



Appendix B: Chapter Map

Every chapter, its domain, and the single most important thing Josh learns in it. Read this before a second pass through the novel or use it to locate the chapter most relevant to a problem you're working through.

Ch.	Title	Domain	What Josh learns
P	The All-Hands	<i>Context</i>	Project Meridian is announced. Eleven people are eliminated. Josh learns who was lost before he understands what was lost.
1	Inheritance	<i>Agentic Systems</i>	Josh inherits a broken agent. Error rate 12.4%. He recognizes Tomas's code by the comment style. First attempt: more instructions. Result: 11%.
2	Vera	<i>Agentic Systems</i>	Customer harm made concrete. Josh discovers that better instructions produce marginal improvement - until the error rate spikes overnight.
3	The Whiteboard	<i>Agentic Systems</i>	Mary Cate draws the agentic loop. The model doesn't remember - it re-reads. The system prompt is not a persistent instruction. It is text the model encounters fresh on every turn.
4	Control Flow	<i>Agentic Systems</i>	The team implements the loop correctly. Derek: 'We're not telling it what to do. We're building a track it runs on.'
5	Tomas	<i>Agentic Systems</i>	Josh calls Tomas. The multi-concern decomposition pattern. Seven hours at \$165. The knowledge is still here - it's just expensive.
6	Caleb Wants a Demo	<i>Agentic Systems</i>	A \$2,400 refund processes because the threshold hook was in staging, not production. Josh sits in the parking garage where Mary Cate was six months ago.
7	Hooks	<i>Agentic Systems</i>	Mary Cate gets in the passenger seat. Deterministic vs. probabilistic compliance. Pre-call interception. PostToolUse filtering. The handoff summary that sets the human reviewer up to decide.

Appendix B: Chapter Map

Ch.	Title	Domain	What Josh learns
8	The Hub and the Spokes	<i>Agentic Systems</i>	Priya's coordinator breaks in four different ways. Priya: 'What if the coordinator is supposed to be dumb and the subagents are supposed to be smart?'
9	Parallel	<i>Agentic Systems</i>	The dumb coordinator works. Four subagents, parallel execution, synthesis. Josh writes the first line of the architecture document.
10	Priya's Garden	<i>Tool Design</i>	Eighteen tools, six MCP servers, everything connected. The agent calls the wrong tool on every query. Priya: 'I don't understand why.'
11	Read the Description	<i>Tool Design</i>	Josh makes Priya read the descriptions aloud. They're the same. Four hours of rewriting changes nothing in the code and everything the model can see.
12	Eighteen Is Too Many	<i>Tool Design</i>	Mary Cate counts the tools. She counts everything. The model's selection problem and every future human's maintenance problem are the same problem.
13	What Errors Are For	<i>Tool Design</i>	Derek brings a spreadsheet: 34 failures, all labeled 'Operation failed.' The agent has been guessing at what failures mean. Four error types. One boolean.
14	Derek's Taxonomy	<i>Tool Design</i>	Derek implements structured error responses. He finds an account that has been returning false transient errors for 18 months. Six customer-facing failures. Nobody knew.
15	Project vs. Personal	<i>Tool Design</i>	An Operations team member modifies .mcp.json and breaks production for 17 hours. Nobody told them it was shared infrastructure. Tomas committed the last clean version 14 months ago.
16	tool_choice	<i>Tool Design</i>	Identity verification before account actions: 95% with a prompt instruction. 100% with forced tool selection. Guaranteed behavior requires code.
17	Six Weeks	<i>Claude Code</i>	14 features, 3 developers, 6 weeks. Two pipeline items that need Tomas. Josh schedules the call and moves it twice.
18	Everyone Is Using It Differently	<i>Claude Code</i>	Priya generates components that don't match conventions. Derek's tests are consistent with themselves but not with

