Prof. Aviv Zohar

Semester A, 2026–7

Writing a Skill

Write a procedure, use it, and improve it

spec-discussion/SKILL.md

```
---
name: spec-discussion
description: Help clarify…
---
# Discuss the feature
1. Read the request…
2. Ask about decisions…
3. Draft SPEC.md…
4. Review, then stop.
```

You explain the same procedure again and again

you Monday

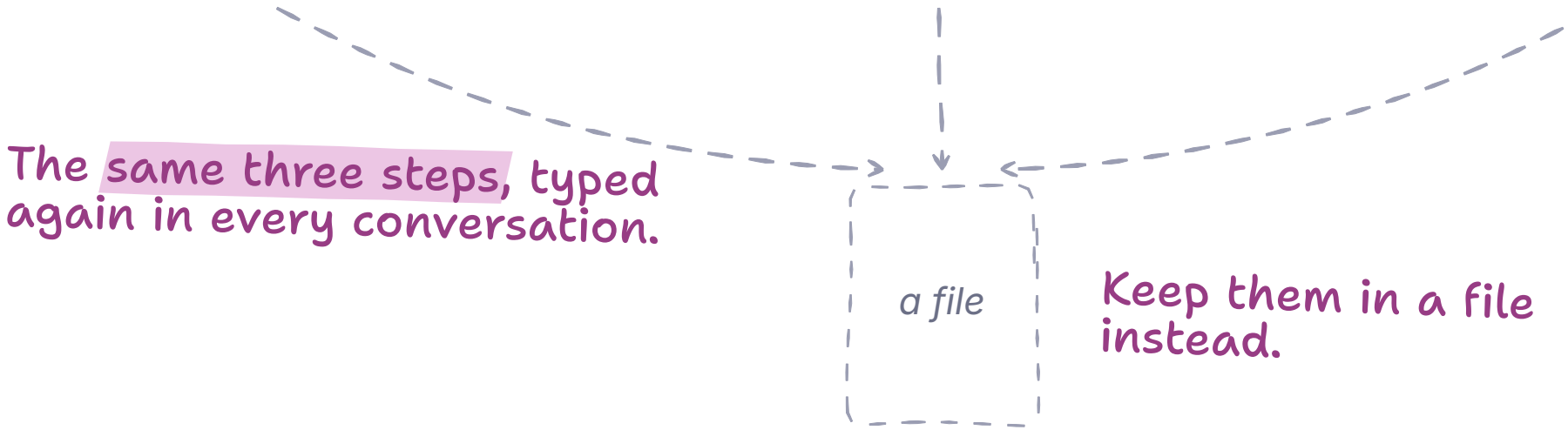
Before you build **tags**:
read the project first,
raise open decisions,
write down our agreement.

you Wednesday

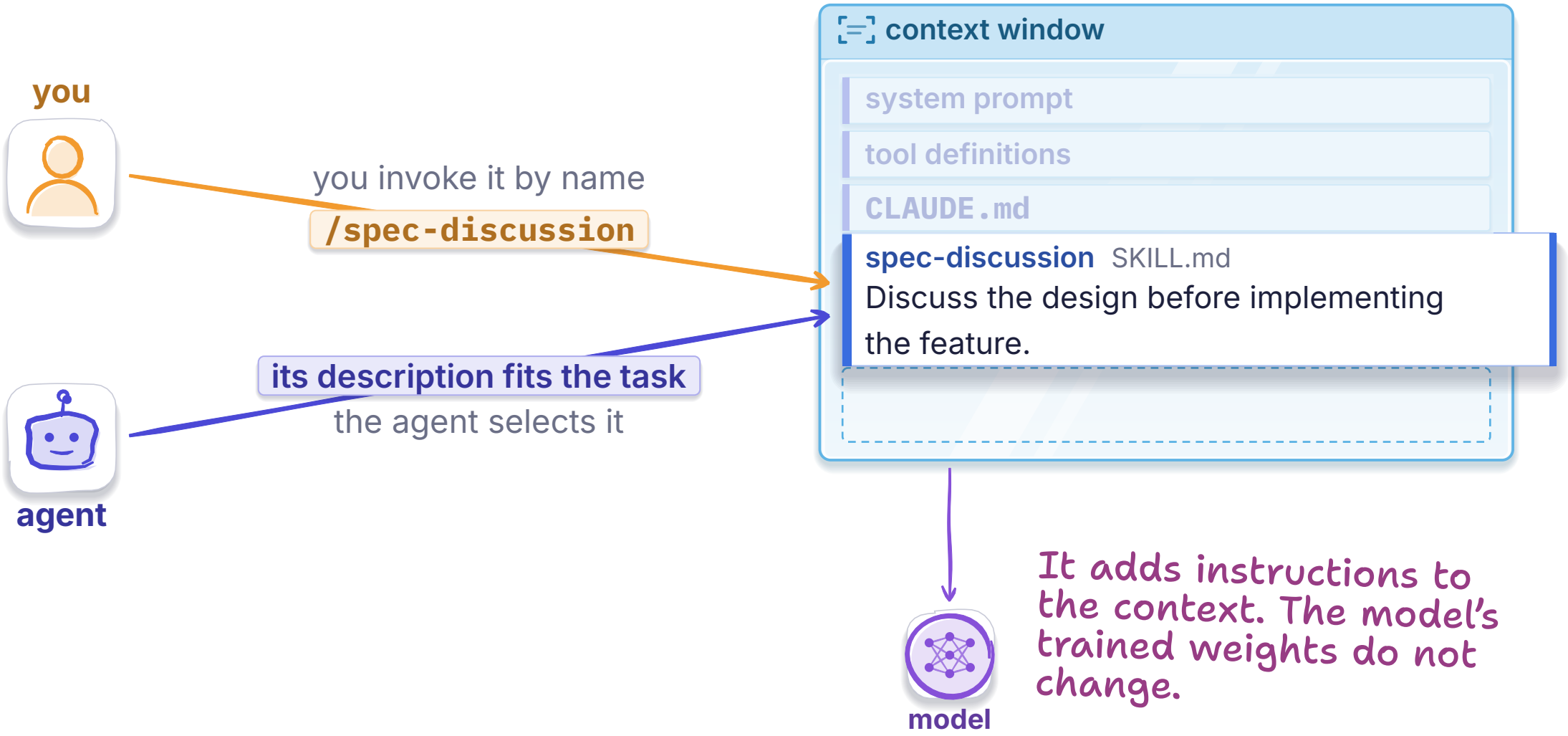
Before you build **export**:
read the project first,
raise open decisions,
write down our agreement.

