Episode 068 Artifact

Claude Code Plugins: Standardizing Agentic Rollouts Across Your Team

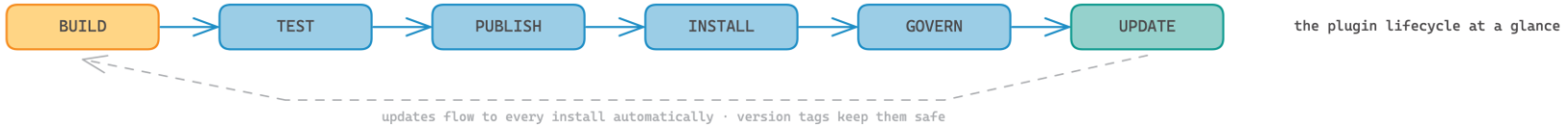


A leave-behind for listeners, from Cirrius Solutions

CIRRIUS TALK · EP 068

Claude Code Plugins: Standardizing Agentic Rollouts Across Your Team

2026-08-07 · A leave-behind for listeners, from Cirrius Solutions · Sources: official Anthropic + OpenAI docs and GitHub



the plugin lifecycle at a glance

1 · WHAT A PLUGIN IS

one folder, tracked in git, that packages every way you customize Claude Code

```
your-team-plugin/
├── .claude-plugin/plugin.json
│     the manifest: name, description, version
├── skills/
│     know-how Claude applies on its own
├── commands/
│     shortcuts a person types (/new-proposal)
├── agents/
│     preconfigured helpers for specific jobs
├── hooks/
│     scripts that run automatically on events
├── .mcp.json
│     connections to your other systems (MCP)
└── settings.json
      default settings applied on install
```

plugin.json

```
{
  "name": "your-team-plugin",
  "description": "Shared team
    skills + commands",
  "version": "1.2.0"
}
```

plugins can also carry code-intelligence servers, monitors, and small CLI tools

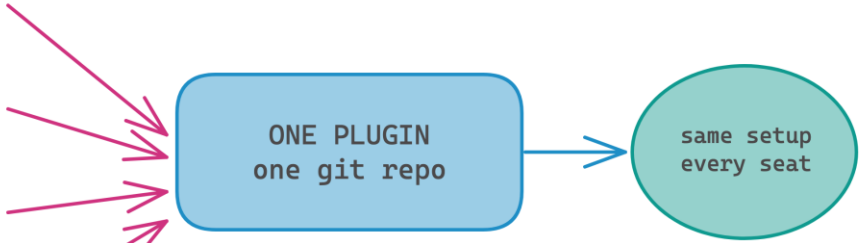
commands are namespaced (/plugin-name:command) so plugins never collide

2 · WHY PLUGINS FOR TEAM ROLLOUTS

from per-seat drift to one versioned artifact

without a package

- copy-pasted prompts
- snowflake configs
- manual updates
- no version history



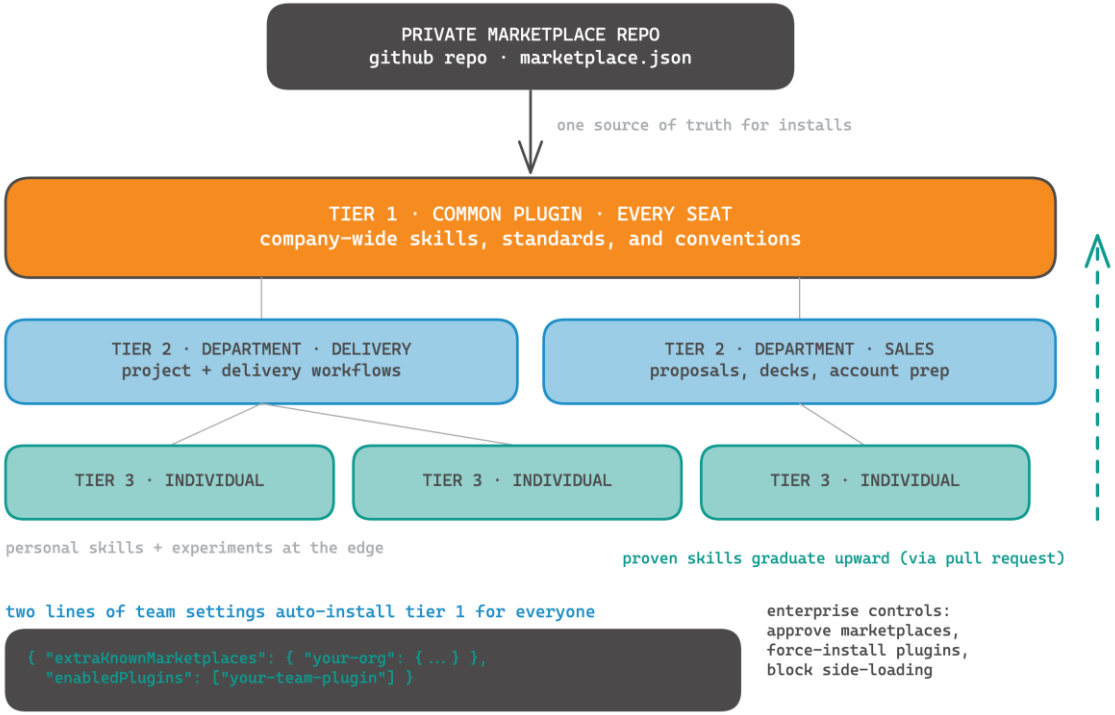
install once, update centrally
version numbers gate every change