Appendix B: Chapter Map

Ch.	Title	Domain	What Josh learns
			the codebase. Josh writes a 400-word Slack message. Nobody changes anything.
19	Mary Cate's Forty Lines	<i>Claude Code</i>	Mary Cate sends her CLAUDE.md without explanation. 40 lines. Two hours to write, three months of editing. Josh asks what she cut.
20	Caleb Checks In	<i>Claude Code</i>	Caleb: 'So you've got it all figured out.' Josh: 'We've got a better starting point.' Caleb: 'Same thing.' It is not the same thing. Tomas passes the conference room without looking in.
21	Rules for the Path	<i>Claude Code</i>	Every rule loading for every file dilutes every rule. Path-specific rules with glob patterns. Josh finds Tomas's README in .claude/ - the full architecture, described 14 months ago, waiting.
22	The /review Command	<i>Claude Code</i>	Derek proposes a shared slash command. He almost puts it in the wrong place. The .gitkeep in .claude/commands/ has a comment: 'per-user in ~/.claude/commands, per-project here.'
23	Plan Before You Build	<i>Claude Code</i>	Priya starts the microservices restructuring in direct execution mode. 23 files, 3 circular dependencies, a shared config object nobody had mapped. She reverts.
24	Priya Recovers	<i>Claude Code</i>	Plan mode is slower because it asks questions. The questions find the utility function in notification-helpers.ts that would have broken the admin interface silently.
25	The Pipeline	<i>Claude Code</i>	CI integration with -p flag. Existing tests as context for generation. Prior review findings as context for incremental review. Q3 ships: 11 of 14. Vera: 'First sprint since February where I haven't had to apologize to a key account.'
26	The Call from Vera	<i>Structured Output</i>	First Meridian Bank. Wrong capital ratios in a regulatory filing. The compliance contact is Bea Okafor. Josh is very still for a moment.
27	What Instructions Can't Do	<i>Structured Output</i>	Mary Cate: 'You described the output you wanted. Did you show it what the output looks like?' Then Josh calls Bea. Two pages.

Appendix B: Chapter Map

Ch.	Title	Domain	What Josh learns
			Entirely professional. Neither of them mentions Project Meridian.
28	Schema	<i>Structured Output</i>	Required fields produce fabricated values. Null is not empty - null is a finding. The other enum category. Extraction provenance. The schema change reveals correct extractions that had been labeled fabrications.
29	The Retry Loop	<i>Structured Output</i>	Priya builds the retry loop from the actual failure taxonomy. You can't retry your way to information that isn't there. Re-validate the full output after any successful correction.
30	False Positives	<i>Structured Output</i>	Margot's team dismisses 40% of flags without investigating. A false alarm, after enough repetitions, becomes indistinguishable from no alarm. Precision of criteria determines whether a flag produces review or habit.
31	Overnight	<i>Structured Output</i>	Batch vs. synchronous by latency tolerance. Josh almost deploys batch mode to a blocking workflow. The \$500 cost savings would have missed a regulatory deadline.
32	Two Instances	<i>Structured Output</i>	Derek: the model that produced the output retains its reasoning. It won't catch its own errors as effectively as an instance reviewing a stranger's work. 'Don't ask the builder to inspect the building.'
33	What It's Worth	<i>Structured Output</i>	The first meeting without a crisis. Mary Cate shows Josh what she's been building. The citizen developer framework. Josh finally asks what it's for. Mary Cate: 'The case for what comes after this.'
34	Everything, Slowly	<i>Context Management</i>	Three systems degrade simultaneously - not failing, drifting. Josh writes in his notes: 'The system is lying in a way that looks like the truth.'
35	Caleb Notices	<i>Context Management</i>	Caleb: 'It's like the system has a memory problem.' Josh: 'That's actually exactly what it is.' Caleb accepts not knowing. It is the first time in the novel he has done this.
36	What Accumulates	<i>Context Management</i>	Two days in the logs. Verbose tool outputs. Numerical summarization loss. Absent

Appendix B: Chapter Map

Ch.	Title	Domain	What Josh learns
			case facts. 'We've been adding context without subtracting anything.' 'Subtraction requires knowing what matters.'
37	Scratchpads and Summaries	<i>Context Management</i>	Priya solves the citation-accuracy problem on a Friday without telling Josh. Structured claim-source mappings. Conflict annotation over resolution. Tomas flagged this problem in a TODO comment 14 months ago.
38	When to Ask a Human	<i>Context Management</i>	Aggregate accuracy masks segment performance. Stratified sampling. Routing thresholds calibrated to actual error rates. Mary Cate shows Josh the full citizen developer proposal.
39	Escalation	<i>Context Management</i>	A customer asks three times to speak with a human. The agent keeps helping. Caleb watches Josh fix it in real time. Something shifts in his face.
40	The Document	<i>Context Management</i>	Josh finishes the architecture document. Sends it to Tomas. Tomas: 'This is right. You should send it.' The board approves the governance proposal. Yemi arrives.
Ep.	The Architecture Document		Three months later. Yemi reads the document on her first day. Bea emails Josh two sentences. He reads it twice. He doesn't reply that day.