you Friday

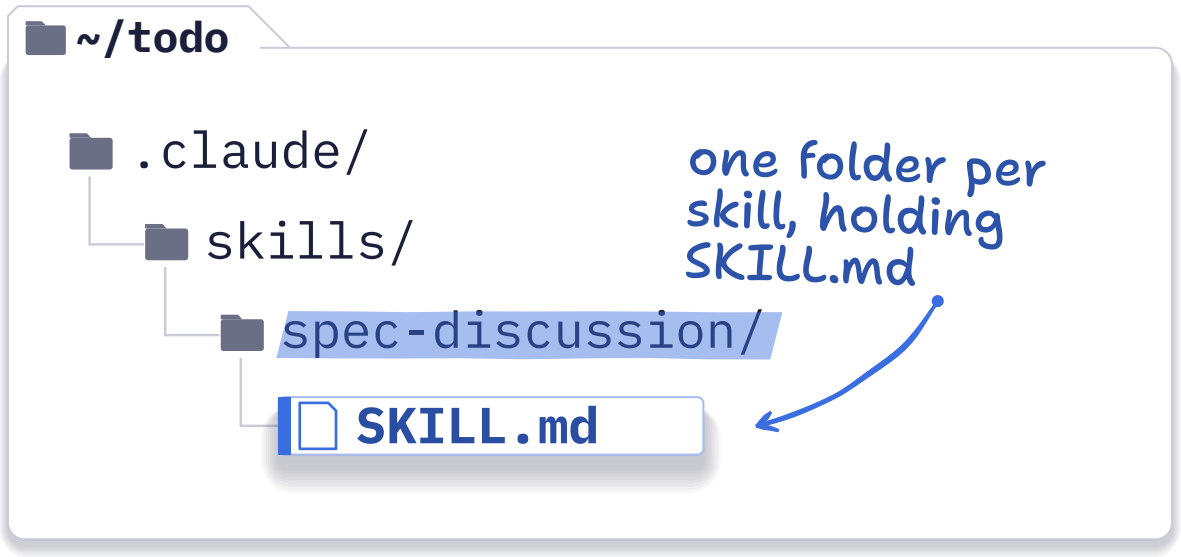
Before you build **search**:
read the project first,
raise open decisions,
write down our agreement.



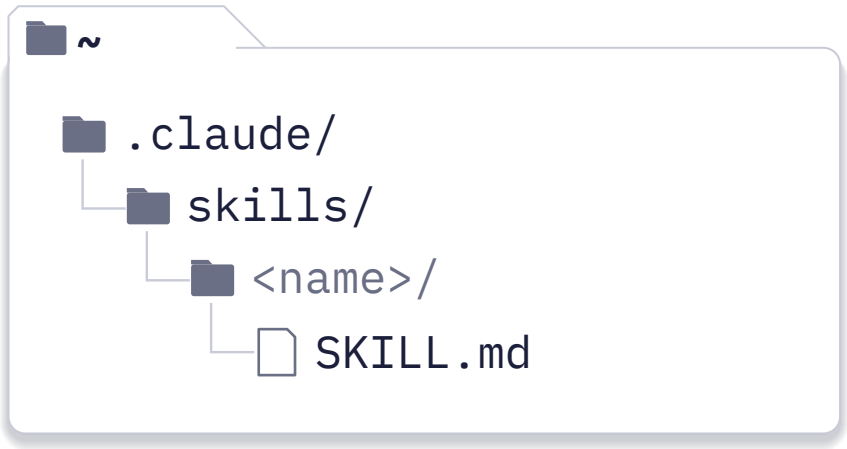
A skill keeps the procedure, ready to load



A project skill is a folder under .claude/skills



Project skill
Commit it, and everyone working on the repository has it.



Personal skill
In your home folder: available in all your local projects.

Plugins can package skills too.

SKILL.md: front matter, then the procedure

.claude/skills/spec-discussion/SKILL.md

```
1  ---
2  name: spec-discussion
3  description: >-
4    Help clarify a feature and draft its specification.
5    Use when the user requests a design discussion or a
6    feature has unresolved decisions about behavior,
7    scope, or integration. Skip routine changes whose
8    requirements are already clear.
9  ---
10
11 # Discuss the feature
12
13 Discuss the design before implementing the feature.
```

on the very first line

its name matches the folder

front matter
YAML: a name and a description

body
the procedure, in Markdown

Agent Skills standard: the same folder and file in other tools; install paths and extra settings differ.

The description says when the skill helps

.claude/skills/spec-discussion/SKILL.md

```yaml
---
name: spec-discussion
description: >-
  Help clarify a feature and draft its specification.
  Use when the user requests a design discussion or a
  feature has unresolved decisions about behavior,
  scope, or integration. Skip routine changes whose
  requirements are already clear.
---

# Discuss the feature

Discuss the design before implementing the feature.
```

ⓐ what it does
ⓑ the situation where it helps

Load it for ✅ a feature with important unresolved design decisions

not for ❌ "no spec exists yet": that would interview you before every small change

The description says when the skill helps

.claude/skills/spec-discussion/SKILL.md

```
---
name: spec-discussion
description: >-
  Help clarify a feature and draft its specification.
  Use when the user requests a design discussion or a
  feature has unresolved decisions about behavior,
  scope, or integration. Skip routine changes whose
  requirements are already clear.
---

# Discuss the feature

Discuss the design before implementing the feature.
```

includes a design discussion you ask for, and open questions of behavior, scope or integration

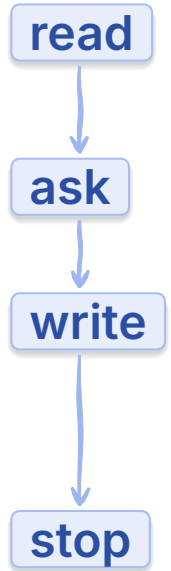
excludes routine changes with clear requirements

An existing spec can still need discussion when the feature changes.

