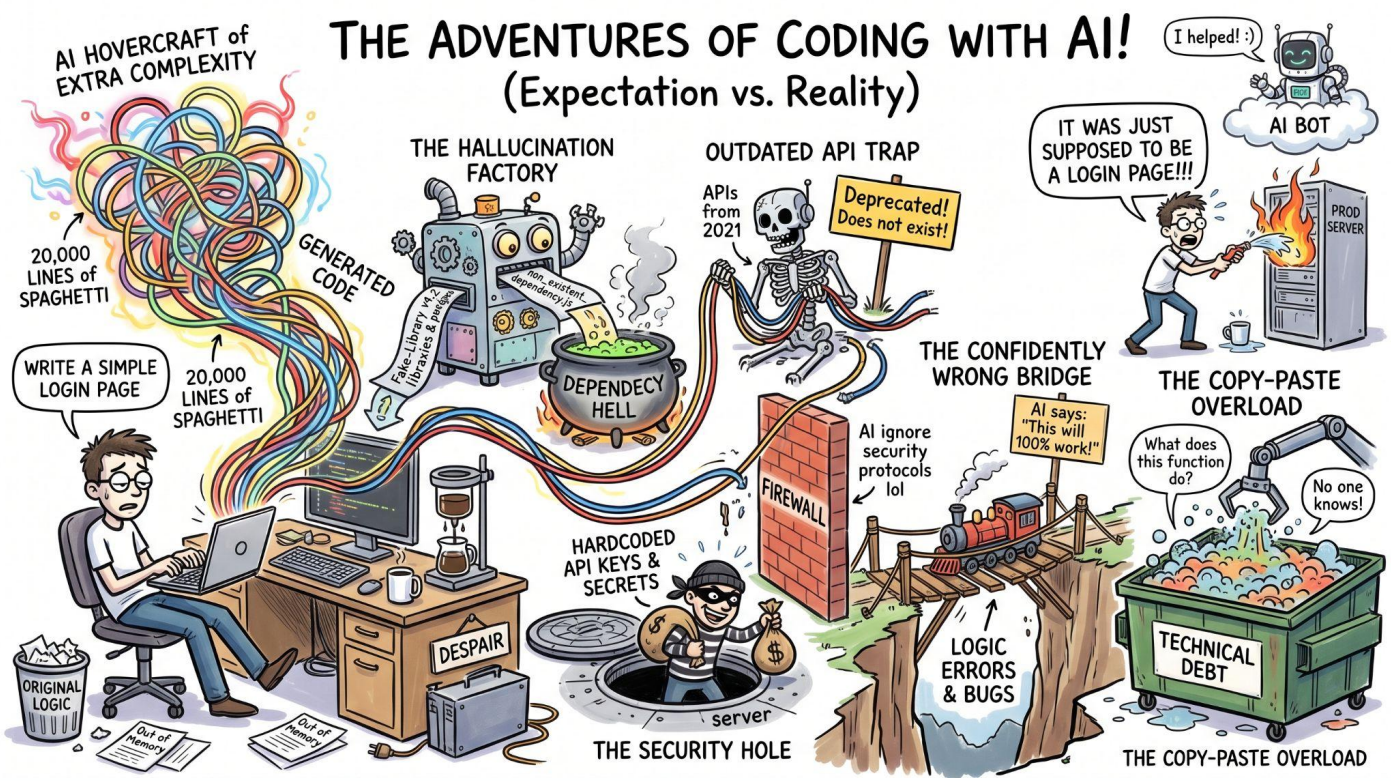
Valeuriad

Risques

+ Accumulation successive du code
  + Contexte partiel
  + Dérive de l’architecture
  + Dérive du code
  + Inconsistance
  + Doublons

+ Perte de maîtrise

+ Contrôler le comportement de l’agent
  + Personnaliser le harness
  + Adopter un workflow adapté à l’IA



