LEDEX · FARO · 6 MAY 2026

From Prompts to Pipelines

Automation that actually runs your business

IGNÁCIO CORREIA

Founder · SuperCognit

47

ChatGPT tabs open on your team's laptops — right now.

Busier with AI than ever.

No more automated than last year.

Because prompts don't compound. Pipelines do.

Tonight: how to get from one to the other — through Skills, MCP, and agents that run unattended.

Why most AI initiatives don't move the P&L

Three failure modes — see if you recognize any of them.

01

ARTISANAL

Prompt as silver bullet

A clever prompt copied around Slack. Works once. Breaks the second time the input shape changes. No version. No owner. No tests.

02

DEMO-WARE

Demos that never ship

An impressive notebook on the CTO's laptop. It dies in the gap between prototype and production: auth, retries, observability, eval, on-call.

03

HUMAN-IN-LOOP

Forever-supervised AI

"Automation" that requires a human to read every output before it's used. That's not automation — that's a very expensive intern.

There is a maturity ladder.

Most teams stop at rung 2. Real ROI lives at rung 5 and above.

> *Every rung up the ladder swaps a human action for a machine action — and compounds the value of every previous rung.*

Move up or stay artisanal.

01	Prompt	
02	Prompt + Context	
03	Tool Use	
04	Skills	
05	MCP	ROI
06	Agent Loop	ROI
07	Pipeline	ROI

The seven rungs

From a single prompt to a pipeline that runs your business.

Each rung swaps a human action for a machine one — and compounds the value of every rung below.

RUNG

01

Prompt

Ask once. Read once. Move on.

WHAT IT IS

A single instruction handed to a model. The classic ChatGPT moment. Useful for brainstorming, drafting, getting unstuck — anywhere a smart human would happily read the answer themselves.

CEILING

No memory, no tools, no state. The output is only as good as the human who copies it into the next system. Scales linearly with how much your team is willing to copy-paste — which is to say, it doesn't.

EXAMPLE

```
> Summarize this 12-page contract and flag anything unusual.
```

RUNG

02

Prompt + Context

System prompts. Examples. Retrieval.

WHAT IT IS

Wrap the model with a system prompt, few-shot examples, and retrieval over your own documents (RAG). Suddenly it sounds like your team — and quotes your data instead of inventing it.

CEILING

Still synchronous. Still in-chat. The model can read but it cannot do — it can't send the email, update the CRM, kick off the refund. A human is still the executor of every action.

EXAMPLE

```
system: You are a Tier-1 support agent for SuperCognit. Answer only from the docs in <ctx>. If unsure, escalate.
```

RUNG

03

Tool Use

Function calling. The model picks the action.

WHAT IT IS

Expose a small inventory of typed functions. The model decides which one to call, with what arguments, and reads the result. This is the first rung where AI stops describing the world and starts changing it.

CEILING

Tools are bespoke. Every project re-implements the same five integrations (CRM, email, calendar, search, DB). No discovery, no auth standard, no permissioning model. Every team rebuilds the same plumbing.

EXAMPLE

```
tools: [{ name: "create_invoice", input_schema: { customer_id: string, amount: number, currency: enum } }]
```

RUNG

04

Skills

Reusable capability bundles. Composable. Versioned.

WHAT IT IS

A skill is a folder: SKILL.md describing when to use it, plus scripts, templates and reference docs. The agent loads the right skill on demand. You write a 'pptx' skill once, and every future agent in your org knows how to make slides — properly.

CEILING

Skills give you composable know-how, but they still run inside one model's runtime. To reach across systems — Slack, Salesforce, your data warehouse — you need a transport. That's what MCP brings on the next rung.

EXAMPLE

```
skills/pptx/  ├── SKILL.md  ├── scripts/  └── references/   →  agent self-loads when 'create slides' is needed
```

RUNG

05

MCP — Model Context Protocol

One protocol. Any model. Any tool.

WHAT IT IS

MCP is the USB-C of AI tooling: a standardized protocol where any model can connect to any server exposing tools, resources and prompts. Auth, transport, discovery — solved once. n × m integrations collapse into n + m.

CEILING

Even with MCP, an agent that just answers when asked is still a chat box. To run the business, the agent has to take initiative — plan, act, observe, replan — without you holding its hand. That's the agent loop.

EXAMPLE

```
mcp.servers: { slack, gdrive, linear, postgres, supercognit-ops }   →   model calls them like local tools
```

RUNG

06

Agent Loop

Plan · Act · Observe · Replan.