The body is the procedure, step by step

.claude/skills/spec-discussion/SKILL.md

```
11 # Discuss the feature
13 Discuss the design before implementing the feature.
15 1. Read the request, relevant project instructions,
16    existing specs, and relevant code. Identify settled
17    decisions and important open questions.
18 2. Ask about unresolved decisions that affect this
   ⋮
21    careful discussion. Do not repeat settled questions.
22 3. Draft or update SPEC.md at the project root during
   ⋮
28 4. Summarize what will be built next and ask the user
   ⋮
31    Finish when there is shared understanding.
32    Wait for the user's instruction to implement.
```



A rule you care about goes in as a rule:
the agent follows the text literally.

Invoke the skill by name

shell

```
ⓐ
~/todo $ claude

❯ /spec-discussion Build me a small command-line todo app in Python.
```

Type the slash, its name, then your request. Check that its instructions load.

A direct way to try the procedure, without waiting for automatic selection.

Not found? Check:

ⓐ ⓑ ⓒ

`~/todo/.claude/skills/spec-discussion/SKILL.md`

ⓐ where you started the session

ⓑ the folder's name

ⓒ the file, SKILL.md

Use it, and review the procedure as you go

agent session `~/todo`

you typed by you

`/spec-discussion` Build me a small command-line todo app in Python.

agent written by the model

Storage: in memory, a JSON file, or SQLite?

you typed by you

A JSON file, in the project folder.

agent written by the model

With a JSON file: keep done todos, or delete them?

tool call asked by the model

```
Edit SPEC.md
+ Storage: todos.json in the project folder
```

ⓐ Useful questions?

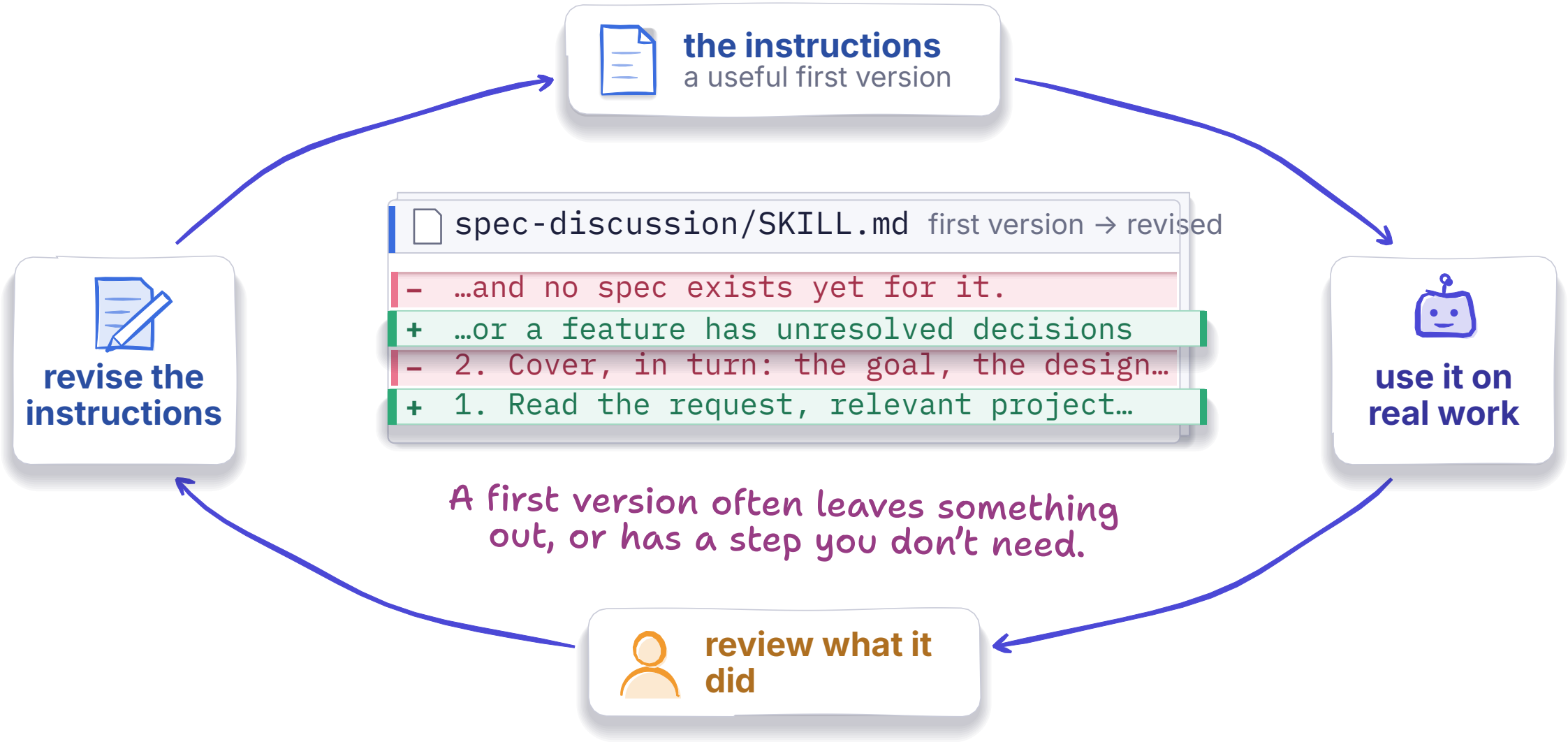
ⓑ Uses your answers?

