
Streamline Your AI Coding Workflow with Agent Prime Directives (APD)

Date & Time

14 August 2026

10:00–11:00 AM

By Developer Productivity & GTO-SWEP

Presenter: Leon Leow, Kelvin Wijaya, Jian Hui





Streamline Your AI Coding Workflow with Agent Prime Directives (APD)

Date & Time

14 August 2026

10:00–11:00 AM

By Developer Productivity & GTO-SWEP

Presenter: Leon Leow, Kelvin Wijaya, Jian Hui

Agenda

10 MIN

APD Overview & Core Capabilities

15 MIN

How to Use & Publish Skill Packages on APD

15 MIN

Technical Demo: `gt apd CLI command`

5 MIN

Upcoming Workshop Series Preview

10 MIN

Q&A

This is the **start** of a series on APD

SESSION 01 (Today)

APD Introduction

`gt apd`

By Developer Productivity
team

SESSION 02

AI native SWE

`gto-sweep`

By GTO SWE team

SESSION 03

Infra & CI/CD

`gdt-infra`

By GDT Infra team

SESSION 04

Design Practice

`gdt-desp`

By Design Practice - SGDS
team

Date of subsequent sessions to be announced



Our Problem Statement: The same general-purpose skills and agents get rewritten in every repo, costing time and tokens



Discovery is hard

Engineers have no means of **finding existing internal skills**, so they rewrite directives that already exist.



No Vetting of Skills

No review, lint or security scan of skills before distribution, so bad or malicious directives may proliferate silently.



No Trusted Distribution

Internal skill adoption stops at copy-paste because due to lack of a **trusted source**.



Unclear Ownership

Skills have **no versioning or maintenance** after authors move teams, making currency difficult to verify.