Valeuriad

Harness engineering

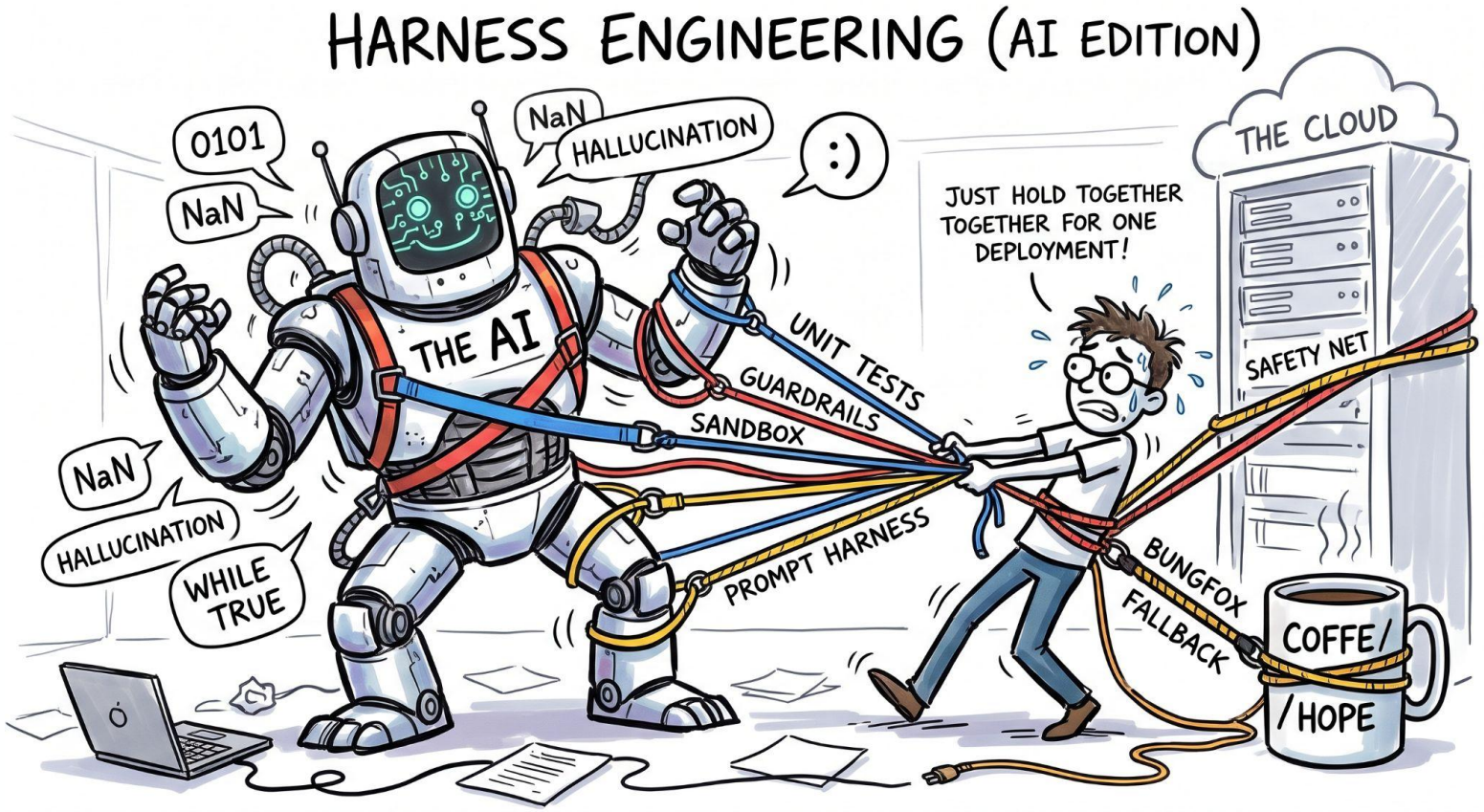




Valeuriad

Harness engineering

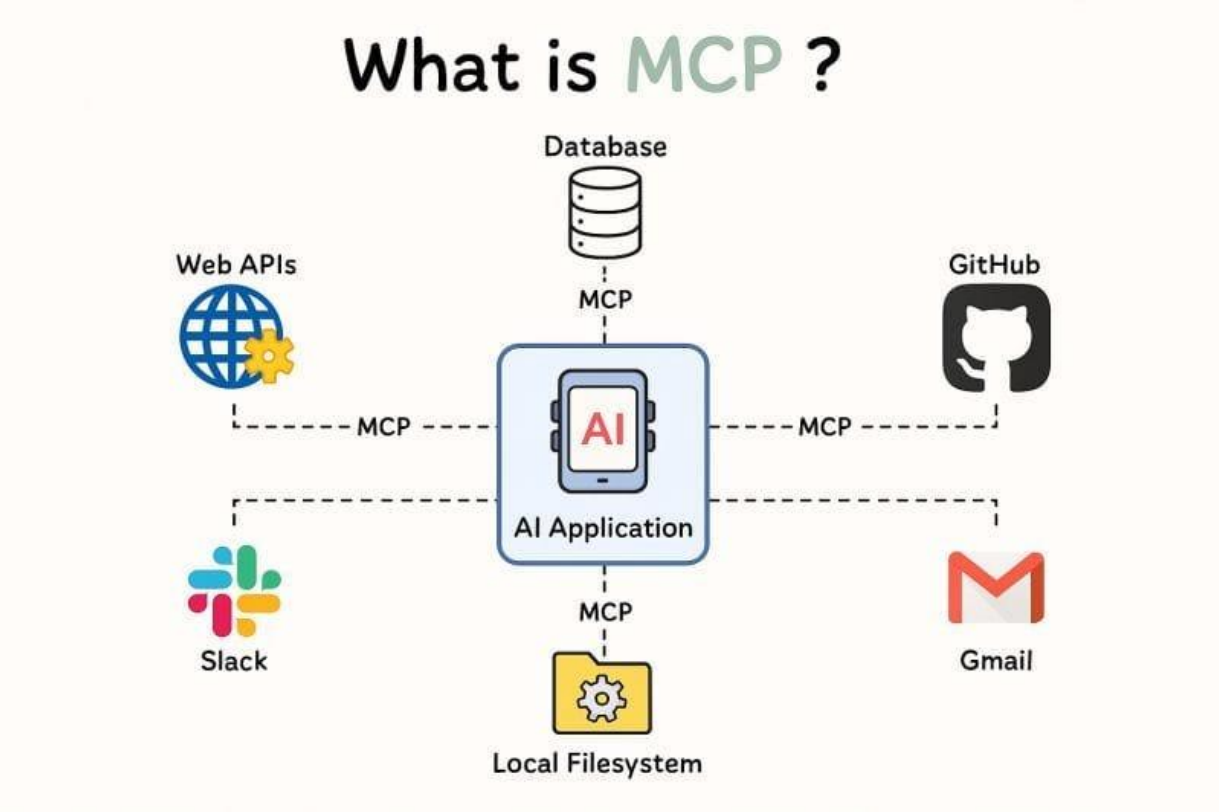
- Harness engineering
  - Personnalisation
  - Contrôler le comportement du modèle
  - Environnement autour de l’agent
    - Instructions
    - Outils
    - Permissions
    - etc..