ⓒ Faithful record?

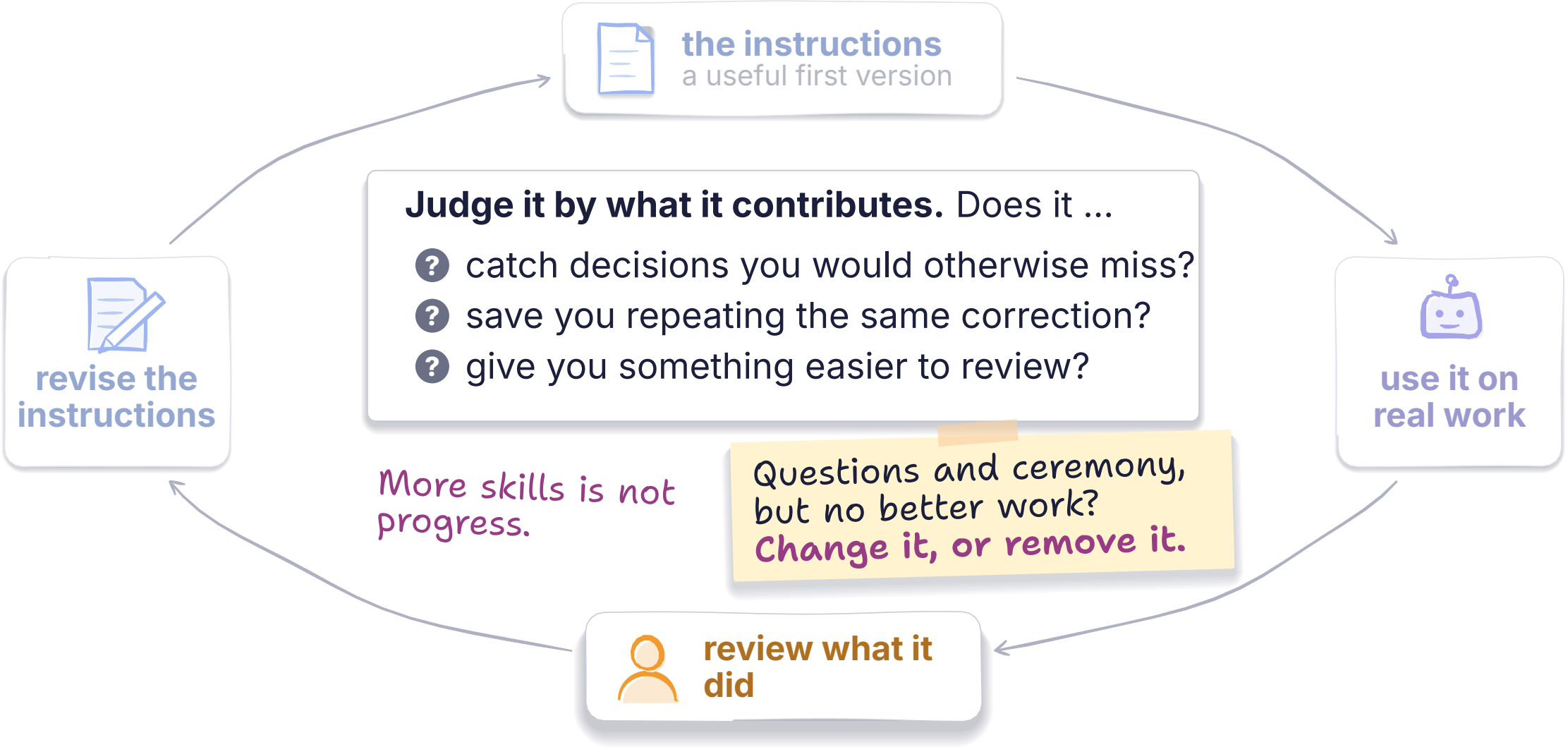
More telling than its name.

continued for illustration

Improve the skill by using it



Improve the skill by using it



Compare the same request with and without it

you the same request, in two fresh sessions

Build me a small command-line todo app in Python.

Without the skill

agent session `~/todo`

agent

I'll set up the project with uv ...

tool call

Bash `uv --version`

tool result

```
This command requires approval
```

With the skill invoked by name

agent session `~/todo`

tool call

Bash `ls -la … && find …`

agent

A fresh, empty project ... no settled decisions yet. First question: **Storage:** how should todos persist? ...

Compare the same request with and without it

you the same request, in two fresh sessions

Build me a small command-line todo app in Python.

Without the skill

agent session `~/todo`

agent

I'll set up the project with uv ...

tool call

Bash `uv --version`

tool result

```
This command requires approval
```

With the skill invoked by name

agent session `~/todo`

tool call

Bash `ls -la … && find …`

agent

A fresh, empty project ... no settled decisions yet. First question: **Storage:** how should todos persist? ...

As it goes on: how much do you explain and correct, next to similar tasks without it?

✅ **useful questions resolved: worth the time**

❌ **repeated or irrelevant questions: revise it**

captured 2026-09-16 · Claude Code 2.1.273 · sonnet · trimmed

Review the spec it writes, too

SPEC.md

```
# Todo app
## Intended behavior                    ⓐ
Add, list and complete todos from the shell.
## Scope
One user, one machine; no sync.
## Integration                          ⓑ
A uv project; tests run with pytest.
## Decisions and reasons                ⓒ
- Storage: todos.json, easy to read.
## Checks
## Open decisions
- Keep finished todos, or delete them?
```

ⓐ Does it capture the intended behavior?

ⓑ Does it fit the existing project?

ⓒ Does it keep the reasons for the important decisions?

headings from step 3 of the skill · content illustrative

Check that the agent picks it at the right time

you typed by you

Build me a small command-line todo app in Python.