WHAT IT IS

An autonomous loop: the agent sets a sub-goal, picks a tool, runs it, reads the result, decides what's next. It can spawn sub-agents for sub-tasks. Long-horizon work — research, triage, refactors, multi-step ops — finally fits in one task.

CEILING

Agents still wait for a human to start them. To actually run the business, the loop has to be triggered by the world: a new ticket, a calendar event, a webhook, 9am every Monday. That's the pipeline.

EXAMPLE

```
while not done:  plan = agent.plan()  result = agent.act(plan)  agent.observe(result)
```

RUNG

07

Pipeline

Unattended. Triggered. Observable. Owned.

WHAT IT IS

A scheduled or event-driven agent system, deployed like any other production service: triggers, retries, queues, evals, dashboards, on-call. This is the rung where AI stops being a feature and becomes infrastructure — work happens whether you're at your desk or at dinner.

CEILING

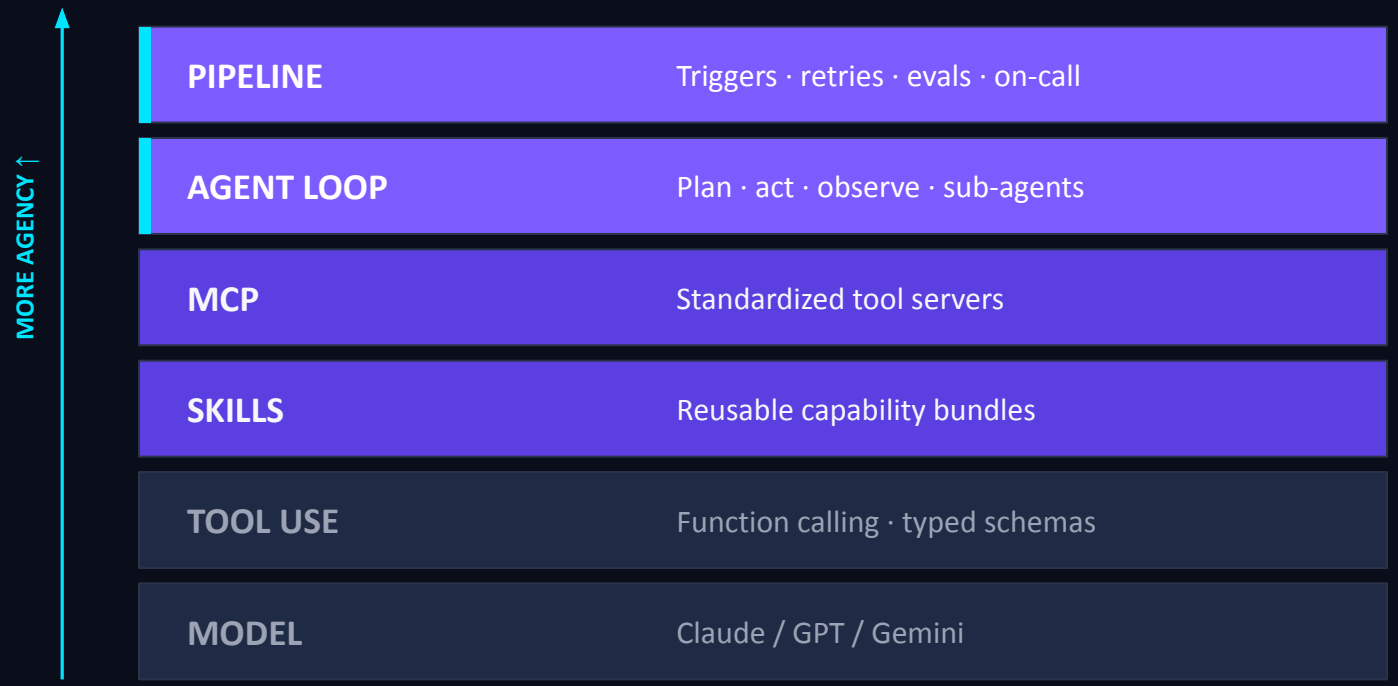
Now you have a real engineering problem: governance, cost control, drift, regression. Good news — these are the problems the rest of your stack already knows how to solve. Welcome to production.

EXAMPLE

```
trigger: new_email(label=billing) → agent.run(playbook=refund_v3) → audit_log → human_review_if(amount>500€)
```

How the rungs stack into a system

Each layer above only works because the layer below is solid.



