DEAN LAI · TGE INNOVATIONS

FIELD GUIDE · v1.0

The Engineer's AI Playbook

Built by engineers, for engineers.
A starter guide for leveling up your AI skill,
without leaving the shop floor behind.

INSIDE

deantge.com

By Dean Lai · Founder, TGE Innovations · April 2026

01 MANIFESTO

Why engineers should lead the AI conversation

I spent 10 years on the floor before I started building AI tools. Here's what I learned: the people closest to the process are the ones who'll make AI actually work — and most of them don't know it yet.

The model is the easy part.

Anyone can fine-tune a model in a weekend. The hard part is knowing which signal matters, which failure mode is worth predicting, and which SOP changes when the model is right. That's engineering knowledge. That's you.

You already have the hardest ingredient.

AI without domain context produces confident nonsense. The software people can't get that context from a textbook. You've been building it since your first commissioning job.

Waiting for "someone technical" to figure it out is a trap.

Every year you wait, the gap between engineers who use AI and engineers who don't gets wider. The ones who start now — even clumsily — will compound.

This playbook is a starting line, not a finish line.

Seven pages won't make you an AI engineer. They'll give you a map, a first sprint, and a set of tools that work. The rest happens in the community — and in the real work you bring back to it.

02 THE STACK

The engineer's AI stack

You don't need a CS degree to use AI well. You need to know what you already bring, and what to layer on top.

WHAT YOU ALREADY HAVE

- **Domain knowledge**
  You know why a bearing fails, what a good batch looks like, what a drifting sensor reads like.
- **Systems thinking**
  You already debug across PLC, HMI, SCADA, mechanical, and human layers. That's AI thinking.
- **Root-cause instinct**
  Your job has always been 'why did this really happen.' Same skill for model debugging.
- **Process data**
  Your plant has historian data most software engineers would kill for.

WHAT TO ADD

- **AI fundamentals**
  What a model actually is. Training vs inference. Supervised vs unsupervised. Enough to call BS.
- **One scripting language**
  Python. You don't need to be fluent. You need to read it and edit it.
- **Tool fluency**
  ChatGPT or Claude for thinking. Cursor or Copilot for coding. One notebook tool. That's the minimum.
- **A building habit**
  Ship one small thing a week. A script. A query. A prompt. Shipping is the only way this sticks.

> **Notice the left column is bigger. That's the point. Engineers are closer to useful AI than most people in tech.**

03 REAL WORK

3 problems engineers are solving with AI

Not demos. Not hype. Things engineers I know are actually shipping in SEA plants right now.

01 Shift-handover intelligence

Engineers are feeding shift logs, alarm history, and maintenance notes into an LLM to generate next-shift briefings. Saves 15–20 minutes per handover and surfaces patterns no one person could spot.

02 Spec-matching from PDFs

Datasheet, bill of materials, old drawing — dump them in, get a structured comparison. What used to eat a week of an application engineer's time is now an afternoon.

03 First-pass alarm triage

Instead of an operator Googling fault codes mid-shift, a local LLM fine-tuned on plant docs gives a three-line answer: likely cause, first check, escalation path. Not perfect. Better than blank.

> **None of these needed a PhD. They needed an engineer who understood the workflow well enough to define 'good.'**

Your 30-day AI skill sprint

Four weeks. One hour a day. No CS degree, no new job title. Just compounding skill.

WEEK 1 Get fluent with one AI tool

- Pick ONE: ChatGPT, Claude, or Gemini. Stop hopping.
- Run your real work through it — meeting notes, reports, SOPs.
- Learn what it's bad at. That's the most valuable lesson.

WEEK 2 Learn the vocabulary

- Prompt, context, token, hallucination, fine-tune, embedding, RAG.
- 30 minutes a day reading. Skip the math for now.
- Follow 3 practitioners on LinkedIn or YouTube. No influencers.

WEEK 3 Build your first micro-tool

- Pick a task you do weekly that could be a prompt or script.
- Build it. Badly is fine. Ship it to yourself.
- Use Cursor or Copilot if you've never coded.

WEEK 4 Show someone on your team

- Walk a colleague through what you built and what you learned.
- Their questions will show you the next thing to learn.
- Write a 200-word summary — post it, share it, send it to me.

Tools worth your time

The tool list gets longer every week. This is the short list I actually reach for. Pick one per category to start.

THINKING & WRITING

- ChatGPT / Claude / Gemini — daily driver for reasoning, writing, research.
- Perplexity — when you need AI with sources, not AI with opinions.
- NotebookLM — dump PDFs in, ask questions, get audio summaries.

BUILDING

- Cursor — AI-first code editor. The fastest on-ramp for non-coders.
- GitHub Copilot — pair-programmer inside your normal editor.
- Replit — browser-based; good for quick prototypes.

DATA

- Jupyter / Google Colab — run Python in a browser, no install.
- Grafana + InfluxDB — the OT engineer's time-series duo.
- DuckDB — query CSVs like they're a database. Underrated.

AUTOMATION

- n8n / Make / Zapier — connect tools without writing glue code.
- Gumloop — AI-first workflow automation, good for LLM chains.

How to think like an AI engineer

The gap between engineers who get AI and engineers who don't isn't tools. It's how they frame problems.

01 Start from the decision, not the data

Bad question: "What can I do with this data?" Good question: "What decision would change if I had a better answer?" Walk backward from there.

02 Good enough beats perfect

A 78% accurate model that ships beats a 95% model that never leaves the notebook. Engineers overfit on precision; AI rewards iteration.

03 The prompt is the spec

Writing a clear prompt is just writing a clear requirement. If you can spec a PLC routine, you can write a prompt.

04 Treat the AI like a new graduate

Smart. Fast. No context. Doesn't know your plant. Doesn't know what 'normal' looks like. Your job is to orient it — same as you'd orient a new hire.

05 Every output is a draft

Review like you review a junior's work: does it match reality, does it miss anything, would I sign my name to it? If not, iterate.

07 ACTION SHEET

The one-page version

Print this. Pin it above your desk. Revisit at the end of each week.

THIS WEEK

- Pick one AI tool. Use it daily.
- Identify one workflow to automate.
- Follow 3 real practitioners.
- Learn: prompt, context, RAG.
- Ship one small thing.

THIS MONTH

- Finish the 30-day sprint.
- Build one micro-tool for real work.
- Show it to a colleague.
- Write a short reflection.
- Join the community — share your build.

THIS QUARTER

- Level up one engineer on your team.
- Document one AI win publicly.
- Propose one pilot internally.
- Read one paper or case study monthly.
- Define your next skill to add.

> **The engineers who start now — even clumsily — are the ones who compound. The rest will catch up late.**

ABOUT THE AUTHOR

Dean Lai

Founder, TGE Innovations

10 years on the industrial floor before founding TGE.
I'm not writing this as a software person looking in.
I'm writing as an engineer who had to learn AI the hard way —
and wants the next engineer to have it easier.

THE COMMUNITY

Built by engineers, for engineers.

A free space for engineers leveling up their AI skill.
Real builds. Real questions. Real experience,
shared by people who actually ship.

WANT TO BUILD SOMETHING TOGETHER?

dean@tgecorporation.com

STAY SHARP

deantge.com
youtube.com/@deantge
instagram.com/dean.tge

This playbook is v1. Send corrections — I'll credit you in v2.