AI TOOLS WIKI · MÉMO

# BMAD Method *mémo*

Les skills à connaître pour utiliser **BMAD Method** dans Claude Code ou Codex. BMAD structure le travail en spécifications, stories et sessions de build, et adapte la quantité de planification à la taille du changement.

**Famille** Méthodes de développement · **Éditeur** BMad Code · **Licence** MIT
**Note du wiki** 13/15 · **Commandes vérifiées le** 25 septembre 2026
**Fiche, notes et avis** aitoolswiki.viellard.fr/outils/methodes-de-developpement/bmad-method

## Installation

| | |
|---|---|
| `npx skills add bmad-code-org/BMAD-METHOD` | Via le CLI skills (dans le projet) – inclure bmad, bmod-method et bmod-core-tools |
| `/plugin marketplace add bmad-code-org/bmad-plugins` | Claude Code : ajouter la marketplace, puis installer bmad-method et bmad-core-tools |
| `codex plugin marketplace add bmad-code-org/bmad-plugins` | Codex : même marketplace, depuis le terminal |
| `npx bmad-method install --directory . --modules bmm --tools claude-code --yes` | Alternative documentée dans le guide de démarrage |

Prérequis : Node.js, Git, Python 3 et uv.
Ensuite, dans l'agent : demander au skill `bmad` de lancer `bmad setup`.

## Commandes d'orientation

| | |
|---|---|
| `bmad setup` | Configure (ou rafraîchit) le runtime BMAD du projet. À relancer après chaque mise à jour. |
| `bmad status` | Affiche les versions installées et ce qu'il faut lancer ensuite. |
| `/bmad-help` | Recommande la prochaine skill selon où tu en es, ou explique ce qu'une skill vient de faire. |

## Choisir son chemin selon la taille du travail

| Taille | Enchaînement |
|---|---|
| **Une session** (changement clair) | `bmad-spec` → `bmad-build` |
| **Un epic** (plusieurs sessions, un résultat) | `bmad-spec` → `bmad-preview-ticketing` → `bmad-build` par story → `bmad-retrospective` |
| **Un projet** (plusieurs epics, ~20+ sessions) | Documents de planification partagés → pour chaque epic : `bmad-spec` → `bmad-preview-ticketing` → builds → rétrospective |

## Toutes les skills

### Analyse (optionnelle)

| Skill | Ce qu'elle fait |
|---|---|
| `bmad-brainstorming` | Session d'idéation animée, quand l'idée n'est pas encore claire. |
| `bmad-forge-idea` | Met une idée à l'épreuve jusqu'à ce qu'elle tienne… ou meure à moindre coût. |
| `bmad-deep-recon` | Recherche documentée pour appuyer une décision. |
| `bmad-product-brief` | Capture la vision produit une fois le concept clair. |
| `bmad-prfaq` | Teste un concept en partant du communiqué de presse (méthode « working backwards »). |

### Planification

| Skill | Ce qu'elle fait |
|---|---|
| `bmad-prd` | Crée, met à jour ou valide un PRD (périmètre produit partagé). |
| `bmad-ux` | Documente l'apparence et le comportement du produit. |
| `bmad-spec` | Condense l'intention en un contrat court. **Tous les chemins passent par elle.** |

### Solution et découpage

| Skill | Ce qu'elle fait |
|---|---|
| `bmad-architecture` | Prend les décisions techniques qui gardent cohérentes des parties construites séparément. |
| `bmad-create-epics-and-stories` | Découpe les exigences en epics et stories. |
| `bmad-preview-ticketing` | Ordonne les stories d'un epic et les enregistre dans `tickets.toml`. |
| `bmad-sprint-planning` | Vérifie que les stories sont prêtes, puis suit leur statut. |

### Construction et bilan

| Skill | Ce qu'elle fait |
|---|---|
| `bmad-build` | Une session d'implémentation : pose des questions, propose un plan à valider, puis code. |
| `bmad-build-auto` | Même chose sans attendre de validation humaine. |
| `bmad-retrospective` | Juge le résultat combiné d'un epic par rapport à sa spec. |

## Conseils d'utilisation

- **Commencer par** `/bmad-build` avec ce que tu veux changer (ex. `/bmad-build write an implementation of mars rover kata`), puis `/bmad-help` pour comprendre ce qui s'est passé.
- **Ne préparer que la planification nécessaire** : la documentation officielle déconseille de générer tous les documents par principe.
- **Sauter BMAD pour les petits changements** : une faute de frappe, une retouche CSS ou un changement de moins de trois fichiers sans vraie décision de conception ne justifient pas le cycle.
- **Garder la spec courte** : environ 40 pages au maximum ; condenser les documents bruts avant de les donner à `bmad-spec`.
- **Le découpage n'est pas un contrat** : mettre à jour la spec et redécouper les stories quand l'implémentation révèle des contraintes.
- `bmad-build-auto` **seulement une fois les décisions stabilisées.**
- **Un commit par unité logique** : plus facile à relire, valider et annuler.
- **Mettre à jour** : `npx skills update` (ou la marketplace), puis relancer `bmad setup`.

## Sources (documentation officielle)

- README BMAD Method
- Choisir un chemin de planification
- Construire son premier changement
- BMAD Method + Claude Code – DEV Community