→ should select `spec-discussion`

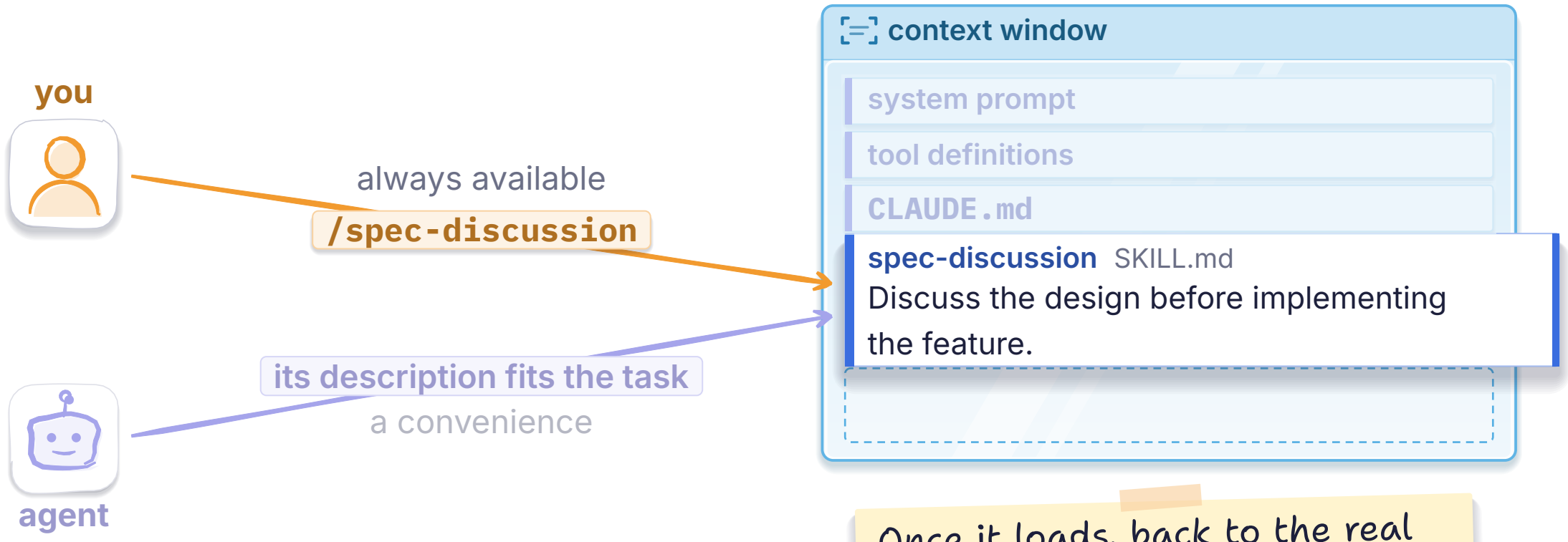
you typed by you

What does this Python error mean: `ModuleNotFoundError`?

→ should stay out of the way

Selected too often, or missed when it should help? **Revise the description.**

Invoking it by name always works



Checking selection is separate from judging the work.

Two settings close one way in

SKILL.md

```
---
name: spec-discussion
description: Help clarify…
disable-model-invocation: true
---
```

- ✓ /spec-discussion you, by name
- ✗ description fits the agent, on its own

You start the procedure yourself.

SKILL.md

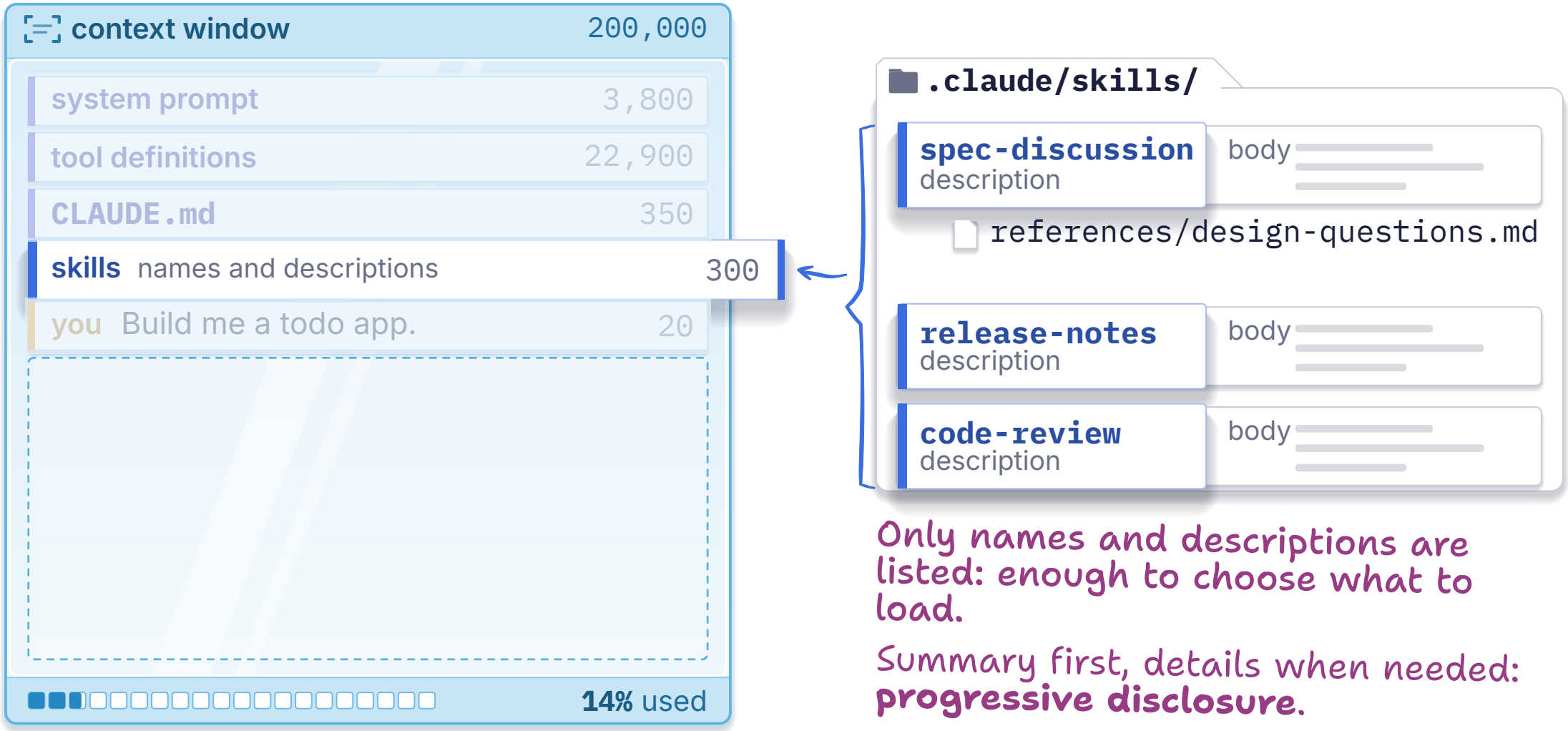
```
---
name: spec-discussion
description: Help clarify…
user-invocable: false
---
```

