

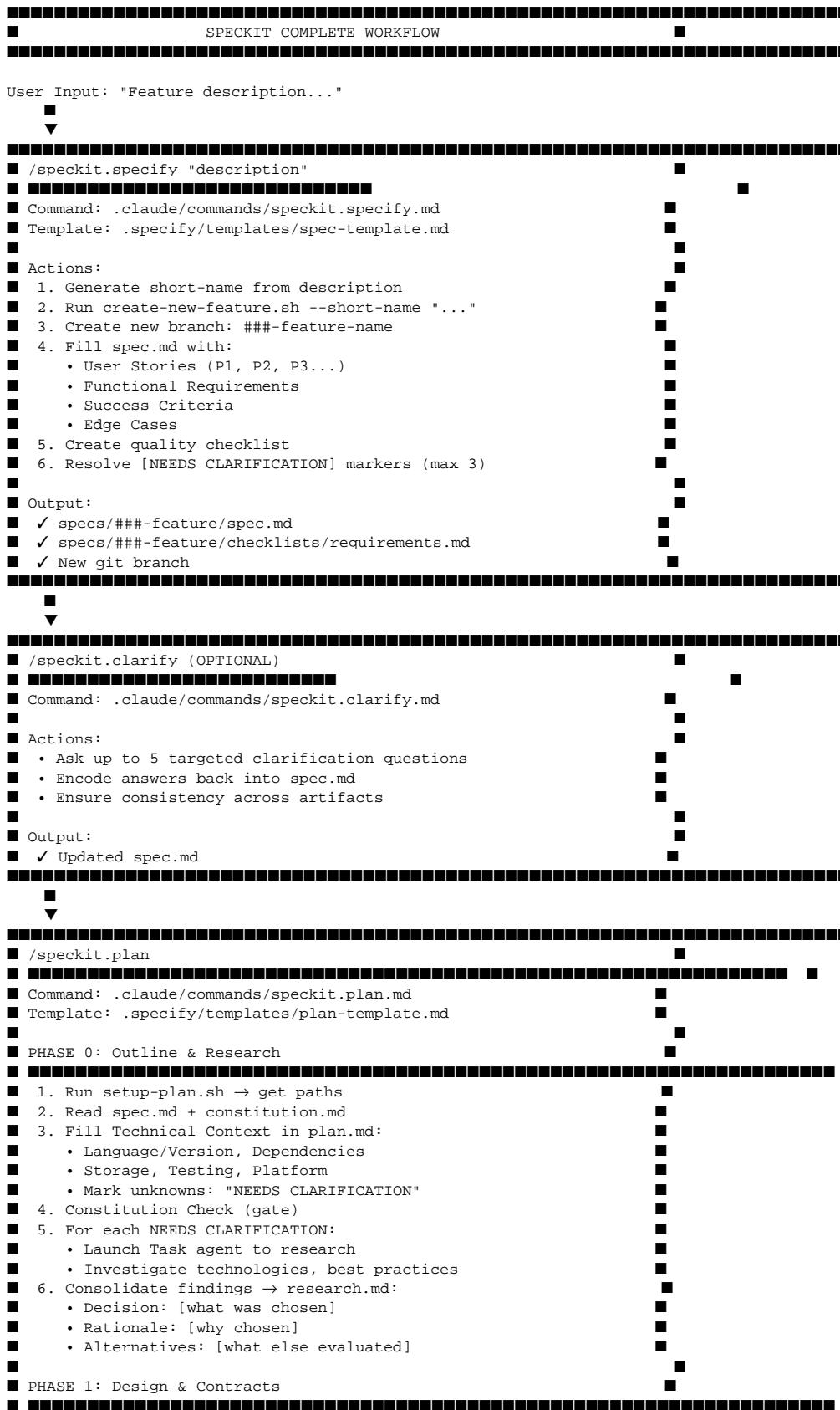
SpecKit Workflow Guide

AI Coding Framework - Complete Process Documentation

Generated: 2025-10-26

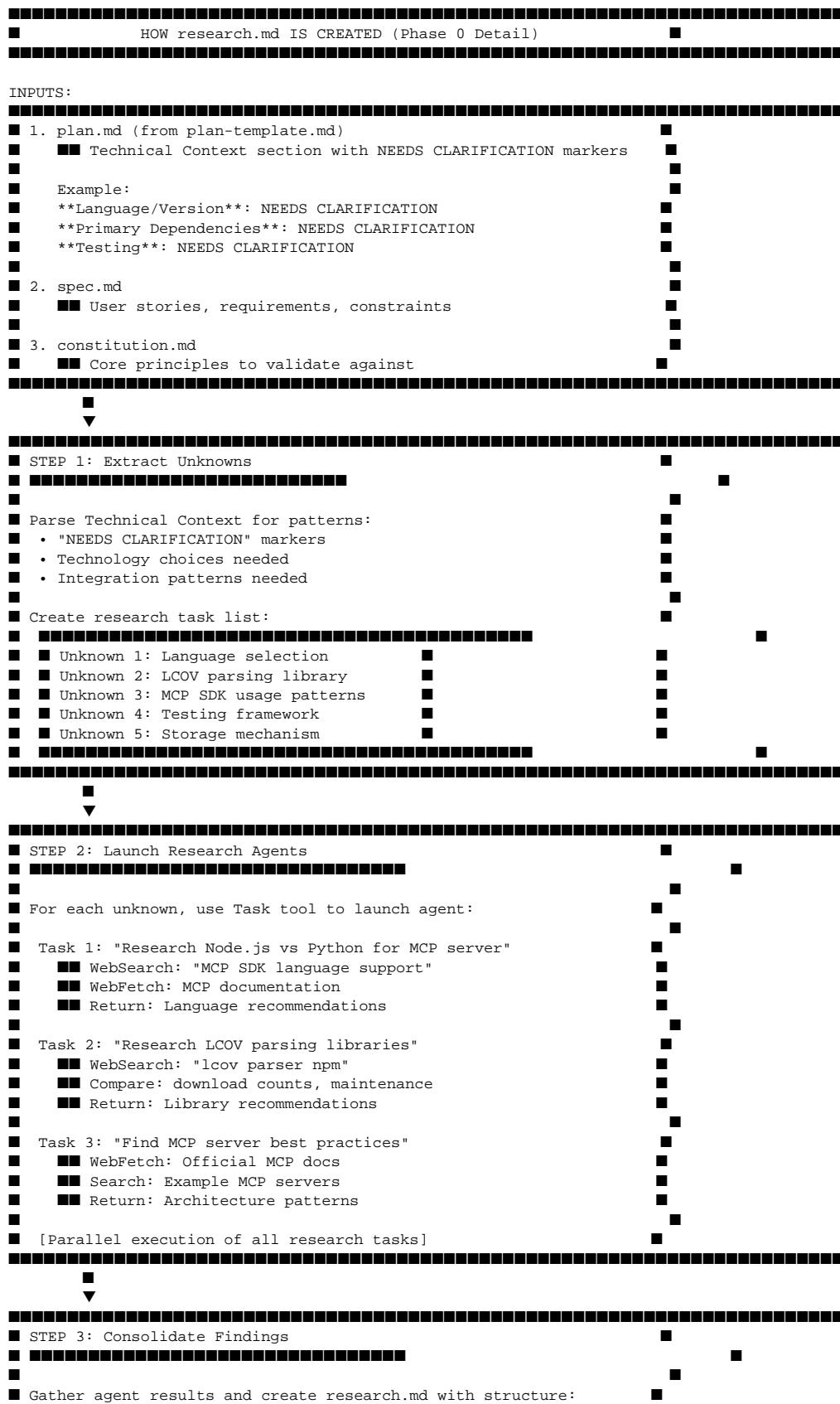
Version: 1.0

1. Complete Feature Development Flow



```
■ Prerequisites: research.md complete  
■  
■ 1. Extract entities from spec → data-model.md  
■ 2. Generate API contracts → /contracts/  
■ 3. Create quickstart.md  
■ 4. Update agent context → CLAUDE.md  
■ 5. Re-check Constitution compliance  
■  
■ Output:  
■ ✓ specs/###-feature/plan.md  
■ ✓ specs/###-feature/research.md ← CRITICAL FILE  
■ ✓ specs/###-feature/data-model.md  
■ ✓ specs/###-feature/quickstart.md  
■ ✓ specs/###-feature/contracts/*  
■ ✓ CLAUDE.md (updated)  
  
■ /speckit.tasks  