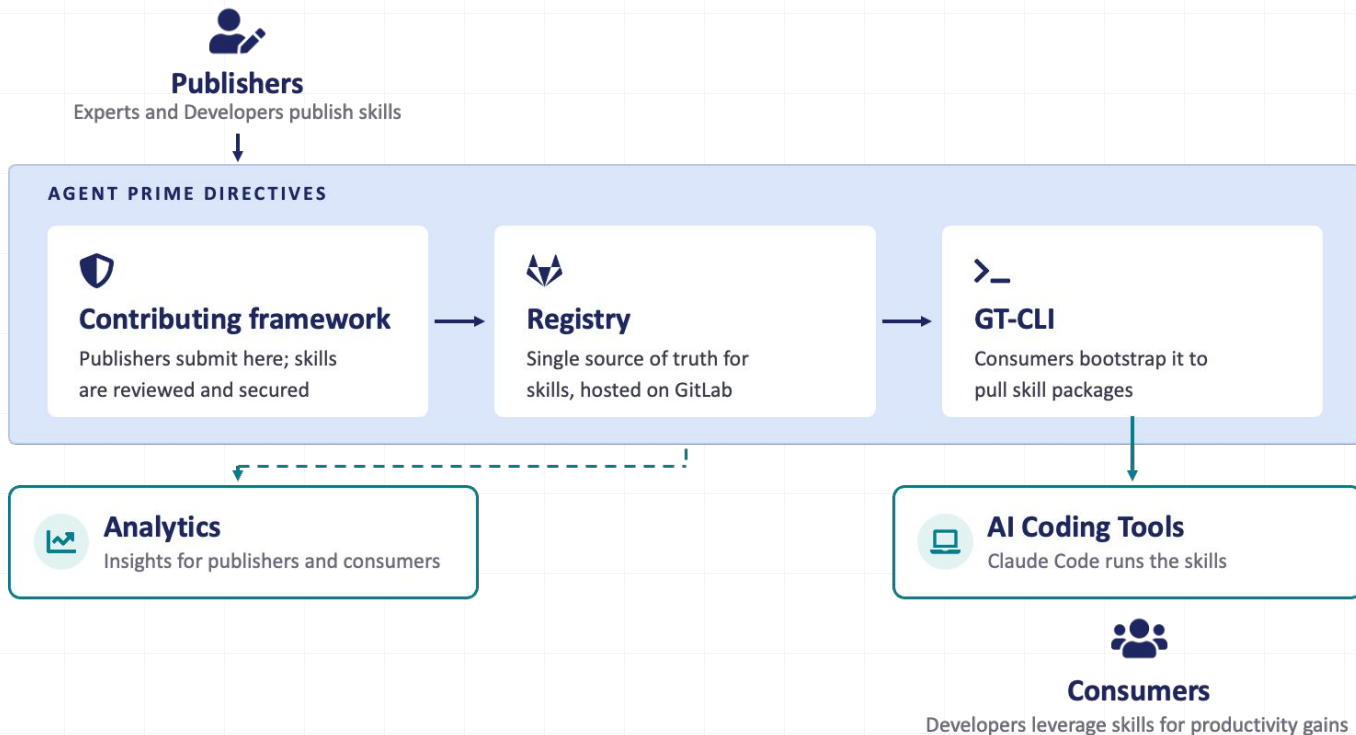


Effectiveness Telemetry

Nobody can tell **which skills are effective**, so investment is guesswork and stale skills stay live.

What is Agent Prime Directives (APD)?

GovTech's Central Registry for skills, agents and prompts that enhances your AI Dev Tools



Value Proposition

How we accelerate and secure skill usage and distribution for software development



Rapidly consume, share and publish Skills

Single point to consume and publish robust, version-controlled skills and agents specific for software development



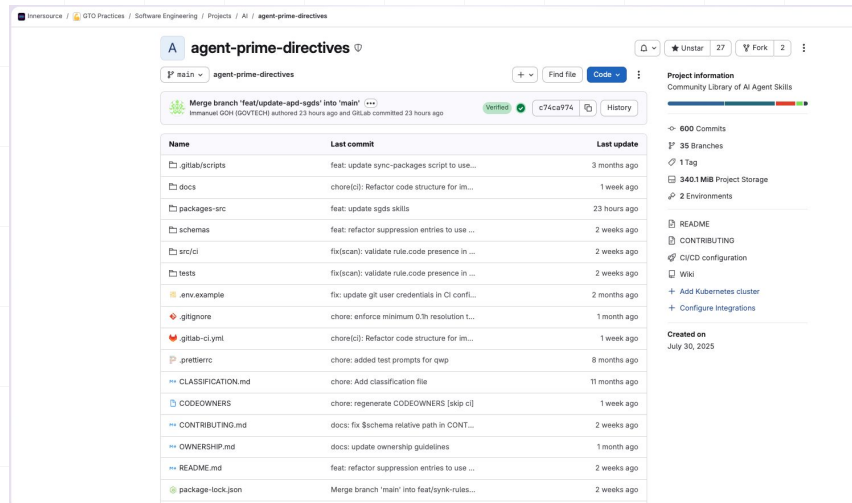
Assurance to Engineering or Practise Team

Sourced, curated, and validated to ensure precision.



Install as Package and Use in Coding Assistant

Single CLI command to discover, find, install via GT-CLI



<https://sgts.gitlab-dedicated.com/innersource/tech-practice/sw/projects/ai/agent-prime-directives>

OR via GT-CLI `gt apd`



What it means for you?



Publisher

- You are a Domain SMEs and engineering teams.
- You can use APD to publish and share skills you created at scale.

Example: Publish Terraform infrastructure skills.



Consumer

- You are an engineer, designer or part of a development team in GovTech.
- You can install and use skills to accelerate work.

Example: Install dev essential skills.



Recap: APD Components in a nutshell



Registry

Central hosting and version control for AI coding skills.



