

AI Powered Development

for Power Platform Professionals

How Power Platform professionals use Claude Code, GitHub Copilot, and the MCP ecosystem to build faster, govern better, and ship with confidence.

13

Chapters covering every asset type

50+

Production-ready prompts

3

Parts · Full-stack coverage

● **Part 1** The Foundation — Setup, tools, MCP servers

● **Part 2** Building With Claude Code — All 7 asset types

● **Part 3** Beyond the Stack — Connectors, governance, scale



Victor Dantas

Microsoft Power Platform MVP · Author

PART 1 – THE FOUNDATION

The biggest change isn't what AI can do — it's **who builds** and **how fast**.

Power Platform professionals now have a second AI superpower: not just building AI *into* apps, but using AI *to build* apps — faster, more consistently, and with higher quality.

✗ THE OLD WAY

AI as a Feature

- Add a Copilot Studio chat widget
- Call Azure OpenAI from a flow
- Embed AI into a Power Pages site
- AI is the product you ship

✓ THIS BOOK'S APPROACH

AI as a Team Member

- Claude Code generates data models
- AI writes flows, Canvas apps, Pages
- You review, govern, and deploy
- AI is the practice that ships faster

10x

Prototype time reduction for well-scoped Power Platform work

3 min

To scaffold a 5-table Dataverse data model from a description

\$0

Additional Power Platform licensing required for AI dev tools

THREE LEVELS OF AI INVOLVEMENT

Automation
If-this-then-that logic. You define every step. Deterministic, repeatable. AI writes it faster, but the ceiling is what you anticipated.



Orchestration
Multi-system coordination across Dataverse, Teams, Outlook, SharePoint. Still deterministic — but AI generates complex pipelines in seconds.



Agency
You define the goal. Claude Code reasons through the steps, takes action, observes results, adapts. It feels like a developer, not autocomplete.

WHY POWER PLATFORM PROS HAVE THE EDGE

01 Business Process Knowledge
You already know the domain, approval chains, edge cases, and table structures. Your prompts are precise. A dev learning your domain takes weeks just to gather requirements.

02 Full Ecosystem Familiarity
You know Dataverse, Canvas Apps, Power Automate, Power Pages, Copilot Studio. When AI generates something wrong for your DLP policy, you catch it.

03 Governance & ALM Literacy
Managed solutions, environment variables, connection references, deployment pipelines. AI without governance literacy produces demos that can't ship. You already have it.

• PART 2 — BUILDING WITH CLAUDE CODE

Seven chapters. Every asset type in the Power Platform stack.

From Dataverse schemas to Copilot Studio agents — generated, validated, version-controlled, and deployable.

DATVERSE**Data Model Generation**

- dv-metadata skill generates full schemas from descriptions
- Live MCP schema awareness — no invented column names
- PAC CLI round-trip: create, pack, deploy in one command
- ER diagrams auto-generated as Mermaid

POWER AUTOMATE**Flow Generation**

- Full JSON flows with connection references — never hardcoded
- Environment variables for every configurable value
- Error handling and approval timeouts built in by default
- pac solution export → edit → pack → import ALM workflow

CANVAS & MODEL APPS**App Scaffolding**

- pa.yaml per screen, App.pa.yaml, full naming conventions
- Empty-state messages, loading indicators, error handling
- Model-driven: TypeScript generative pages via /genpage
- PCF components scaffold as typed React + Fluent UI v9

CODE APPS**React + TypeScript SPAs**

- /create-code-app skill scaffolds full React + Vite app
- Typed Dataverse service layer with OData \$select, \$filter
- useWorkOrders hook pattern — consistent across team
- Fluent UI v9 component library, strict null checks

POWER PAGES**Portal & Code Sites**

- Liquid templates with FetchXML — schema-accurate
- Web API JS with CSRF token pattern auto-generated
- Code Sites: React Modern Apps with Dataverse OData service
- CSS variables for theming, partner extranet scenario

COPILLOT STUDIO**Agent Design & YAML**

- 4 plugin commands: manage, author, test, troubleshoot
- Production system prompts with escalation criteria
- Topic YAML: trigger phrases, actions, adaptive cards
- Simulate conversations before importing — catches gaps early

SKILLS LIBRARY**Write Your Own Skills**

- Skill = markdown file with YAML header — no framework
- 5 sections: Goal · Prerequisites · Steps · Standards · Checklist
- Enterprise skills shared via git across the whole team
- Skills compound: write once, every session inherits them

THE CORE LOOP**Author → Test → Troubleshoot → Push**

- pac agent push syncs YAML changes to Copilot Studio live
- Always deploy via solutions — Dev → UAT → Prod
- Every system prompt change tracked in git — diffable, reversible
- Skills encode governance constraints, not just style preferences

• PART 3 — BEYOND THE STACK

Two tools. One runtime. Zero rework.

Part 3 turns the direction around: from AI *building* Power Platform to Power Platform *calling* AI — and from one developer's setup to an enterprise team's standard.