■ Command: .claude/commands/speckit.tasks.md  
■ Template: .specify/templates/tasks-template.md  
■  
■ Inputs:  
■ • spec.md (user stories with priorities)  
■ • plan.md (technical structure)  
■ • data-model.md (entities)  
■ • contracts/ (API endpoints)  
■ • research.md (technical decisions)  
■  
■ Actions:  
■ 1. Generate tasks organized by user story  
■ 2. Phase 1: Setup tasks  
■ 3. Phase 2: Foundational (blocking all stories)  
■ 4. Phase 3+: User Story tasks (P1, P2, P3...)  
■ 5. Final Phase: Polish & cross-cutting  
■  
■ Output:  
■ ✓ specs/###-feature/tasks.md  
  
■ /speckit.implement  
■ Command: .claude/commands/speckit.implement.md  
■  
■ Actions:  
■ • Execute tasks.md sequentially  
■ • Write code, tests, configs  
■ • Run tests after each phase  
■ • Commit incrementally  
■  
■ Output:  
■ ✓ Implemented feature in src/  
■ ✓ Tests in tests/  
■ ✓ Git commits
```

2. research.md Generation Deep Dive



```
■ # Research: [Feature Name]
■
■ ## 1. Language Selection
■ **Decision**: TypeScript with Node.js
■ **Rationale**: @modelcontextprotocol/sdk is Node-native
■ **Alternatives Considered**:
■   • Python: Less mature MCP support
■   • Go: No official MCP SDK
■
■ ## 2. LCOV Parsing Library
■ **Decision**: @friedemannsommer/lcov-parser
■ **Rationale**: 600k+ weekly downloads, actively maintained
■ **Alternatives Considered**:
■   • lcov-parse: Abandoned, last update 2018
■   • Custom parser: Violates constitution
■
■ [... more research sections ...]
■
■ NO TEMPLATE FILE - Format defined in speckit.plan.md lines 47-50
```