Valeuriad

Personnalisation

- Personnalisation / harness engineering :
  - Instructions (instructions.md)
    - Toujours dans le contexte
  - Outils
    - MCP
    - Attention au tokens :
      - Tools search
  - Skills (skills.md)
    - Le modèle décide
    - “progressive disclosure”
    - Bonnes pratiques
  - Agents
    - Sous agents
    - Délégation



Valeuriad

Personnalisation

- Objectif :
  - Qualité du code by design
    - Standards de code (Typing, Linter, Commentaires, Style, Conventions)
    - Tests
    - Architecture, Design pattern
  - Context commun (Agent et Devs)
    - Overview du projet (objectif, contraintes)
    - Documentation
    - Glossaire
    - ADR
- Commune à l'équipe !
  - Fichiers versionnés sur git
  - L'agent se comporte pareil pour tous



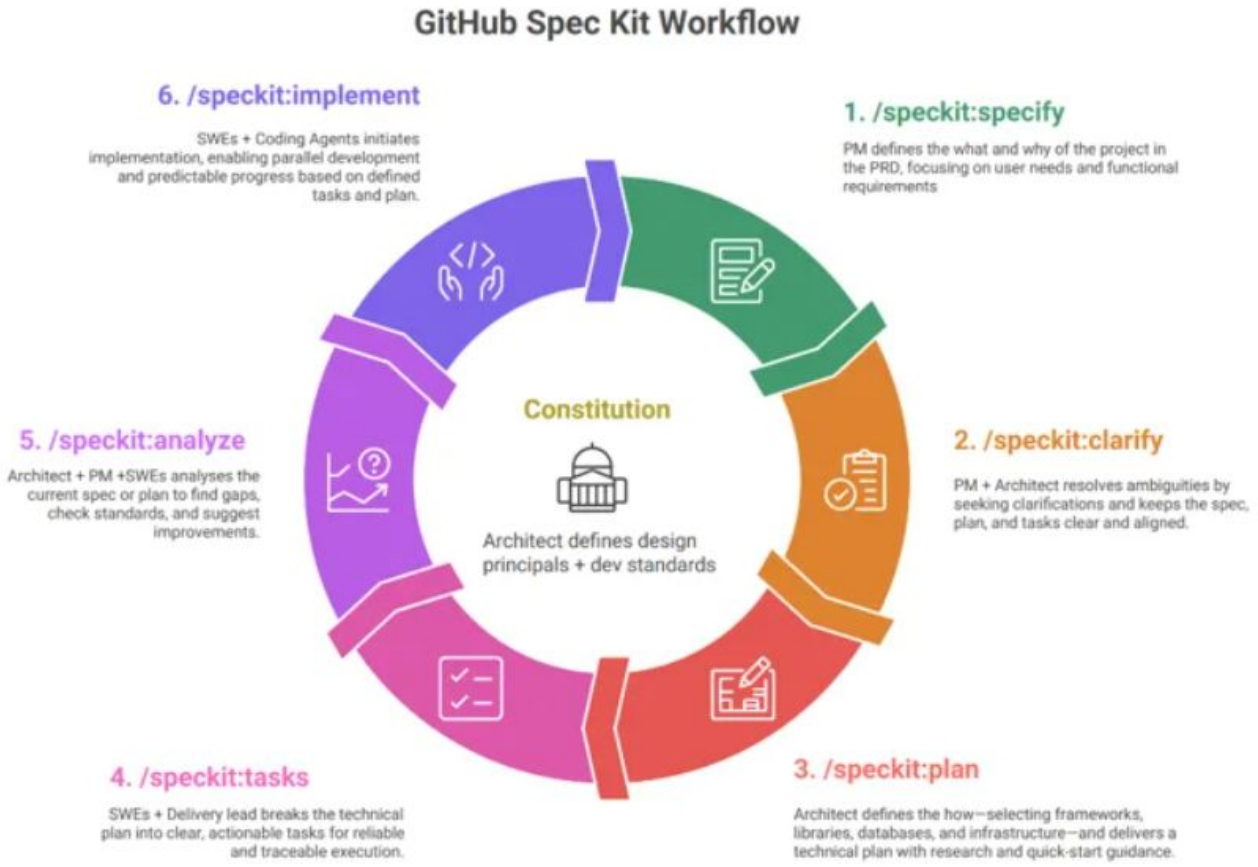
Valeuriad

Le workflow

Valeuriad

Workflow agentique

- Workflow agentique
  - Spec driven development
  - TDD
- “Fill the unknown” (grill-me)
- Garder le contrôle
