

Blue Sheen

Claude user guide

For the Acme Mid-Market Engineering and Customer Success teams

May 23, 2026

Prepared for Acme Mid-Market Co.

Sample for bluesheen.com — Acme Mid-Market Co. (fictional)

Why Claude

Acme’s stack is Python / Django, React, AWS, Snowflake. The team already uses ChatGPT Team (60 seats) and GitHub Copilot Enterprise (45 dev seats). This guide adds Claude — specifically Claude Code for engineers and Claude Projects for Customer Success — to that mix.

Claude is not better than ChatGPT at everything. It is better at three things you do every day: long-context analysis (your full incident reports fit in one prompt), structured code reviews (Claude Code reads your repo and follows your CLAUDE.md), and customer-voice drafting that doesn’t sound like a robot wrote it.

This is not a generic Claude tutorial. It is the playbook for the work Acme actually does.

Account setup

Each team uses a different Claude entry point. The matrix:

Team	Plan	Entry point
Engineering (45 people)	Claude Team — to provision	Claude Code in VS Code + Cursor
Customer Success (28 people)	Claude Team — same workspace	Claude Projects with the “Incident Triage” project
Marketing (12 people)	Claude Pro individual until usage proves Team value	claude.ai web
Executive (4 people)	Claude Pro individual	claude.ai web + iOS app

Provisioning steps for the Engineering and Customer Success Team workspace are in Appendix A.

Your team’s approved use cases

Per the policy your AI Governance Committee approved on 2026-05-22, Claude is approved for:

- Engineering: code completion, code review, incident postmortem drafts, ADR drafts, debugging assistance on non-PHI codepaths

- Customer Success: customer-facing email drafts (for human review), incident triage research, knowledge base updates
- Marketing: blog drafts, landing page copy, email campaign drafts
- Executive: board memo drafts, all-hands prep, strategic brief reviews

Claude is NOT approved for: anything touching PHI, anything that ships to customers without human review, anything that affects compensation or performance decisions.

Ten prompt patterns for your team

1. The “structured incident postmortem” pattern (Engineering)

Paste the Slack incident channel transcript. Prompt:

“ Read this incident transcript. Produce a postmortem in the format we use at Acme: Summary, Timeline (HH:MM in UTC), Root cause, Contributing factors, What went well, What went poorly, Action items (with owner + by-when). Match the tone of the example postmortem I'll paste next.

Then paste one example postmortem. Claude calibrates voice and structure from it.

2. The “code review with our conventions” pattern (Engineering)

Open the file in VS Code with Claude Code installed. Comment:

“ /review — apply our CLAUDE.md conventions, focus on error handling and PHI safety. Flag anything that violates the patterns in /internal-docs/python-conventions.md .

Claude Code reads your CLAUDE.md and the conventions doc, then reviews accordingly.

3. The “customer email draft, our voice” pattern (Customer Success)

In Claude Projects, the “Customer Comms” project has Acme’s brand voice doc + the last 50 sent customer emails as project files. Prompt:

“ A customer wrote in saying [paste excerpt]. Draft a response in the voice the project documents establish. Don't apologize unless we did something wrong. Be specific about the next step.

Claude reads the project context and drafts in your established voice — not generic-CS-speak.

4. The “what changed in this PR and why it matters” pattern (Engineering)

In a PR review:

“ Read the diff. Explain what changed, why the change was needed, and what reviewers should look at first. Output in the format: ## Summary (2 sentences) / ## Files to review carefully / ## Open questions.

5. The “knowledge base article from a customer thread” pattern (Customer Success)

“ A customer hit this issue [paste the thread]. We resolved it via [paste solution]. Write a knowledge base article in our standard KB format: Problem, Why it happens, Resolution, Prevention. Audience: clinic admins, not engineers.

6. The “weekly leadership prep brief” pattern (Executive)

Paste the past week's Slack #leadership channel + the prior board memo. Prompt:

“ Brief me on what changed this week that the board would want to know. 5 bullets max. Flag anything that contradicts the strategic priorities I defined in [paste priorities].

7. The “design doc to ADR” pattern (Engineering)

Long design doc → tight Architecture Decision Record:

“ Read this design doc. Produce an ADR in the format we use at Acme: Context, Decision, Consequences, Alternatives considered. Maximum one page.

8. The “blog draft from an internal lesson” pattern (Marketing)

“ Engineering shipped [feature] last quarter. The internal retrospective is attached. Write a 1,200-word blog post for an audience of healthcare CTOs. No buzzwords. Don't oversell. Lead with the problem, not the feature.