- ✗ /spec-discussion you, by name
- ✓ description fits the agent, on its own

No slash command; the model can still load it.

Hiding a command is not an access-control boundary.

Skills enter the context in stages

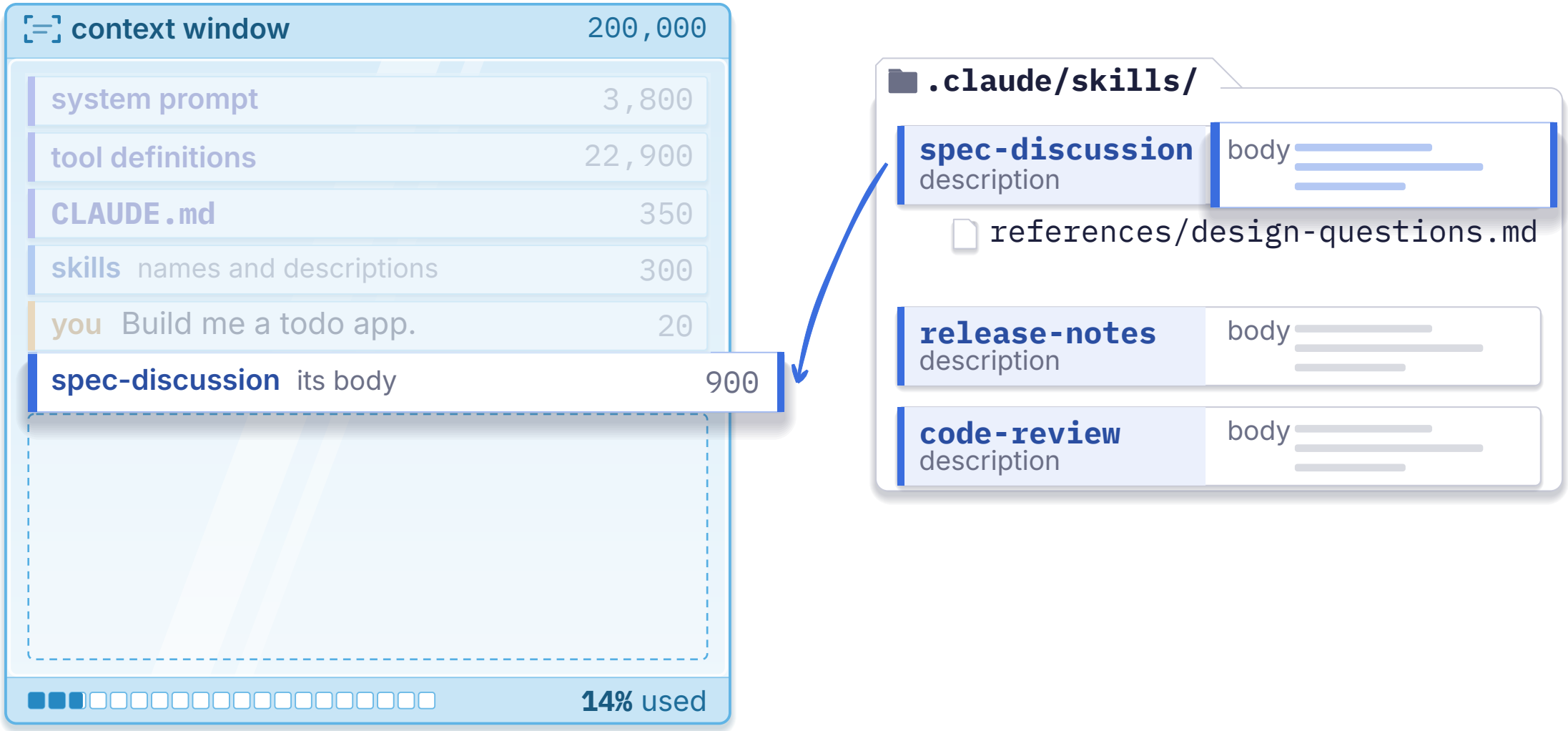


Only names and descriptions are listed: enough to choose what to load.