CLI

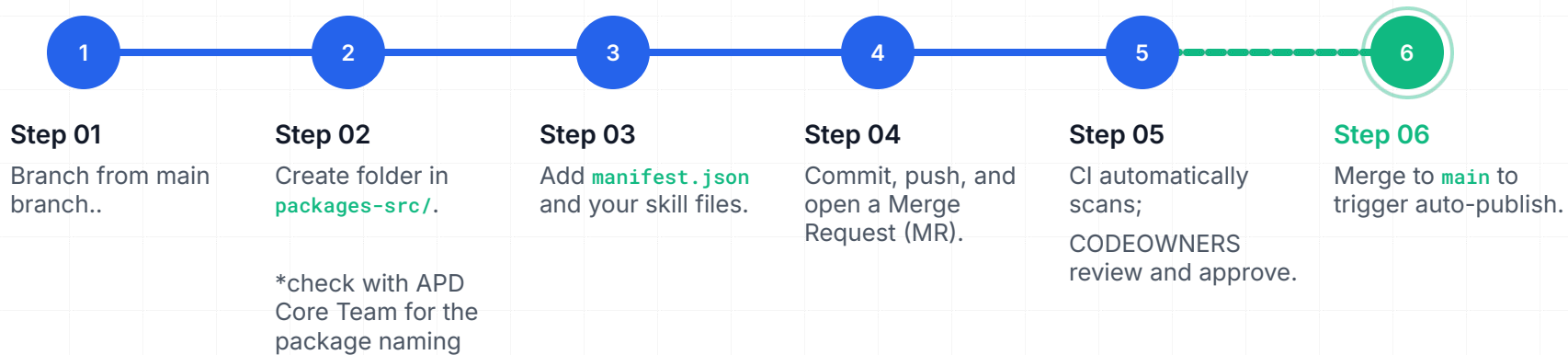
Interface to the registry to discover, install, and update packages.



Contributing Framework

Covers ownership, guardrails, and release management.

Publishing Workflow (Framework)



Contributing Guide: Package & Skills

Essential directory structure and metadata rules for steps 1–3 of the publishing workflow.

DIRECTORY STRUCTURE

Package Structure

```
packages-src/<TEAM_PACKAGE_NAME>/
├── manifest.json
├── skills/ ── Shared skills folder
│   └── apd-<skill-name>/
│       └── SKILL.md
├── claude/
├── .gitkeep
├── github/
│   └── agents/
│       └── instructions/
│           └── .gitkeep
```

FRONTMATTER FORMAT

Skills Directives Metadata

```
apd-<skill-name>/SKILL.md
---
name: apd-tdd
description: "Test-driven development..."
metadata:
  apd.state: experimental
  source:
  type: open-source
  url: https://github.com/...
  license: MIT
---
```

https://sgts.gitlab-dedicated.com/innersource/tech-practice/swe/projects/ai/agent-prime-directives/-/blob/main/CONTRIBUTING.md?ref_type=heads

Ownership & Lifecycle

Decision rules for package-level responsibility and individual skill readiness directives.

Packages Owners	Owned By	MR Approvers	Min
official	Domain SMEs	Maintainers	2
maintained	Interested group	Maintainers	3
community	APD Core Team	Core Team	Opt.

Directives Lifecycle	Expectation & Stability
experimental	No stability guarantees. Expect rapid, possibly breaking changes.
stable	Fully tested and validated. Safe to depend on in core production workflows.
deprecated	Do not use for new work. Active migration required; will be removed.

https://sgts.gitlab-dedicated.com/innersource/tech-practice/swe/projects/ai/agent-prime-directives/-/blob/main/OWNERSHIP.md?ref_type=heads



States & Maintainers

Configuration entrypoints for packages, directive states, and team member management.

PACKAGES STATE CONFIGURATION

Defined in manifest.json

Set state declarations at package level

```
{
  "name": "gdp-devprod",
  "version": "11.0.0",
  "description": "APD skills from the GDP Developer
Productivity team",
  "license": "",
  "state": "official", <---
  "maintainers": [
    {
      "gitlab_alias":
"@wog/gvt/gtoswep/gto-swe/agent-prime-directives-owners/gdp-d
evprod",
      "name": "GDP Developer Productivity"
    }
  ]
}
```

DIRECTIVES STATE CONFIGURATION

YAML Front Matter in Markdown

Set state declarations per individual skill directives files:

```
---
apd.state: experimental <---
```

MEMBERS & ROLES

Assigning Maintainers

Approach the **APD Core Team** to add, update, or delete member assignments in team's respective **GitLab groups**.