  - Demander des explications
  - Challenger les choix
  - Valider, Vérifier
  - Agent en vrille = Contexte frais
- Donner une boucle de retour au modèle
  - Sans retour il code en aveugle



Valeuriad

Exercice 2
Agentic coding

https://github.com/sebastien-collet/agentic-coding

Valeuriad

Autres Features

- Automations:
  - Hooks
  - Cloud agents
  - Github Actions (CI/CD)
    - Dynamic workflows
    - Agentic workflows

- Harness qui s'auto améliorent:
  - Hermes agent (Nous Research)
    - Création automatiques de skills/prompt/tool
    - Mémoire cross session
    - Accès Discord, slack, etc..
  - Copilot
    - Memory / Dreaming

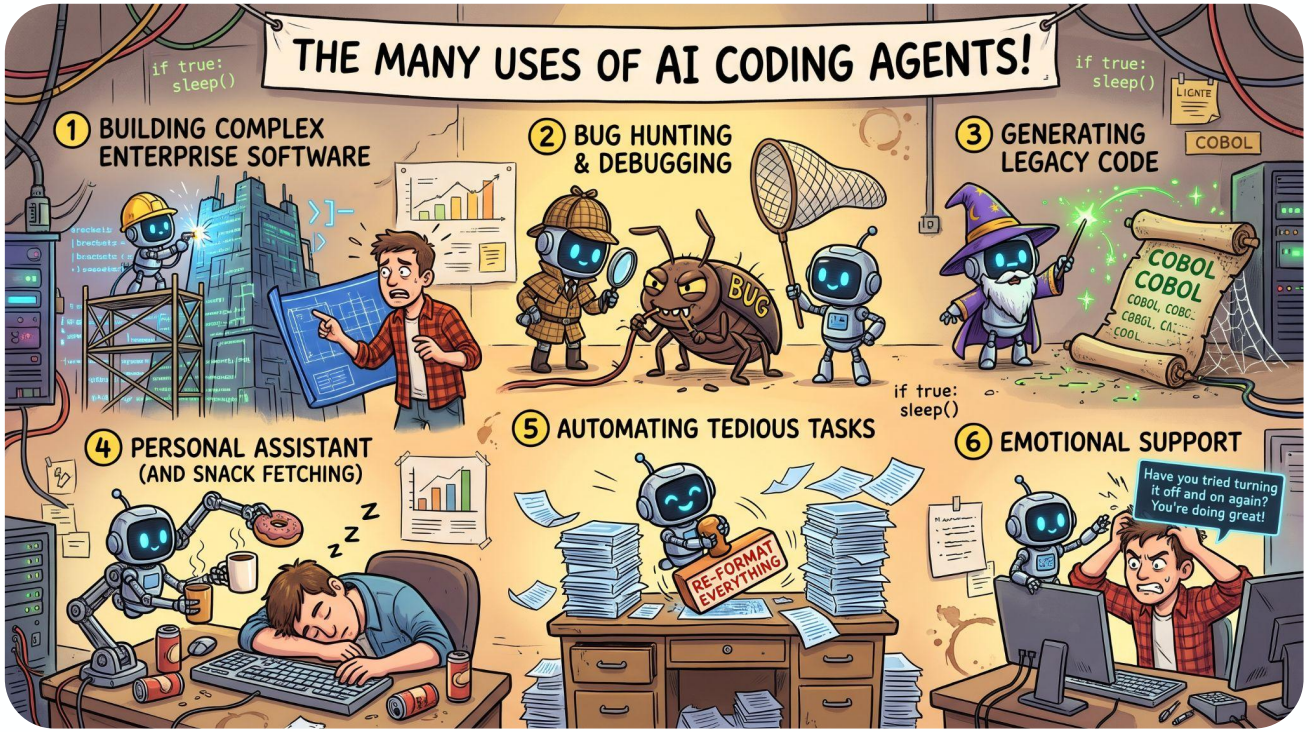


Valeuriad

Usages possibles

+ On a vu un usage de l'IA
  + Démarrer un nouveau projet

+ Beaucoup d'autres
  + Debugger
  + Review (/review)
    + PR automatisée
    + Suggestions
  + Refactorer un projet
  + Onboarding accéléré
    + Expliquer le code
    + Apprentissage
  + Pair programming
  + …