a plugin turns 'everyone, set up your AI the same way'
from a request into a guarantee

the rollout becomes an artifact you version, not an email you hope people read

3 · THREE-TIER GOVERNANCE FOR TEAM PLUGINS

a marketplace is just a git repo with a catalog of approved plugins



two lines of team settings auto-install tier 1 for everyone

```
{ "extraKnownMarketplaces": { "your-org": { ... } },
  "enabledPlugins": ["your-team-plugin"] }
```

enterprise controls:
approve marketplaces,
force-install plugins,
block side-loading

4 · BUILD IT, SHIP IT, ROLL IT OUT

the how-to, in the order you'd actually do it



.claude-plugin/marketplace.json

```
{ "name": "your-org",
  "plugins": [
    { "name": "your-team-plugin",
      "source": "./your-team-plugin",
      "version": "1.2.0" } ] }
```

a release is just a git tag

```
git tag your-team-plugin--v1.2.0
version ranges resolve to these tags
```

install scopes:
user · all your projects
project · shared via the repo
local · just you, this repo
managed · org policy, can't disable

5 · DO'S AND DON'TS

DO

- ship modular pieces users can toggle
  (so unused ones don't weigh Claude down)
- version every release (semver)
  tag it plugin-name--vX.Y.Z
- pin cross-plugin dependencies
  ("version": "~2.1.0")
- test locally (--plugin-dir) and validate
  before you publish
- write the README · descriptions
  drive discovery

DON'T

- put skills/ or commands/ inside
  .claude-plugin/ (manifest only)
- use local side-loading as your rollout
  (that's what marketplaces are for)
- force-enable org-wide without
  a real policy decision
- leave deps unversioned and get
  broken by an auto-update

6 · THE OPENAI SIDE · CODEX IN BRIEF

SAME IDEAS, DIFFERENT NAMES

- AGENTS.md · layered instruction files (global, repo, folder)
- skills · SKILL.md folders on the open agent skills standard (the Anthropic-originated format)
- plugins · .codex-plugin/plugin.json manifest bundles skills + MCP servers
- config.toml + Team Config · settings checked into the repo
- enterprise · requirements.toml (locked policy) + managed_config.toml (default settings)

KEY DIFFERENCES

- distribution · universal plugin directory shared with ChatGPT (OpenAI / Workspace / Personal) vs git marketplaces
- Codex plugins carry skills + MCP only · hooks, agents, and config travel via git Team Config instead
- plugin update semantics undocumented on the Codex side · Claude Code auto-updates from the marketplace
- custom prompts deprecated in favor of skills

THE CONVERGENCE HEADLINE
Codex CLI 0.146.0 (Jul 2026) can consume Claude Code plugin marketplaces · the skills format is becoming the cross-vendor standard · build once, run on both

MCP = Model Context Protocol, the open standard for connecting AI agents to outside tools and data
sources: code.claude.com/docs · claude.com/blog · learn.chatgpt.com/codex · developers.openai.com · github.com/openai/codex · github.com/anthropics

Thanks for watching.

Explore all three Cirrius Talk series:

Agentic AI
Designing, building, and operating AI agents that do real work, minus the hype.

Salesforce at Scale
Real-world lessons for building and running Salesforce that scales with the business.

The Military Minute
Veterans and military families on service, transition, and the people who help.

Questions, need help, or a topic idea? Reach out: CirriusTalk@CirriusSolutions.com