DESIGN RULES

1. Tools are typed. The model never sees free-form fields it can corrupt.
2. Skills carry the know-how. The agent runtime carries the wiring.
3. MCP is the boundary — auth, scope and audit live on the server.
4. Agents own outcomes, not steps. The loop decides the steps.
5. Pipelines are services. They have SLOs, evals and an on-call rota.

From inbox triage to autonomous ops

B2B services company · 30-person ops team · invoice-billing chaos.

BEFORE — RUNG 1–2

5 humans · 3 tools · zero memory.

Ops team copy-pastes between Gmail, the billing tool and a shared spreadsheet. Each refund needs ~14 minutes and four context switches. ChatGPT used as a glorified phrasebook for tone-of-voice — nothing else.

AFTER — RUNG 5–7

1 pipeline · MCP into Stripe + Gmail + Notion.

Agent triages new billing emails on arrival. Skill bundles the refund playbook (policy, edge cases, tone). Stripe + Gmail via MCP. Refunds < 500€ auto-resolve; the rest land in a single human-review queue with a draft ready.

82%	auto-resolved
14m ⟶ 40s	median handle time
0	policy violations / Q

From SDR copy-paste to a pipeline that prospects

B2B SaaS · 8 SDRs · fragmented tooling, generic outreach.

BEFORE — RUNG 1–2

Prompts in a Notion page nobody updates.

SDRs paste a prospect name into ChatGPT, get a generic email, edit it, send. Personalization is 'their first name + their company.' Reply rates plateau. The 'AI workflow' is mostly a Slack folklore of clever prompts.

AFTER — RUNG 5–7

Agent loop with research → draft → review.

Pipeline pulls new leads from HubSpot via MCP, agent researches with web + LinkedIn skills, drafts a 3-touch sequence using the company's playbook skill, and queues it in the SDR's inbox for one-click send. SDRs become editors, not researchers.

3.4×	reply rate vs. control
11×	leads / SDR / day
<2 wks	to roll out

Five ways pipelines die in production

Build for these from day one — not when the on-call alert fires.

01	**Brittle prompts**	A prompt that worked on Monday breaks on Friday because the input shape drifted. Treat prompts as code: versioned, reviewed, tested.
02	**No observability**	If you can't see every tool call, every retry, every token, you can't debug. Ship traces and dashboards before you ship the agent.
03	**No eval loop**	Without a held-out evaluation set, you don't know if today's change made the system worse. Evals before scale, every time.
04	**Untrusted hallucinations**	Don't ask the model to be right; ask it to be checkable. Constrain outputs with schemas; verify with a second model or rule.
05	**No rollback**	Pipelines act on the world. Build a kill-switch and per-action reversibility. 'Undo' is a feature, not a footnote.

How to climb the ladder without falling off

The path that compounds — instead of accruing demo debt.

1

Start at rung 4, not rung 1

Don't ship a 'ChatGPT for X' wrapper. Identify a real, repeated workflow. Build a Skill for it.

2

Wrap your tools as MCP

Front your CRM, your data warehouse and your ops apps with MCP servers. Auth and audit live there.

3

Make it loop, then make it triggered

Get the agent loop solid first. Only then connect the trigger — webhook, schedule, queue.

4

Instrument before you scale

Tracing, evals, dashboards, alerts. If it doesn't have an SLO, it isn't production.

5

Humans at the boundaries

Approve high-stakes actions; sample low-stakes. Trust grows with every clean run, not with hope.

6

Compound, don't replace

Every new pipeline reuses the previous skills, tools and evals. That's the moat.

What to do on Monday morning

Three concrete moves. None require a budget approval.

1

PICK ONE

List the top 3 workflows your team copy-pastes through.

Pick the one with the most repetition and the clearest definition of done. That's pipeline #1.

2

WRITE ONE

Draft a SKILL.md for that workflow.

Just a markdown file: when to use it, the steps, the edge cases, the tone. You'll already see the agent of it.

3

CONNECT ONE

Stand up one MCP server for one of your tools.

Don't try to MCP your whole stack. One tool, real auth, real scopes. Iterate from there.

By Friday you'll know whether the workflow is a pipeline candidate — and you'll have the artifacts to prove it.

The tooling map — what fits where

Pick by the rung you're on, not by the marketing.

Higher rungs need code-first agency. The top-left is fine — until it isn't.

Obrigado.

*Now go **build** the pipeline.*

IGNÁCIO CORREIA

Founder · SuperCognit

Questions, war stories, work — open invitation.

supercognit.com
linkedin.com/in/igcorreia
ignacio@supercognit.com