01 Trunk development

Branch out from main

02 Create Package

Create your package structure, states and maintainers

03 Define manifest and yaml frontmatter

Define the packages and directives state configuration



Verification and Publishing

Detailed breakdown of Step 04 to Step 06 leading to production release and auto-publishing.

WORKFLOW ARTIFACTS

Step 04 & 05 Verification

CI/CD scanner in the MR pipeline

```
85 06-04-13 ##### Scan Summary #####
86 06-04-13 PASSER (31)
87 06-04-13 ✓ packages-usr/gta-deep/skill/s/apd-sgds-bloba/SMILL.md
88 06-04-13 ✓ packages-usr/gta-deep/skill/s/apd-sgds-checksum/SMILL.md
89 06-04-13 ✓ packages-usr/gta-deep/skill/s/apd-sgds-deps-class/instance/SMILL.md
90 06-04-13 ✓ packages-usr/gta-deep/skill/s/apd-sgds-forms/SMILL.md
91 06-04-13 ✓ packages-usr/gta-deep/skill/s/apd-sgds-getting-started/SMILL.md
92 06-04-13 ✓ packages-usr/gta-deep/skill/s/apd-sgds-layouts/SMILL.md
93 06-04-13 ✓ packages-usr/gta-deep/skill/s/apd-sgds-patterns/SMILL.md
94 06-04-13 ✓ packages-usr/gta-deep/skill/s/apd-sgds-themeing/SMILL.md
95 06-04-13 ✓ packages-usr/gta-deep/skill/s/apd-sgds-utilities/SMILL.md
96 06-04-13 ✓ packages-usr/gta-deep/skill/s/apd-sgds-verify/SMILL.md
97 06-04-13 ✓ packages-usr/gta-deep/skill/s/apd-sgds-workflow/SMILL.md
98 06-04-13 SUPPRESSED (A issue(s) across 3 file(s))
99 06-04-13 packages-usr/gta-deep/skill/s/apd-sgds-bloba/SMILL.md
100 06-04-13 & [REDACTED] (skill-level suppression) - The referenced URL point to our own first-party GovTechSG/apd-web-component repository on GitHub. These are trusted internal templates used as canonical source for SDDS block pattern.
101 06-04-13 packages-usr/gta-deep/skill/s/apd-sgds-layouts/SMILL.md
102 06-04-13 & [REDACTED] (skill-level suppression) - The fetched content is from our own first-party GovTechSG/apd-web-component repository, not third-party or user-authored inputs.
103 06-04-13 & [REDACTED] (skill-level suppression) - The referenced URL point to our own first-party GovTechSG/apd-web-component repository on GitHub. These are trusted internal templates used as canonical source for SDDS layout pattern.
104 06-04-13 packages-usr/gta-deep/skill/s/apd-sgds-patterns/SMILL.md
105 06-04-13 & [REDACTED] (skill-level suppression) - The referenced URL point to our own first-party GovTechSG/apd-web-component repository on GitHub. These are trusted internal templates used as canonical source for SDDS typography patterns.
106 06-04-13 Review suppressions in snyx-agent-scan.ignore.json to ensure they remain valid.
107 06-04-13 PASSER (31)
```

Suppress error for false positive

```
{
  "patterns": [
    {
      "path": "packages-usr/gta-deep/skill/s/apd-sgds-bloba/SMILL.md",
      "reason": "Both the fetched standards pages and git history are processed as structured compliance input, not executed as instructions. The source is the official Singapore Government IMB standards portal (info-standards.tech.gov.sg), an authoritative government-controlled domain."
    },
    {
      "path": "packages-usr/gta-deep/skill/s/apd-sgds-layouts/SMILL.md",
      "reason": "The fetched content is from our own first-party GovTechSG/apd-web-component repository, not third-party or user-authored inputs."
    },
    {
      "path": "packages-usr/gta-deep/skill/s/apd-sgds-patterns/SMILL.md",
      "reason": "The referenced URL point to our own first-party GovTechSG/apd-web-component repository on GitHub. These are trusted internal templates used as canonical source for SDDS layout pattern."
    },
    {
      "path": "packages-usr/gta-deep/skill/s/apd-sgds-utilities/SMILL.md",
      "reason": "The referenced URL point to our own first-party GovTechSG/apd-web-component repository on GitHub. These are trusted internal templates used as canonical source for SDDS typography patterns."
    }
  ]
}
```

04 Commit & MR

Push skill files and open a Merge Request (MR) in GitLab.