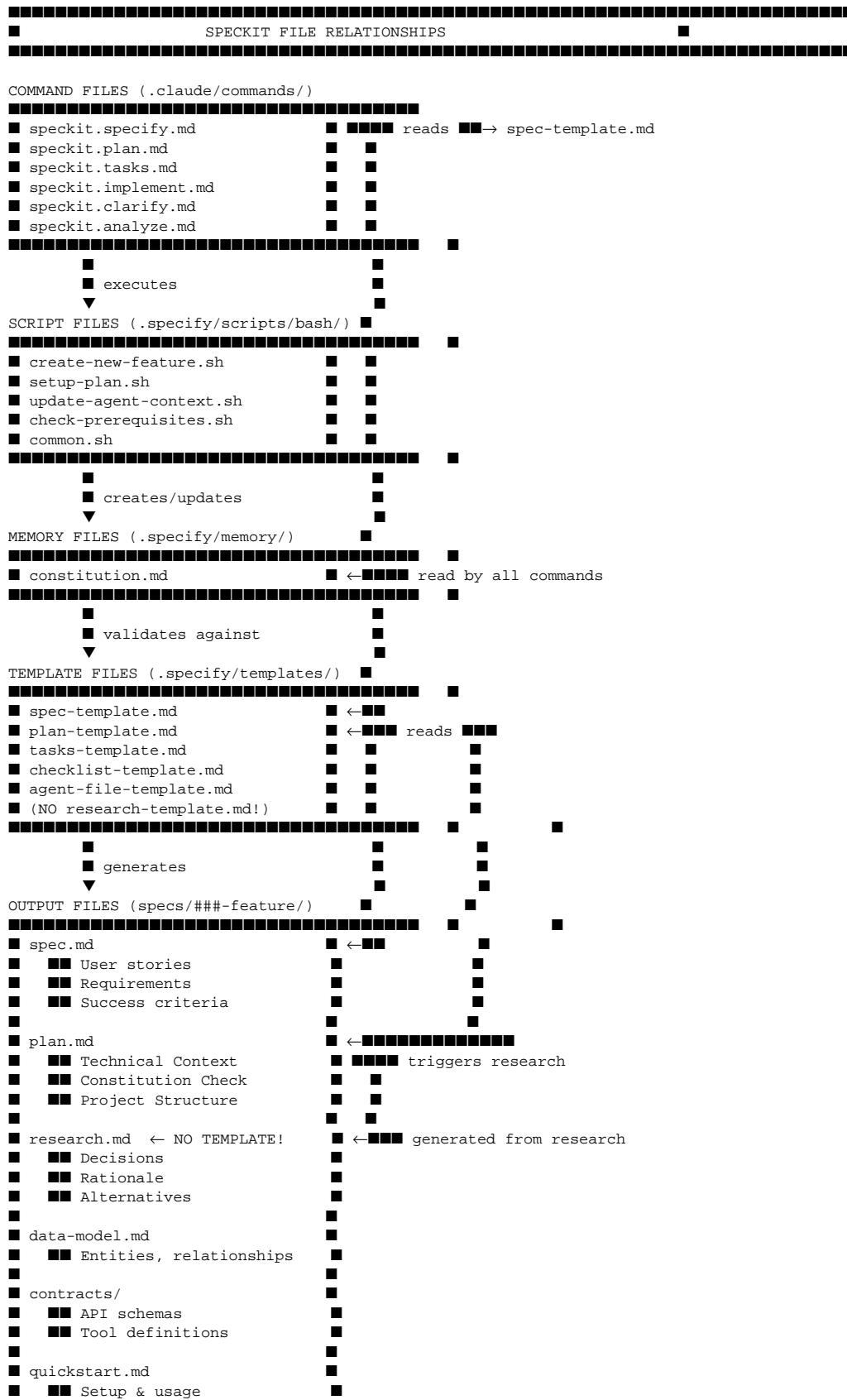
↓

```
■ STEP 4: Update plan.md
■
■ Replace NEEDS CLARIFICATION markers with concrete choices:
■
■ BEFORE:
■   **Language/Version**: NEEDS CLARIFICATION
■
■ AFTER:
■   **Language/Version**: TypeScript with Node.js (latest LTS)
■
■ (Derived from research.md Section 1)
```

↓

```
■ OUTPUT FILES:
■   ✓ specs/###-feature/research.md ← Generated (no template)
■   ✓ specs/###-feature/plan.md      ← Updated with concrete values
```

3. File Relationships & Dependencies



```
■ tasks.md
■   ■ Implementation tasks
■
■ checklists/
■   ■ requirements.md
■
■ implements
▼
SOURCE CODE (src/, tests/)
■ Actual implementation
```

4. Data Flow: From User Input to research.md

User Types:

- I need an MCP tool to analyze LCOV coverage files...

