GUIDE 09

Transformer un bon échange en méthode réutilisable

Niveau 3 — repérer un échange Claude qui a produit de la valeur, et le transformer en actif de travail partageable, testable et maintenable.

Emplacement réservé à l'infographie de la collection (format portrait). Le schéma ci-dessus tient lieu de repère de mise en page.

Un bon chat n'est pas encore une méthode

Quelqu'un passe trente minutes à corriger Claude jusqu'à obtenir un excellent résultat. Trois semaines plus tard : la conversation est introuvable, les instructions décisives sont oubliées, un collègue recommence depuis zéro — et le processus dépend toujours de son inventeur.

La valeur n'a pas été perdue parce que Claude était mauvais, mais parce que **le savoir n'a pas été externalisé**.

> **Le test de transférabilité.** Réutilisable signifie : quelqu'un d'autre réussit sans avoir vécu la conversation d'origine. Tant que ce n'est pas vrai, il y a un souvenir, pas une méthode.

Six questions auxquelles une méthode doit répondre

1. **Quand l'utiliser ?**
   Le déclencheur, formulé de façon reconnaissable.
2. **Quelles entrées fournir ?**
   Les documents et informations sans lesquels rien ne marche.
3. **Quelles étapes comptent ?**
   Celles dont l'ordre change le résultat.
4. **Quel résultat produire ?**
   Format, longueur, structure attendue.
5. **Comment vérifier ?**
   Les critères qui disent « c'est bon » — voir le Guide 01.
6. **Et quand ça ne convient pas ?**
   Les cas où il faut s'arrêter et demander à un humain.

À RETENIR

Un bon prompt dans l'historique d'une personne est une réussite individuelle. Une méthode documentée, testée et partagée devient un capital collectif.

VERSION WEB PERMANENTE
consulting.amplify-belgium.com/ressources/transformer-un-echange-en-methode

GUIDE 09 · COLLECTION KNOWLEDGE CLAUDE
Vérifié le 28 août 2026 · 6 pages

Frontière avec le Guide 10 : ici, « comment industrialiser ce qui marche ? ». Là-bas, « est-ce que cela le mérite ? ».

GUIDE 09 · 01

Instructions, connaissances, Skill et procédure : quatre rôles

Ces quatre supports sont souvent utilisés l'un pour l'autre. Ils ne portent pas la même chose et ne survivent pas aux mêmes changements.

SUPPORT	CE QU'IL PORTE	EXEMPLES · POINT DE VIGILANCE
Instructions de Project	Les comportements stables propres à un périmètre : ton, structure de réponse, rôle, conventions, ordre de priorité. [6]	Elles s'appliquent à tous les chats du Project. Y placer une règle qui ne vaut que pour un dossier crée du bruit partout ailleurs.
Base de connaissances du Project	Ce qui *fait référence* : procédure, documentation, modèle, données de contexte, définitions internes. [5][6]	C'est la source, pas la méthode. Un document obsolète qui y reste continue de faire autorité.
Skill	La manière de réaliser une tâche spécifique et répétable : un dossier d'instructions, de scripts et de ressources que Claude charge dynamiquement. [1]	Claude examine les compétences disponibles, charge celles qu'il juge pertinentes et applique leurs instructions. [1]
Procédure écrite hors Claude	La compréhension humaine du processus : pourquoi il existe, ce qu'il produit, qui en répond.	Indispensable pour auditer, former, ou travailler quand la Skill n'est pas disponible. C'est le seul support qui survit à un changement d'outil.

La règle de rangement

Le Project garde le contexte. La Skill garde la méthode. La procédure garde la compréhension humaine.

Trois supports, trois durées de vie. Le contexte change avec le dossier, la méthode change avec la pratique, la compréhension doit survivre aux deux.

Le symptôme d'un mauvais rangement

Une instruction de Project qui décrit pas à pas comment produire un document : elle s'applique à toutes les conversations du périmètre, y compris celles qui n'ont rien à voir.

Ou une Skill qui contient les données du client : elle devient fausse dès le client suivant, alors que la méthode, elle, restait valable.

À RETENIR

Ranger une méthode au mauvais endroit ne la perd pas tout de suite. Cela la rend seulement impossible à corriger plus tard.

Références de cette page : [1] What are skills · [5][6] Projects · [4] Personalization features

GUIDE 09 · 02

Transformer une conversation en méthode

Extraire les invariants, pas recopier quarante messages. Cinq étapes, dont la plus importante est la troisième.