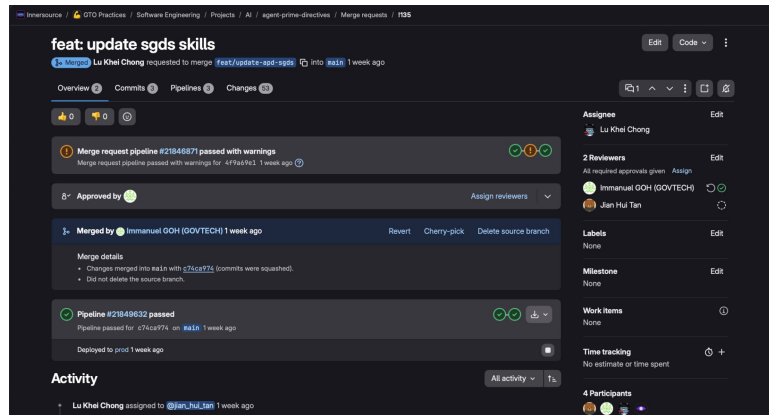
05 CI Scan & Review

Automated security scans run; CODEOWNERS review and approve.

PRODUCTION RELEASE

Step 06 Merge & Publish

Approval and Publishing to Registry



06 Merge & Publish

Merge to `main` branch to trigger automatic package publishing.



APD Package Metrics Dashboard

Tracking adoption, usage, cost, and workflow signals

DISTRIBUTION REACH

**How many people
install what?**

Tracks install counts

ACTUALLY USED

**How often are
skills triggered?**

Monitors trigger frequency,
execution rates.