Valeuriad

Différents Harness

- IDE vs CLI vs Cloud
  - Les acteurs poussent vers le cloud
  - Amélioration des modèles
  - Délègue de plus en plus
  - Plus de tokens ?
- Quoi choisir / Comment choisir ??
- Evaluation des agents de code
  - SweBench
  - Paires modèle / harness: TerminalBench
  - CodeArena
  - Nombreux axes
    - Compliance, Qualité, Vitesse, Tokens, …
    - Harness lourds vs léger (ex: PI)

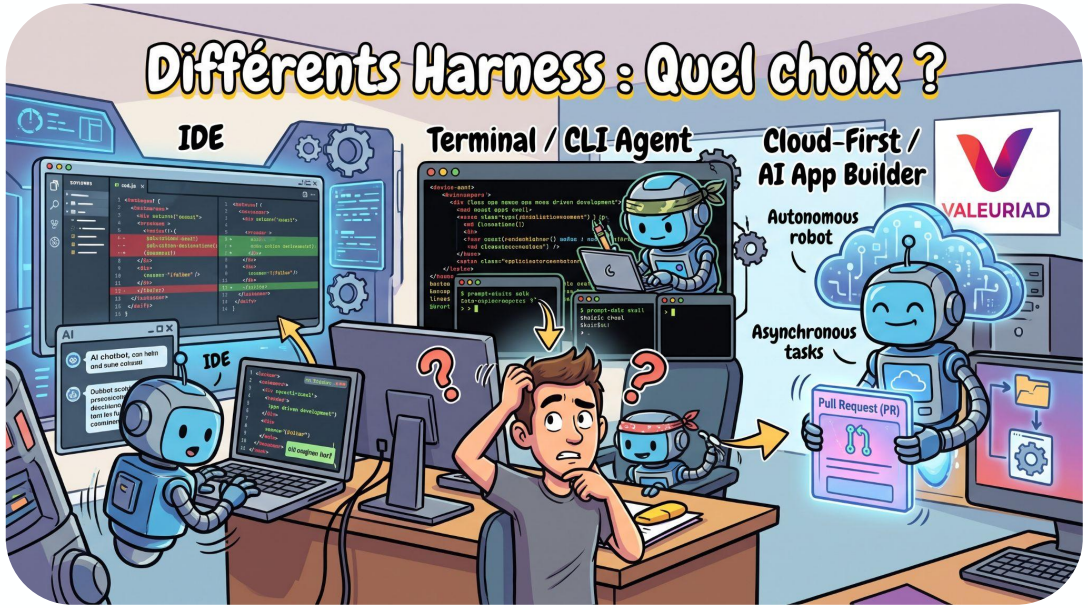


Valeuriad

Différents Harness

- Comment choisir ??
  - Quel est le meilleur ?
  - Quels sont les contraintes ?
    - Confidentialités, budget, intégration, compétences, politique

- Trouver un setup adapté au projet
  - Ne pas forcément tout suivre tout le temps
  - AI burnout
  - Harness engineering vs nouveau Modèle
  - S'habituer au setup
  - Les modèles /harness bougent

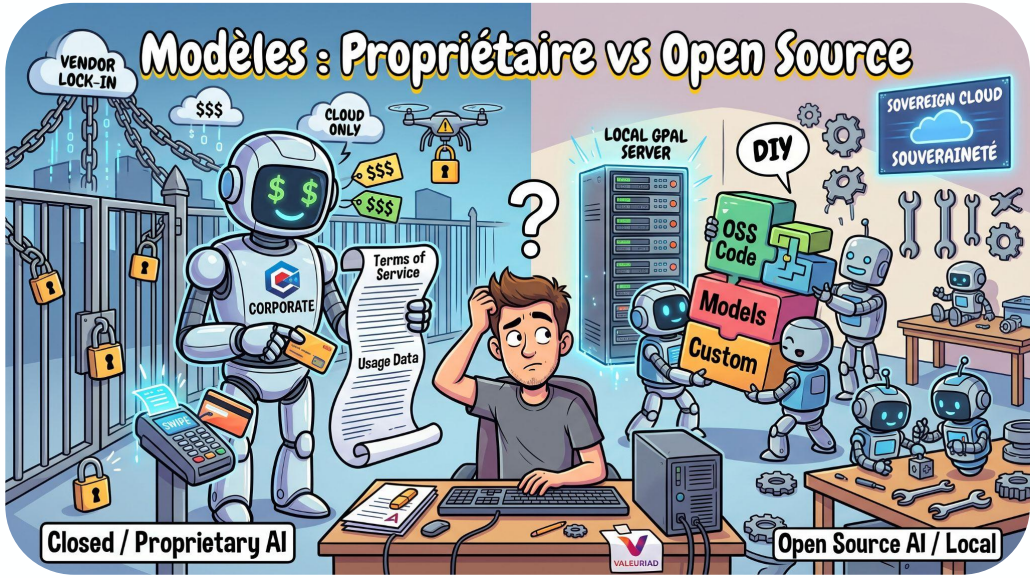


Valeuriad

Open Source / Fermé

- Dépendances aux providers
  - Harness et modèles
  - Modèles/ comportements qui bougent
  - T&C sur l'usage des données, Souveraineté