1. **Repérer ce qui mérite d'être conservé**
   Trois questions : avons-nous obtenu un gain réel ? la tâche revient-elle ? d'autres personnes pourraient-elles en bénéficier ? Trois « non » : on referme la conversation sans remords.

2. **Identifier ce qui a fait la différence**
   Quelle donnée était indispensable ? Quelles corrections ont été nécessaires ? Quel format final était attendu ? Quelles erreurs revenaient ?

3. **Supprimer l'accidentel**
   Ne pas transformer en règle une correction propre à un dossier, un nom de client, un chiffre temporaire ou une exception. C'est l'étape qui décide si la méthode sera réutilisable ou seulement rejouable.

4. **Écrire la méthode**
   Six rubriques : quand utiliser, entrées, procédure, sortie attendue, critères de réussite, exceptions et escalade.

5. **Faire tester par quelqu'un d'autre**
   Si la méthode ne fonctionne que pour son auteur, elle n'est pas encore transférable — elle est un mode d'emploi de sa propre mémoire.

STRUCTURE D'UNE FICHE MÉTHODE — À REMPLIR EN UNE PAGE

```
Quand utiliser      déclencheur clair et reconnaissable
Entrées             documents et informations nécessaires
Procédure           étapes courtes, dans l'ordre qui compte
Sortie attendue     format et niveau de qualité
Critères            comment savoir que le travail est bon
Exceptions          quand s'arrêter et demander à un humain
```

CE QU'IL NE FAUT PAS CONSERVER

« Ajoute toujours une phrase sur la disponibilité en novembre. » — vrai pour un dossier, faux partout ailleurs. Une méthode qui accumule ces règles devient rapidement inutilisable, et personne n'ose plus les retirer.

CE QU'IL FAUT CONSERVER

« Signaler explicitement toute donnée non établie par les documents fournis. » — vrai à chaque exécution, vérifiable, et utile même pour quelqu'un qui n'a pas vu le dossier d'origine.

À RETENIR

Capturer les règles qui ont causé le succès, pas l'historique qui les entourait.

Références de cette page : les critères de réussite sont traités dans le Guide 01 ; la boucle de contrôle dans le Guide 03.

GUIDE 09 · 03

Concevoir une Skill maintenable

Une bonne Skill résout une tâche répétable et possède des frontières. Anthropic décrit les meilleures comme celles qui traitent une tâche spécifique et répétable, donnent des instructions claires, incluent des exemples quand ils aident, définissent quand elles doivent être utilisées, et se concentrent sur un seul workflow plutôt que d'essayer de tout faire. [2]

ÉLÉMENT	CE QU'ON Y MET	CE QUI LE REND EFFICACE
Nom	Orienté tâche, pas orienté outil.	Il doit se reconnaître dans une liste, six mois plus tard.
Description	Ce que fait la compétence et quand l'utiliser.	C'est elle que Claude utilise pour décider d'invoquer la Skill : la documentation demande d'être précis sur les cas d'application, dans un format court. [2]
Instructions	L'ordre de travail et les règles.	Commencer simple, en Markdown, avant d'ajouter des scripts complexes. [2]
Ressources	Modèles, exemples, références.	Des exemples d'entrées et de sorties valent souvent mieux qu'un paragraphe de consignes. [2]
Scripts	Uniquement si nécessaire.	À traiter avec prudence, et à tester après chaque changement significatif. [2]

Cinq tests avant de partager

1. Une demande évidente, où la Skill doit se déclencher.
2. Une formulation différente de la même tâche.
3. Une demande voisine où elle ne *doit pas* s'appliquer.
4. Une entrée incomplète.
5. Un cas limite.

Le troisième test est le plus souvent oublié — et c'est celui qui révèle une description trop large.

Sécurité

Anthropic recommande de ne pas coder en dur d'informations sensibles telles que des clés d'API ou des mots de passe, d'examiner toute compétence téléchargée avant de l'activer, de rester prudent avec les scripts, et d'utiliser des connexions appropriées pour accéder aux services externes. [2]

Une Skill est du contenu exécutable dans un environnement de travail : elle se traite avec le même sérieux qu'un script partagé.

Ce qu'une Skill ne fait pas. Elle réduit la répétition et la variance. Elle ne remplace ni une source de vérité, ni un contrôle qualité : une méthode stable appliquée à une matière fausse produit une erreur stable.

À RETENIR

Une Skill trop générale se déclenche mal, se teste mal et se maintient mal. Le périmètre est la première décision de conception.