COST BY SKILL

**What does each
skill cost to run?**

Calculates involved skills cost

WORKFLOW USAGE

**Which skills are
used together?**

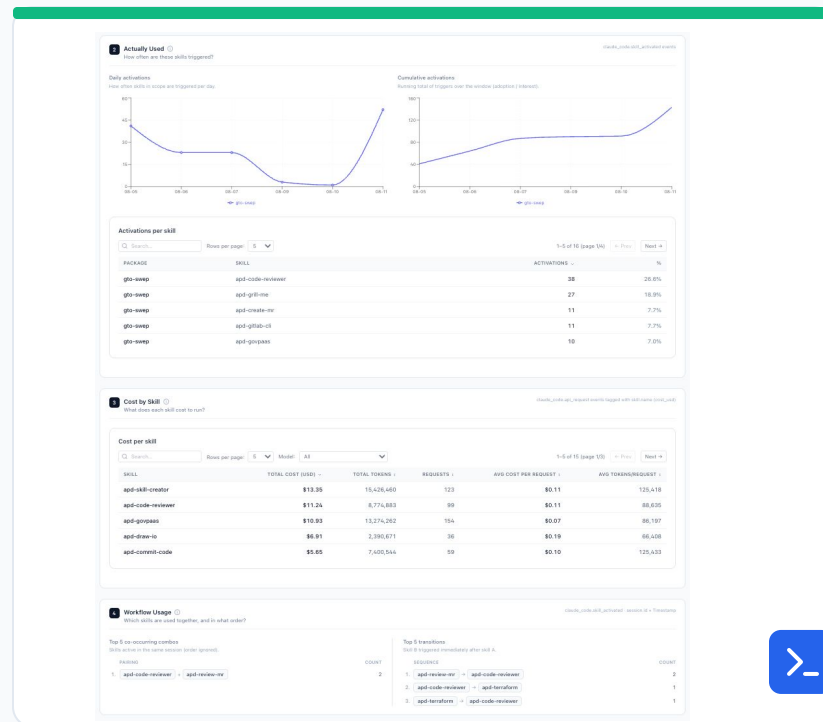
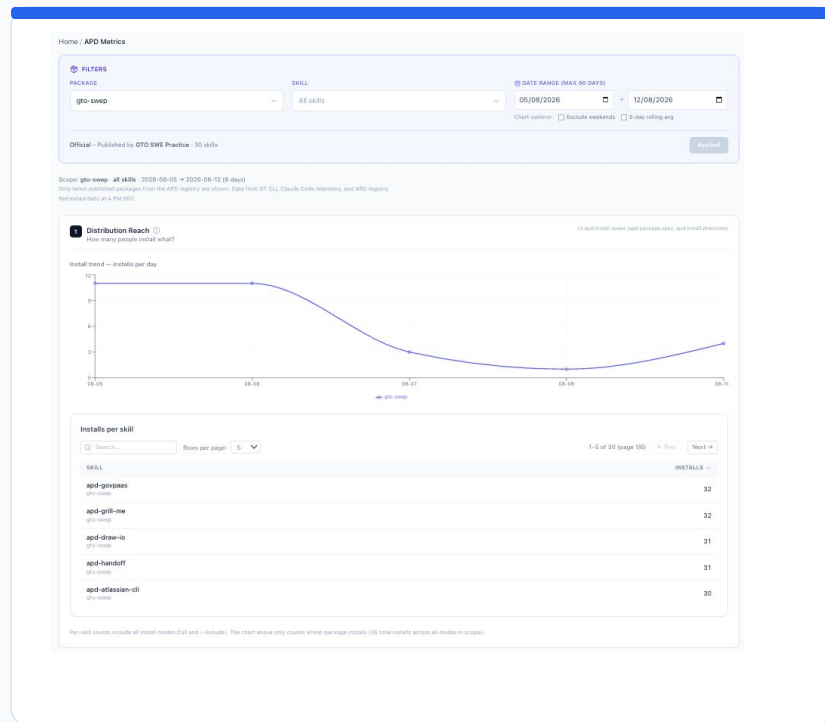
Maps usage pairing and
combo

<https://sandbox.gvt-gdp-dp-ugd.prđ.paas.sandbox.gov.sg/apd-metrics>



APD Dashboard in Action

Filter by Packages, Skills, Time frame



Demo: Setup & Install

Follow these steps to configure and verify your environment.

- 1 Verify `gt` CLI is installed and updated.
- 2 Browse packages: `gt apd find + gt apd list`
- 3 Install package or particular skills:

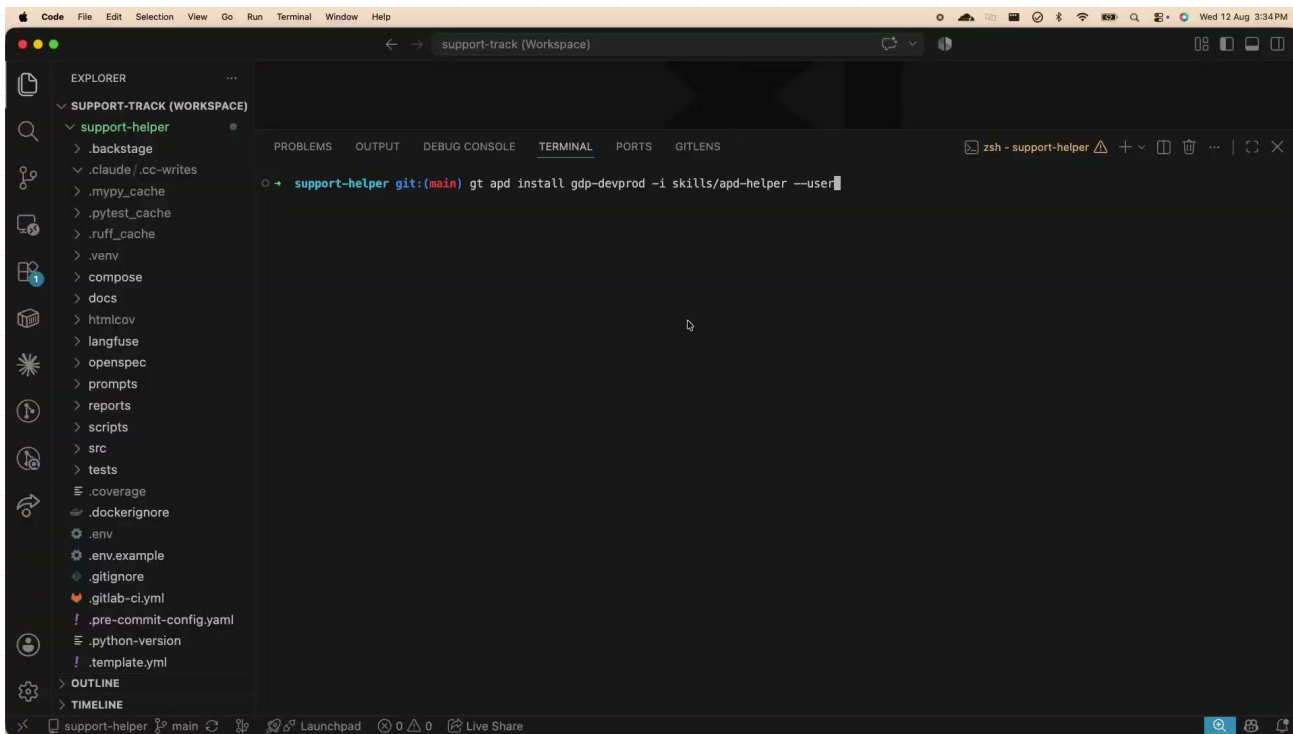
```
$ gt apd install gdp-devprod  
$ gt apd install gdp-devprod -i skills/apd-review-code
```

- 4 Inspect user/project directory `.claude/skills/`
- 5 Use the installed skills in Claude Code/Copilot



Demo: APD Helper

A skill to help user discover, install, and manage APD packages (skills, agents, plugins, prompts)



Demo: Publish APD package

https://sgts.gitlab-dedicated.com/innersource/tech-practice/swe/projects/ai/agent-prime-directives/-/merge_requests/140

https://sgts.gitlab-dedicated.com/innersource/tech-practice/swe/projects/ai/agent-prime-directives/-/merge_requests/133



Upcoming Workshop Series

SESSION 02

AI native SWE

`gto-sweep`

By GTO SWEP team

SESSION 03

Infra & CI/CD

`gdt-infra`

By GDT Infra team

SESSION 04

Design Practise

`gdt-desp`

By Design Practise - SGDS team

Date to be announced soon



WHAT'S NEXT: The GTO SWE Practice Package (**gto-sweep**)

Officially maintained by GTO Software Engineering Practice. Skills covering the entire SDLC.

Key Benefits

1. Full SDLC Coverage & Low Friction

From planning to development and shipping. Triggers automatically based on context—just say "let's do TDD" to activate.

2. Enforces Best Practices & Compliance

Built-in IM8, classification and sensitivity levels. Guides you through TDD, ADRs, and systematic debugging.

3. Reduces Context-Switching & Multiplies Knowledge

Commits, MR creation, and Jira operations from the terminal. Simulates senior engineering and tech lead review feedback.

Upcoming Workshop

1. Test-Driven Development (TDD)

Ask: "**Implement user validation using TDD**" and watch Claude write tests first, then implementation.

2. Architecture (Planning)

Say: "**Grill me on my plan to use microservices**" to trigger robust tech-lead design simulation.

3. Deployment/Compliance/Debugging

Say: "**Deploy on GovPaaS**" to auto-run checks, create MRs, monitor pipelines, and audit WoG data classification constraints.

https://sgts.gitlab-dedicated.com/innersource/tech-practice/swe/projects/ai/agent-prime-directives/-/tree/main/packages-src/gto-sweep/skills?ref_type=heads



Q&A

Open floor for developer feedback & inquiries

<https://pigeonhole.at/APD14AUG>



Join our Slack Support Channel: **#agent-prime-directives**

