

/sp-brainstorming

Source: `~/ .claude/skills/sp-brainstorming/SKILL.md`

name: brainstorming description: "You MUST use this before any creative work - creating features, building components, adding functionality, or modifying behavior. Explores user intent, requirements and design before implementation."

Brainstorming Ideas Into Designs

Overview

Help turn ideas into fully formed designs and specs through natural collaborative dialogue.

Start by understanding the current project context, then ask questions one at a time to refine the idea. Once you understand what you're building, present the design in small sections (200-300 words), checking after each section whether it looks right so far.

The Process

Understanding the idea:

- Check out the current project state first (files, docs, recent commits)
- Ask questions one at a time to refine the idea
- Prefer multiple choice questions when possible, but open-ended is fine too
- Only one question per message - if a topic needs more exploration, break it into multiple questions
- Focus on understanding: purpose, constraints, success criteria

Exploring approaches:

- Propose 2-3 different approaches with trade-offs
- Present options conversationally with your recommendation and reasoning
- Lead with your recommended option and explain why

Presenting the design:

- Once you believe you understand what you're building, present the design
- Break it into sections of 200-300 words
- Ask after each section whether it looks right so far
- Cover: architecture, components, data flow, error handling, testing
- Be ready to go back and clarify if something doesn't make sense

After the Design

Documentation:

- Write the validated design to `docs/plans/YYYY-MM-DD-<topic>-design.md`
- Use elements-of-style:writing-clearly-and-concisely skill if available
- Commit the design document to git

Implementation (if continuing):

- Ask: "Ready to set up for implementation?"
- Use superpowers:using-git-worktrees to create isolated workspace
- Use superpowers:writing-plans to create detailed implementation plan

Key Principles

- **One question at a time** - Don't overwhelm with multiple questions
- **Multiple choice preferred** - Easier to answer than open-ended when possible

- **YAGNI ruthlessly** - Remove unnecessary features from all designs
 - **Explore alternatives** - Always propose 2-3 approaches before settling
 - **Incremental validation** - Present design in sections, validate each
 - **Be flexible** - Go back and clarify when something doesn't make sense
-

Revision #6

Created 2026-02-18 08:39:54 UTC by John

Updated 2026-07-26 20:00:57 UTC by John