Références de cette page : [2] How to create custom skills (support.claude.com)

GUIDE 09 · 04

Une méthode sans propriétaire devient une règle périmée

La reproductibilité n'est pas un sujet d'outil. Dès qu'une méthode est partagée, elle devient un objet de gouvernance — avec les mêmes besoins qu'une procédure.

CE QUE DOIT AVOIR TOUTE MÉTHODE PARTAGÉE	POURQUOI
Un propriétaire	Quelqu'un décide d'une modification. Sans cela, personne ne corrige et personne n'ose retirer.
Une version	On doit pouvoir dire laquelle est en usage, et comparer avec la précédente.
Une date de vérification	Les fonctions de l'environnement évoluent ; une méthode juste il y a six mois peut ne plus l'être.
Un petit jeu de tests	Deux à dix exemples représentatifs suffisent souvent à détecter une régression pratique.
Un historique de changement	Savoir *pourquoi* une règle a été ajoutée évite de la retirer par erreur — ou de la garder pour rien.
Une règle de retrait	Quand une méthode n'est plus utile, elle doit pouvoir être désactivée. Sans procédure de sortie, les méthodes s'accumulent.

Le partage change l'échelle du risque

Sur Team et Enterprise, une organisation peut mettre des compétences à disposition de tous ses utilisateurs : Anthropic indique que la gestion des compétences à l'échelle de l'organisation est réservée aux plans Team et Enterprise, et que seuls les *propriétaires* peuvent ajouter ou retirer une compétence pour toute l'organisation. [3]

C'est exactement ce qui rend la gouvernance nécessaire : une mauvaise méthode diffusée à cent personnes propage une erreur cent fois plus vite qu'un mauvais prompt individuel.

Deux mécanismes à ne pas confondre

Provisionner — mettre une compétence à disposition de toute l'organisation. Réservé aux propriétaires. [3]

Partager — donner accès à des collègues ou à un groupe ; les destinataires la voient dans la section des éléments partagés avec eux. [3]

Le premier engage l'organisation, le second engage une équipe. Les deux méritent un propriétaire, mais pas le même niveau de contrôle.

Standardiser ne signifie pas figer. Une méthode utile est assez stable pour être réutilisée et assez explicite pour être corrigée. Si personne ne sait comment la modifier, elle n'est pas standardisée — elle est gelée, ce qui est un autre problème.

À RETENIR

Toute méthode partagée a besoin d'un propriétaire, de tests et d'une date. Sans cela, elle se dégrade sans que personne ne le voie.

Références de cette page : [3] Provision and manage skills for your organization

GUIDE 09 · 05

Sources, périmètre et mise à jour

Sources primaires consultées le 28 août 2026.

Trois idées à emporter

1. Capturer les règles qui causent le succès, pas l'historique complet.

2. Project, base de connaissances, Skill et procédure ont des rôles différents.

3. Toute méthode partagée a besoin d'un propriétaire, de tests et d'une date.

Ce que ce guide ne couvre pas

Il ne dit pas si un cas d'usage mérite d'être industrialisé — c'est le Guide 10. Il ne traite ni des données transmissibles (Guide 05) ni des permissions (Guide 07).

À revérifier : la disponibilité des Skills et du provisioning dans le tenant Satelliteam, les règles de partage internes, les prérequis techniques d'exécution, et l'évolution du format des compétences.

Sources primaires

[1] **What are skills?**
Claude Help Center — définition et chargement dynamique
support.claude.com/en/articles/12512176-what-are-skills

[2] **How to create custom skills**
Claude Help Center — bonnes pratiques et sécurité
support.claude.com/en/articles/12512198-how-to-create-custom-skills

[3] **Provision and manage skills for your organization**
Claude Help Center — Team et Enterprise, rôle propriétaire
support.claude.com/en/articles/13119606-provision-and-manage-skills-for-your-organization

[4] **Understanding Claude's personalization features**
Claude Help Center — portée des instructions
support.claude.com/en/articles/10185728-understanding-claude-s-personalization-features

[5] **What are projects?**
Claude Help Center
support.claude.com/en/articles/9517075-what-are-projects

[6] **How can I create and manage projects?**
Claude Help Center — instructions et base de connaissances
support.claude.com/en/articles/9519177-how-can-i-create-and-manage-projects

[7] **Building Effective AI Agents**
Anthropic Engineering — principe de sobriété
anthropic.com/engineering/building-effective-agents