4

Chapters covering the full enterprise AI adoption arc

2

AI tools working together — Claude Code + GitHub Copilot

40-60%

Prototype time reduction within a team's first month

GITHUB COPILOT

CLAUDE CONNECTOR

GOVERNANCE

SCALE

GITHUB COPILOT — COMPLEMENTARY, NOT COMPETING

GitHub Copilot — File Scope

- Inline formula completion inside .pa.yaml Canvas files
- OData filter expressions from comments in flow JSON
- Power Fx delegation warnings — explained and rewritten
- TypeScript hook & component completion matching existing patterns
- Copilot Chat: targeted rewrites of agent system prompt sections
- @workspace for project-aware explanations with real file references

vs

Claude Code — Project Scope

- Generates entire solutions across multiple files simultaneously
- Runs PAC CLI, manages git commits, iterates compile-fix loops
- Designs system prompts, topic structures, agent YAML
- Simulates conversations and identifies system prompt gaps
- Manages solution packaging and cross-environment deployment
- Invokes skills to enforce team conventions on every generation

CLAUDE IN PRODUCTION

CUSTOM CONNECTOR

Calling Claude from Power Platform

- Build a custom connector to the Anthropic Messages API (OpenAPI 2.0)
- 3 primary patterns: doc summarisation, intelligent routing, draft assistance
- Prompt caching with **cache_control** reduces high-volume flow costs by ~90%
- API key stays server-side — Canvas apps and portal JS always route through a flow
- Haiku for speed & volume · Sonnet for complex multi-constraint generation
- Deploy connector as a solution component — never create it manually in UAT/Prod

REVIEWING, TESTING & GOVERNING

Quality Gates for AI-Generated Assets

- AI failure modes by asset type: wrong column types, hardcoded values, missing CSRF tokens
- Use Claude Code to review Claude Code — cross-references schema via MCP
- Canvas: Power Apps Test Studio + Test Engine; E2E: Playwright with **data-control-name**
- ALM: every AI-generated asset belongs in source control (.pa.yaml, flow JSON, YAML)
- Run **pac solution check** before every UAT promotion — non-negotiable
- Agents: test prompt injection, distress signals, and out-of-scope attempts before deploy

AI DEV TEAM AT SCALE

Prompting, Onboarding & What's Next

- 5 prompting patterns: lead with output · specify prohibitions · reference real schema · ask for options · self-review loop
- Start with one developer for 30 days — build the skill library, then share it
- Two-layer team standard: official Microsoft plugins + enterprise standards repo in git
- The role shifts: less manual authoring, more requirements clarity & AI quality assurance
- 40-60% prototype time reduction within the first month — compounds as skills grow
- Individual dev environments per developer before scaling AI assistance to a team

THE FINAL WORD

You don't get replaced. You get more done.

- AI handles the mechanical, verbose, and repetitive — you handle the judgment
- Your domain expertise + AI generation speed = more capable than either alone
- The work that used to take a week now takes a day. A day now takes an hour.
- *"Build something worth building."*

• PART 3 – AI IN THE ALM PIPELINE

From commit to merge – AI on the team.

Generation is only the first hour of the workday. The next ten happen in version control, code review, bug triage, and release pipelines. Claude Code participates in every one – as a teammate that reads diffs, opens PRs, and closes incidents.



```

    > /git-status

    On branch main
    3 unstaged solution changes:

    M WorkOrders/Entities/fs_workorder
    + priority field nullable
    M Flows/Notify-Technician.json
    + Teams card body[3] removed
    ?? Tests/fs_workorder.spec.ts

    Claude: the flow change will break the existing notification – the removed card field is referenced by the trigger filter. Stage the schema edit, hold the flow change for review.

    FAIL Notify-Technician @ run 8842
    RuntimeError: cannot read 'location' of undefined

    Claude diagnosed:
    service-location lookup empty for work orders created via portal.

    — patch —
    - "@triggerOutputs()?.body?.fs_servicelocation.address"
    + "@coalesce(triggerOutputs()?.body?.fs_servicelocation.address, 'TBD – see customer record')"
```

WHERE AI FITS IN YOUR ALM TRACK



"The win isn't that Claude writes code – it's that Claude shows up to the standup. It reads the failing test before you do, drafts the fix, and the PR is waiting when you open your laptop. Your job moves from typing to judging."

YOU'VE SEEN WHAT'S POSSIBLE

Want to go deeper? Follow along.

This ebook is a preview. The full book goes **chapter by chapter** through every asset type – with **hands-on labs**, **real YAML**, real prompts, and the complete field service scenario **built from scratch**.

13

Chapters covering the full Power Platform AI stack

50+

Real prompts you can copy and use today

1

Complete field service scenario built end-to-end



Victor Dantas

Microsoft Power Platform MVP · Author

Power Platform architect and AI development practitioner. Follow along for new chapters, prompts, and updates as the book evolves.

STAY TUNED FOR UPDATES



Twitter / X
@thevictordantas




LinkedIn
/victordantas



Instagram
@thevictordantas

 **Full Book Coming Soon**

 **Download Skill Templates**