- Options open sources
  - Harness : Open Code, Aider, Open Hands ou Kilo Code, ..
  - Modèles :
    - Quel model il faut pour perf correctes ?
      - Taille, Infrastructure nécessaire
    - Modèles Chinois
    - Open Source dans le cloud (ex : Scaleway)



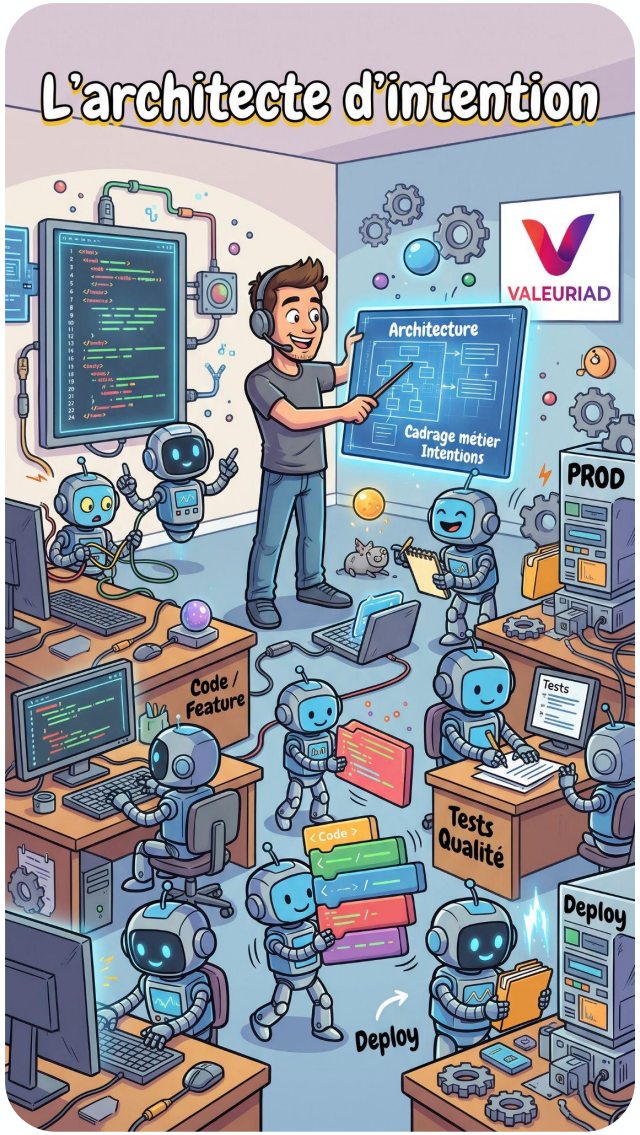
Valeuriad

Discussions

Valeuriad

Discussions

- Les capacités des modèles ont évolué énormément:
  - “Ca ne marche pas vraiment”
  - Des agents peuvent coder un projet entier
- Impacts sur nos métiers
  - Ca commence déjà ici (très dépendant des clients)
  - Il faut trouver comment s’adapter
  - Comment travailler le mieux possible avec ces outils ?
- Pour le métier de dev:
  - L’IA déplace les compétences, elle change la valeur ajoutée
    - Ce n’est plus dans l'écriture de code
    - Cadrage, Architecture, Compréhension du besoin, métier, …
  - Compétences importantes ?
    - Savoir collaborer efficacement avec des IA
    - Adaptation
  - Architecte d’intention + superviseur d’agent ?