/speckit.specify
Extracts: User stories, functional requirements
Creates: spec.md

spec.md contains:

- What features are needed (not HOW)
- Success criteria
- Edge cases

/speckit.plan - Phase 0 Start

1. Load plan-template.md

- Has Technical Context section with fields

2. AI attempts to fill fields from spec.md:

- Can I determine from "analyze LCOV files" that:
 - This needs to parse files → some language needed
 - "MCP tool" → needs MCP SDK
 - Which language? → NEEDS CLARIFICATION
 - Which LCOV parser library? → NEEDS CLARIFICATION
 - How to store recordings? → NEEDS CLARIFICATION

3. Fills plan.md Technical Context:

- **Language/Version**: NEEDS CLARIFICATION
- **Primary Dependencies**: @modelcontextprotocol/sdk + NEEDS...
- **Storage**: NEEDS CLARIFICATION

/speckit.plan - Phase 0 Research

4. Extract all "NEEDS CLARIFICATION" markers

- Create research questions

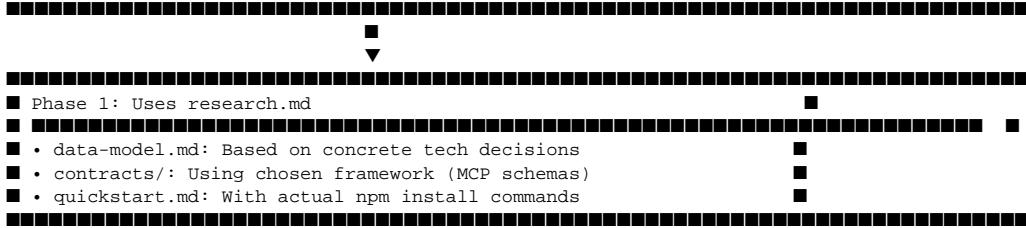
5. For each question, launch Task agent:

- Agent 1: Research "MCP SDK language support"
 - WebSearch + WebFetch
 - Returns: "Use TypeScript + Node.js"
- Agent 2: Research "LCOV parser Node.js npm"
 - Compare libraries
 - Returns: "@friedemannsommer/lcov-parser"
- Agent 3: Research "recording storage patterns"
 - Evaluates against constitution
 - Returns: "Temporary filesystem storage"

6. Consolidate agent results into research.md:
Format source: .claude/commands/speckit.plan.md lines 47-50
(NO template file exists!)

7. Write research.md to specs/###-feature/research.md

8. Update plan.md, replacing NEEDS CLARIFICATION



5. Key Insights

KEY FINDINGS:

1. NO research.md TEMPLATE EXISTS
The research.md file is dynamically generated, not from a template.
Format is defined in: .claude/commands/speckit.plan.md (lines 47-50)
2. RESEARCH IS TRIGGERED BY UNKNOWNs
"NEEDS CLARIFICATION" markers in plan.md's Technical Context section trigger the research phase.
3. TASK AGENTS DO THE RESEARCH
Multiple Task agents run in parallel to research:
 - Technology choices
 - Library comparisons
 - Best practices
 - Integration patterns
4. CONSTITUTION VALIDATES CHOICES
All research results must align with project principles defined in .specify/memory/constitution.md
5. RESEARCH INFORMS ALL DOWNSTREAM ARTIFACTS
research.md results populate:
 - plan.md (Technical Context)
 - data-model.md (entity design)
 - contracts/ (API schema choices)
 - quickstart.md (setup instructions)

FILE LOCATIONS REFERENCE:

Commands:

.claude/commands/speckit.specify.md
.claude/commands/speckit.plan.md
.claude/commands/speckit.tasks.md
.claude/commands/speckit.implement.md

Templates:

.specify/templates/spec-template.md
.specify/templates/plan-template.md
.specify/templates/tasks-template.md
(NO research-template.md)

Scripts:

.specify/scripts/bash/create-new-feature.sh
.specify/scripts/bash/setup-plan.sh
.specify/scripts/bash/update-agent-context.sh

Memory:

.specify/memory/constitution.md

Outputs (per feature):

specs/###-feature/spec.md
specs/###-feature/plan.md
specs/###-feature/research.md ← Dynamically generated
specs/###-feature/data-model.md
specs/###-feature/contracts/
specs/###-feature/quickstart.md
specs/###-feature/tasks.md
specs/###-feature/checklists/

RESEARCH.MD FORMAT:

For each research topic:
Decision: [what was chosen]
Rationale: [why chosen]
Alternatives considered: [what else was evaluated]