9. The “customer support triage” pattern (Customer Success)

“ A customer reported [issue]. Their tier is Enterprise, they've been a customer for 18 months, they had a similar issue 3 months ago [paste prior incident]. Suggest a triage path: who should own this, what's the SLA, what 3 questions should I ask before diagnosing.

10. The “regulatory diff” pattern (Compliance + Engineering)

“ The HHS updated [regulation] in March. Here's the old version [paste]. Here's the new version [paste]. What changed that affects a HIPAA-covered SaaS like Acme? List the actions we should consider, ranked by urgency.

Iteration patterns

The first response is rarely the right response. Standard pattern:

1. First prompt: ask for the artifact
2. Read it. Note three things you'd change.
3. Second prompt: “Three changes — make these three changes only” (list them). Do NOT re-prompt the whole task.

This converges faster than re-prompting from scratch and avoids Claude drifting to a different solution.

Error and hallucination handling

When Claude is wrong, it is often confidently wrong. Triggers to verify by hand:

- A specific number, date, or quote you didn't provide
- A specific function name or library API that you can't immediately verify
- A specific HIPAA, SOC 2, or regulatory citation
- A specific customer name or contract term

If Claude provides any of these, verify before acting. Especially regulatory citations — Claude has been known to invent CFR section numbers that sound plausible.

Data handling do's and don'ts

Do	Don't
Paste internal docs, code, and Tier 1/2 content into the Team workspace	Paste PHI, customer PII, or contract data
Use Claude Projects for stable context (brand voice, prompt library)	Use Claude with personal accounts for Acme work
Save useful prompts back to the team prompt library	Auto-send Claude output to customers without review
Cite Claude when output is substantially AI-drafted and customer-facing	Use Claude for clinical decision support

Integration guide

Slack: Claude is not yet integrated into Slack. When that integration is approved, the team prompt library moves to a `/claude` slash command.

Notion: Use the Claude Notion integration to summarize long Notion docs. Same data-tier rules apply.

GitHub: Claude Code is the primary integration. Configure per Appendix B.

30-day adoption plan

Week	Engineering	Customer Success	Marketing	Executive
1	Install Claude Code; pair-program one task	Set up "Customer Comms" project	Draft one blog with Claude	Pair-prompt one board prep
2	Use pattern 1, 2, 4 on real PRs	Use pattern 3, 5 on real tickets	Draft email campaign	Use pattern 6 weekly
3	Add team prompt library to repo	Audit drafts vs sent — calibrate	A/B vs human-drafted	Use pattern 7 on one ADR
4	Measure: PR review time saved	Measure: ticket resolution time	Measure: blog draft time	Measure: prep time saved

Common pitfalls

- **Pasting PHI into the prompt** even when "just for context." Policy violation. Always sanitize first.
- **Trusting fabricated regulatory citations.** Always verify CFR / HIPAA / SOC 2 references.
- **Re-prompting from scratch** instead of iterating. Wastes time and drifts the solution.
- **Generic prompts** like "summarize this." Be specific: format, length, audience, tone.
- **Forgetting to cite** when the output is substantially AI-drafted and customer-facing.
- **Letting prompts rot.** The library that worked in Q1 may not match Q3 product. Quarterly audit.

FAQ

Will Claude replace ChatGPT? No. Both have a place. Use Claude for long-context analysis, code review, voice-correct drafting. Use ChatGPT for what your team already uses it for.

Can I use Claude with PHI? No, until we have a BAA with Anthropic AND the policy is updated.

What happens to the team prompt library if Claude changes pricing? Prompts are portable. We export the library quarterly. Migration to another model would be a 1-2 week project.

Where do I report a bug or confusing prompt? `#ai-tools` Slack channel, or open a GitHub issue in `acme/ai-tools-feedback`.

Glossary

- **Claude Code:** the CLI / IDE-integrated coding assistant version of Claude
- **Claude Projects:** shared workspace within Claude.ai for stable team context
- **Context window:** the maximum amount of text Claude can hold in one conversation
- **Hallucination:** a confidently stated but false output
- **Prompt library:** the team-maintained list of approved, calibrated prompts

Appendix A: Provisioning the Claude Team workspace

(omitted in sample — the real guide includes screenshot-level steps for IT)

Appendix B: Claude Code in VS Code + Cursor

(omitted in sample)

...

This sample guide was hand-crafted by Blue Sheen for a fictional client. Your real guide will be calibrated to your team's actual workflows, voice, and tool stack. Request your custom guide at bluesheen.com/tools/claude-user-guide-generator/.