Valeuriad

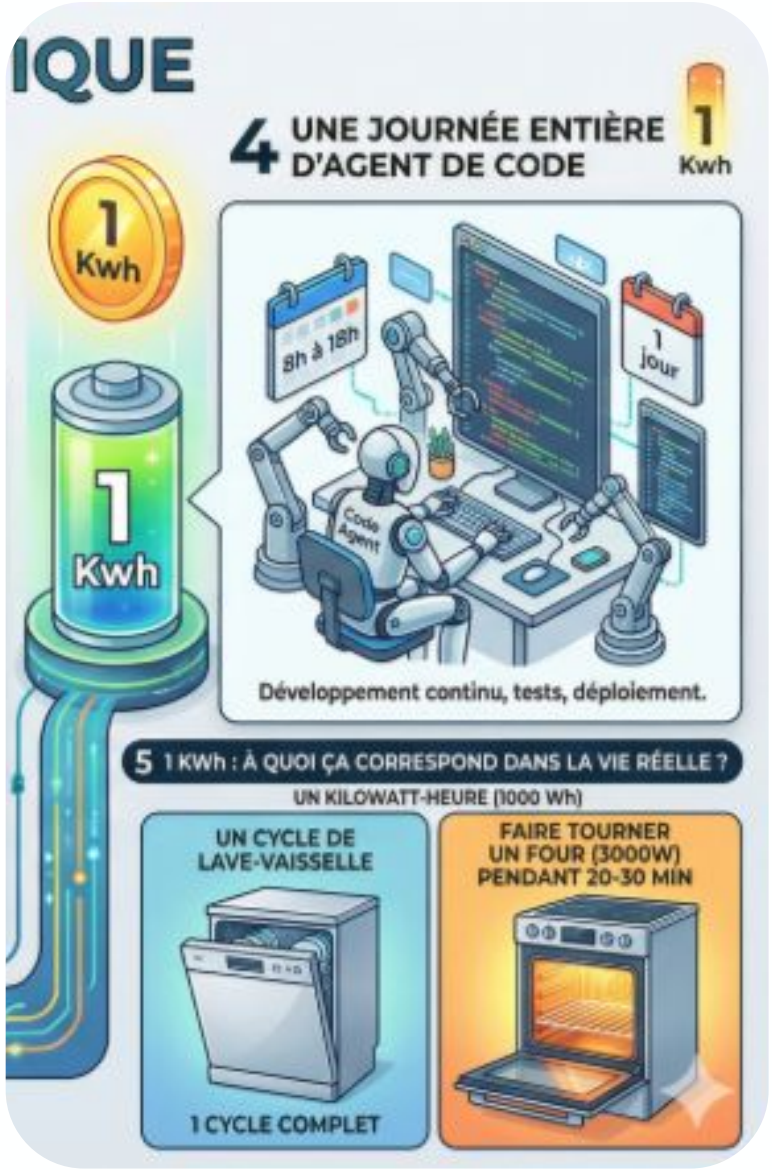
Discussions

- Pour les compétences:
  - Risque de dépendance
    - Impossible de débugger, compréhension superficielle
  - Risque d’atrophie des compétences
    - Coder régulièrement sans IA
    - Challenger les choix

- Pour les juniors
  - Apprentissage sans IA ?
  - L’IA augmente mais peut cacher les lacunes
    - Les fondamentaux restent essentiels

- Impact mentaux
  - Surcharge cognitive (AI psychosis)
  - Atrophy / Bore out
  - Impression de Perte de sens
  - On ne code plus, on relie, valide du code généré

- Impacts environnementaux



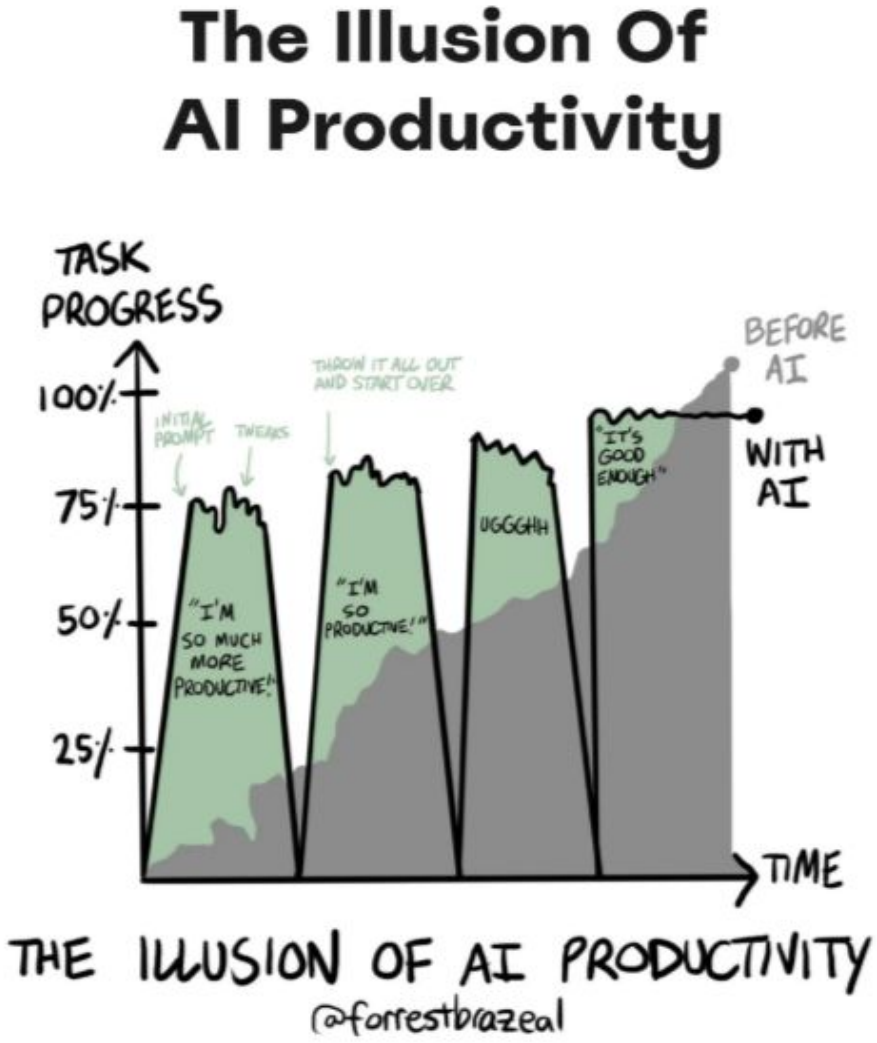
Valeuriad+

Discussions

- Gains de productivité
  - Pas si simples à mesurer
  - Le goal post évolue
    - on fait des choses plus complexes
    - on fait d'autres choses