Summary first, details when needed: **progressive disclosure.**

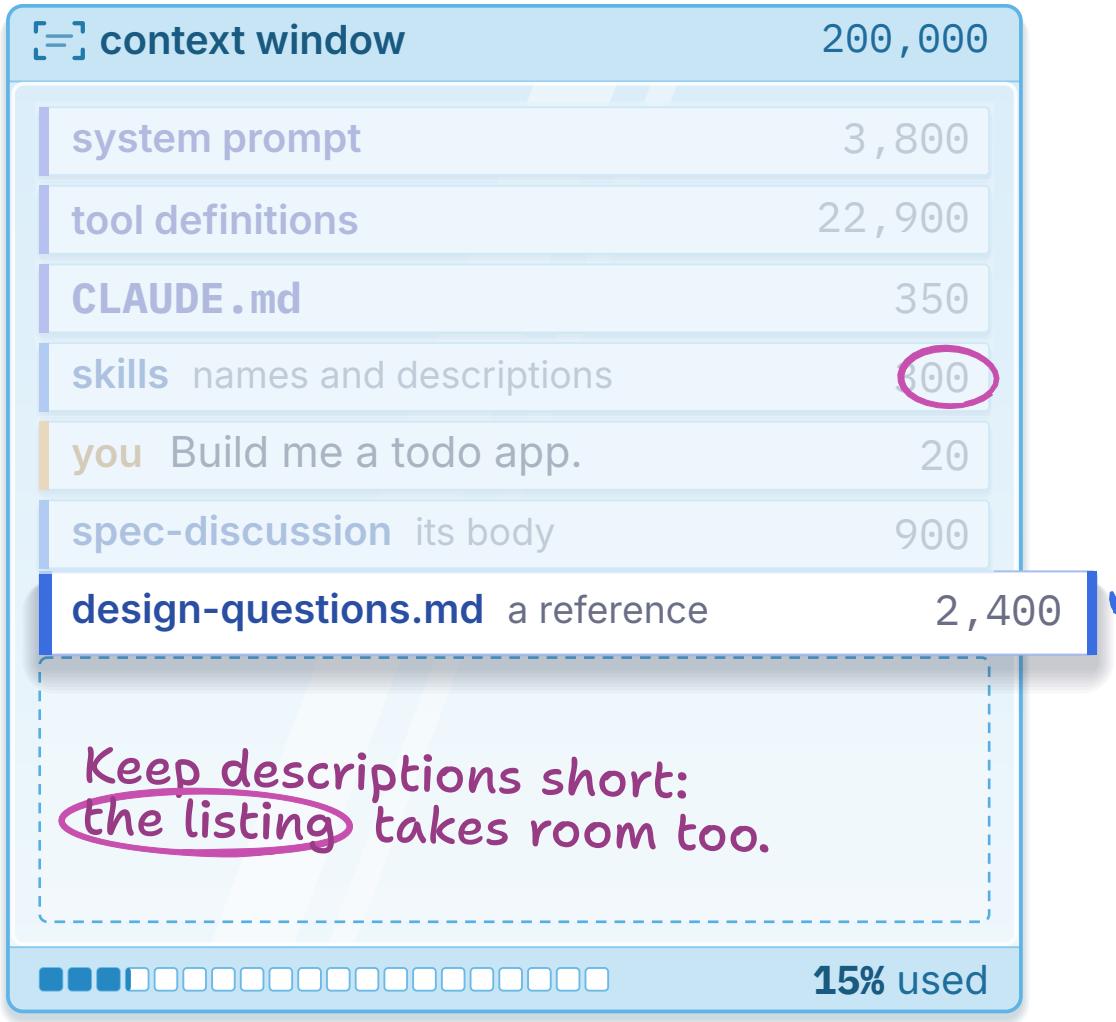
illustrative values

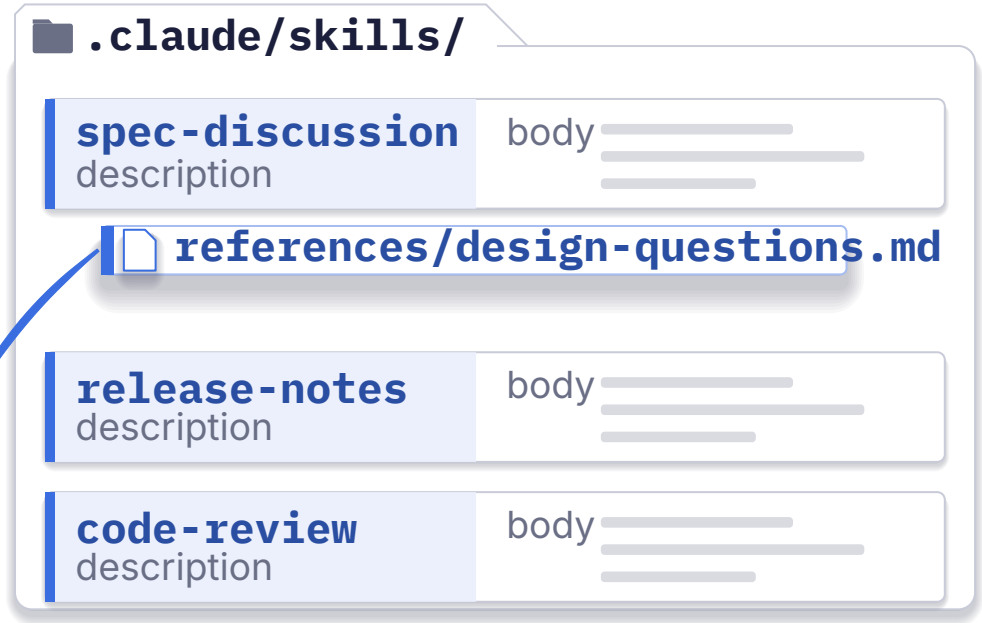
Skills enter the context in stages



illustrative values

Skills enter the context in stages



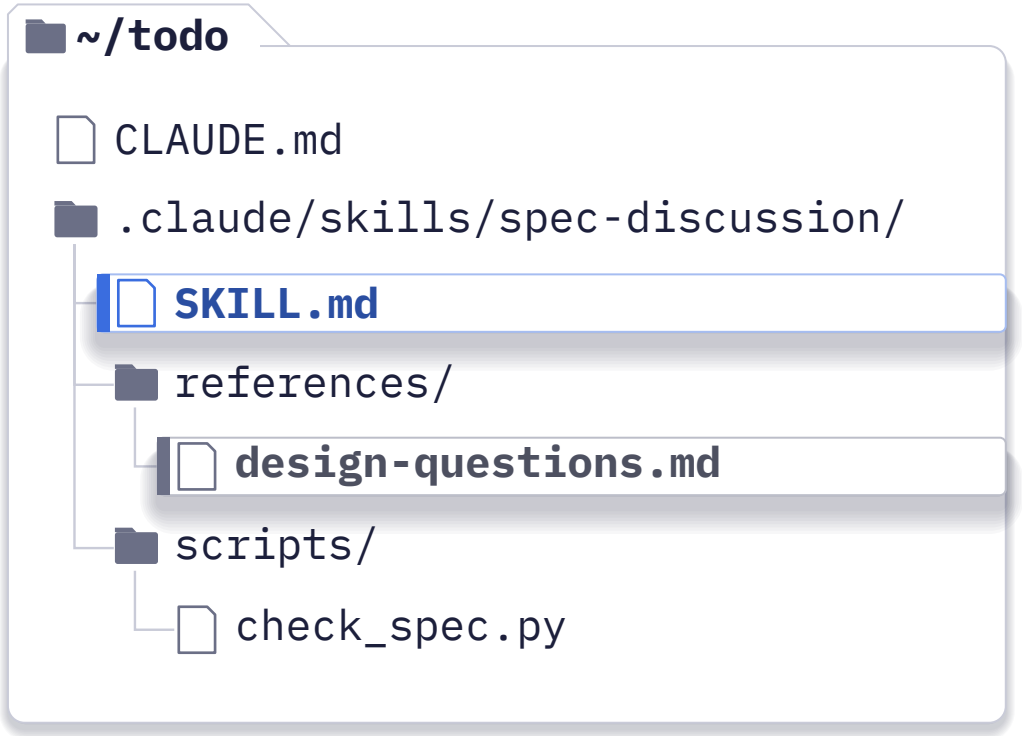


A linked reference is read when the procedure calls for it.

More room for the task than one always-loaded file.

illustrative values

Give each part of a skill its own job



SKILL.md

```
2. Ask about unresolved decisions…
   For a hard trade-off, first read
   references/design-questions.md.
3. Draft or update SPEC.md…
```

Details go in a linked file, **with when to read it.**

Short, but complete.

illustrative layout

Give each part of a skill its own job

~/todo

- CLAUDE.md
- .claude/skills/spec-discussion/
  - SKILL.md
  - references/
    - design-questions.md
  - scripts/
    - **check_spec.py**

SKILL.md

```
4. Before you finish, run:
   python3 scripts/check_spec.py
```

tool call asked by the model

Bash `python3 scripts/check_spec.py`

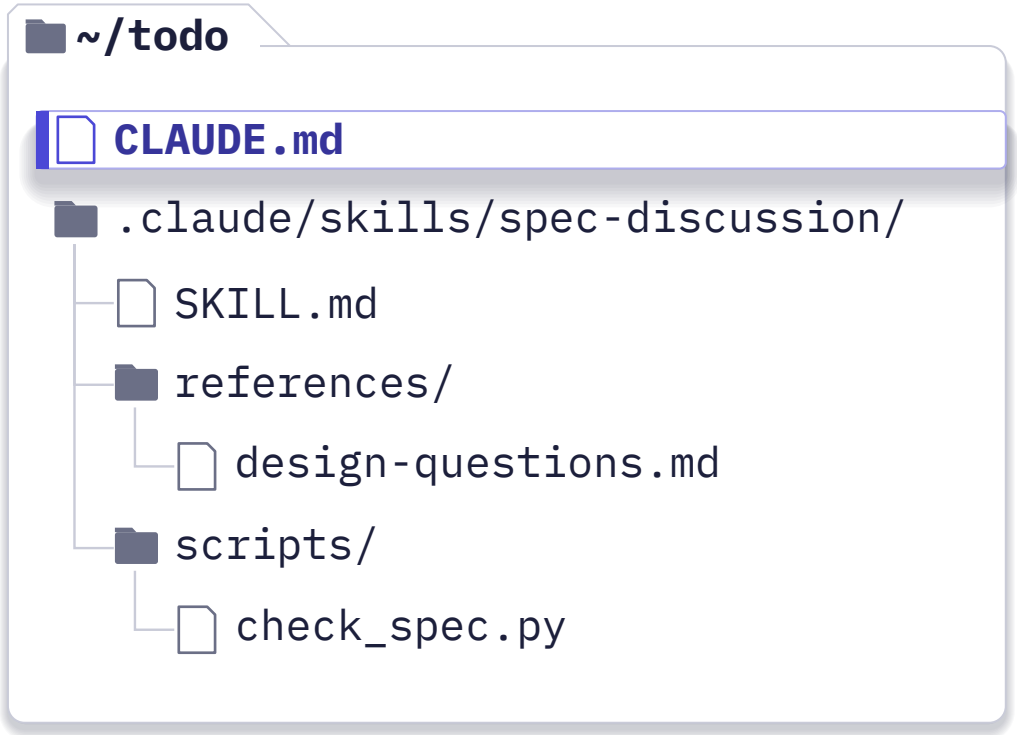
tool result run by the harness

```
SPEC.md: all sections present
```

The skill explains the procedure; **a tool runs the script**, so the model need not rebuild the commands each time.

illustrative layout

Give each part of a skill its own job



CLAUDE.md
rules that apply across the work

SKILL.md
the procedure for this task

Tools
carry out its operations

These work together.

illustrative layout

The course's own build-decks skill

.claude/skills/build-decks/

- SKILL.md
- reference/ ⓑ
  - lesson-review.md
  - teaching-review.md
  - slidesonnet.md
  - narration-production.md
- templates/
  - slidesonnet.toml

build-decks/SKILL.md

```
 2 name: build-decks
11 ## Pedagogy
26 ## Beautiful slides
77 ## Mechanics
93 ## Building a deck's slides
102 4. Build the plain PDF (`just compi
```

ⓐ slides and narration, together

ⓑ detailed references it points to

ⓒ how to build and check the result

The main procedure and its supporting material have different jobs.

A skill keeps a procedure you repeat



One file

The instructions you kept retyping, kept in SKILL.md.



Two ways in

By name, or when its description fits the task.



Learn from use

Each use shows what to fix in the instructions.

Keep asking: does the skill make the work better?

Choose one procedure you repeat, write it down, and improve it as you work.