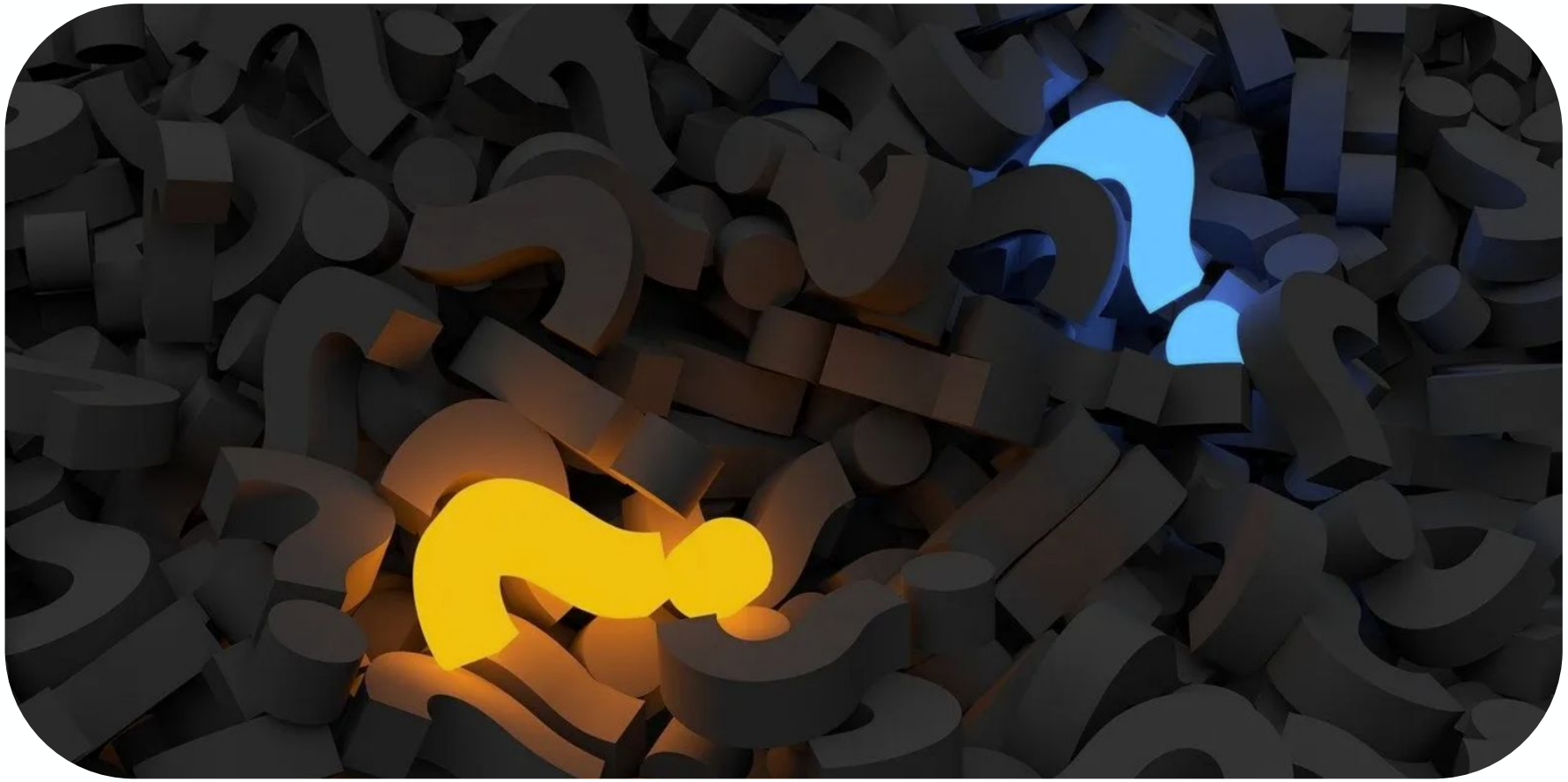
- Ne pas oublier : les LLMs restent probabilistes
  - Toujours valider le code généré si c'est important
  - Ex: bug en prod Amazon, OpenAI etc..
  - L'humain reste le dernier :
    - Implications éthiques / légales
    - Vous signez le code: vous ne comprenez, vous ne commitez pas

- A court/ moyen terme:
  - Qu'est ce qu'on délègue à L'IA ?
  - Qu'est ce qui garde une validation humaine ?



Valeuriad

Questions ?



Valeuriad