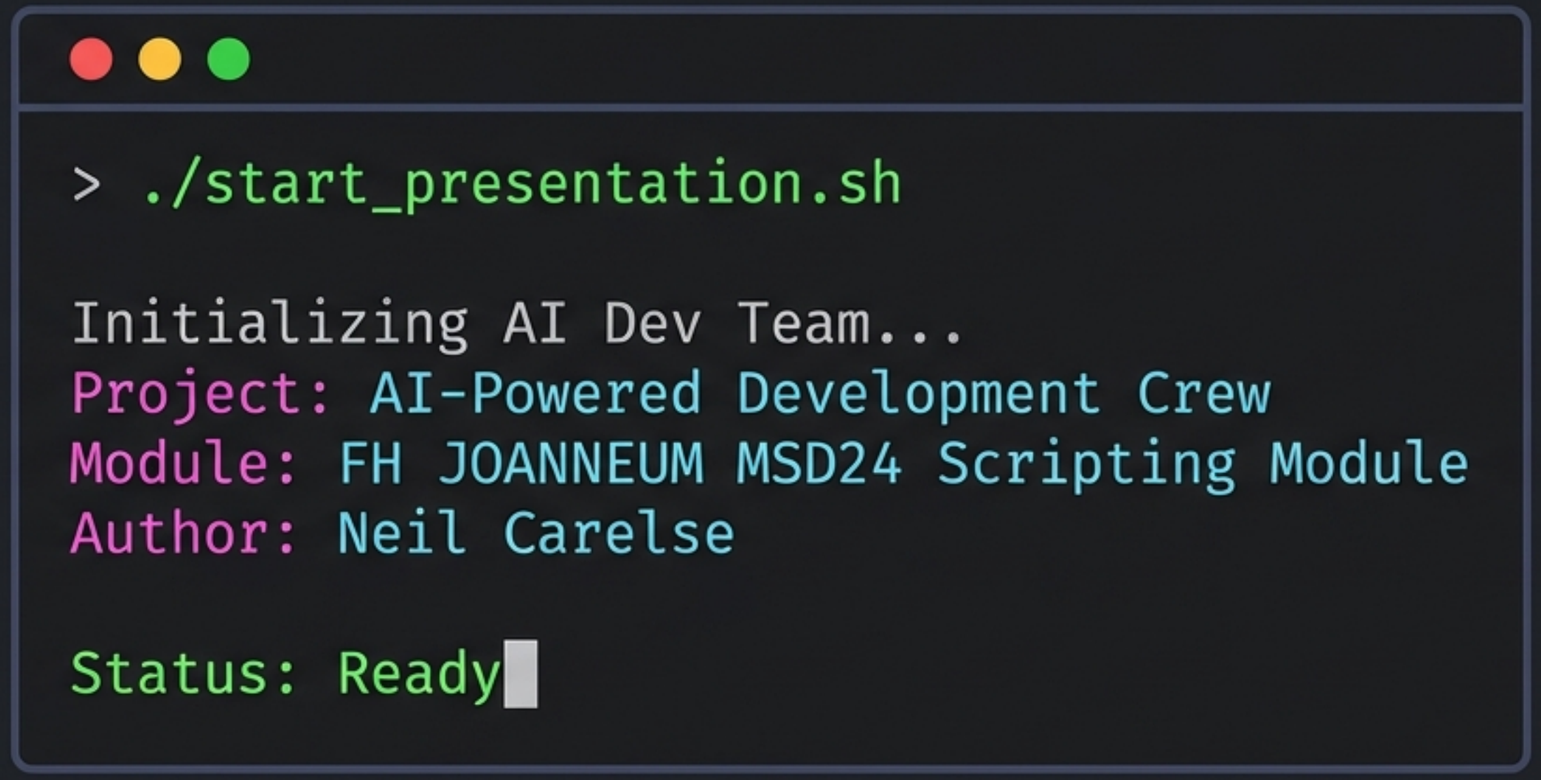

> ./start_presentation.sh
Initializing AI Dev Team...
Project: AI-Powered Development Crew
Module: FH JOANNEUM MSD24 Scripting Module
Author: Neil Carelse
Status: Ready

> man ai-dev-team

DESCRIPTION

An AI-powered development crew designed to read GitHub issues, analyze the codebase, and propose implementation plans.

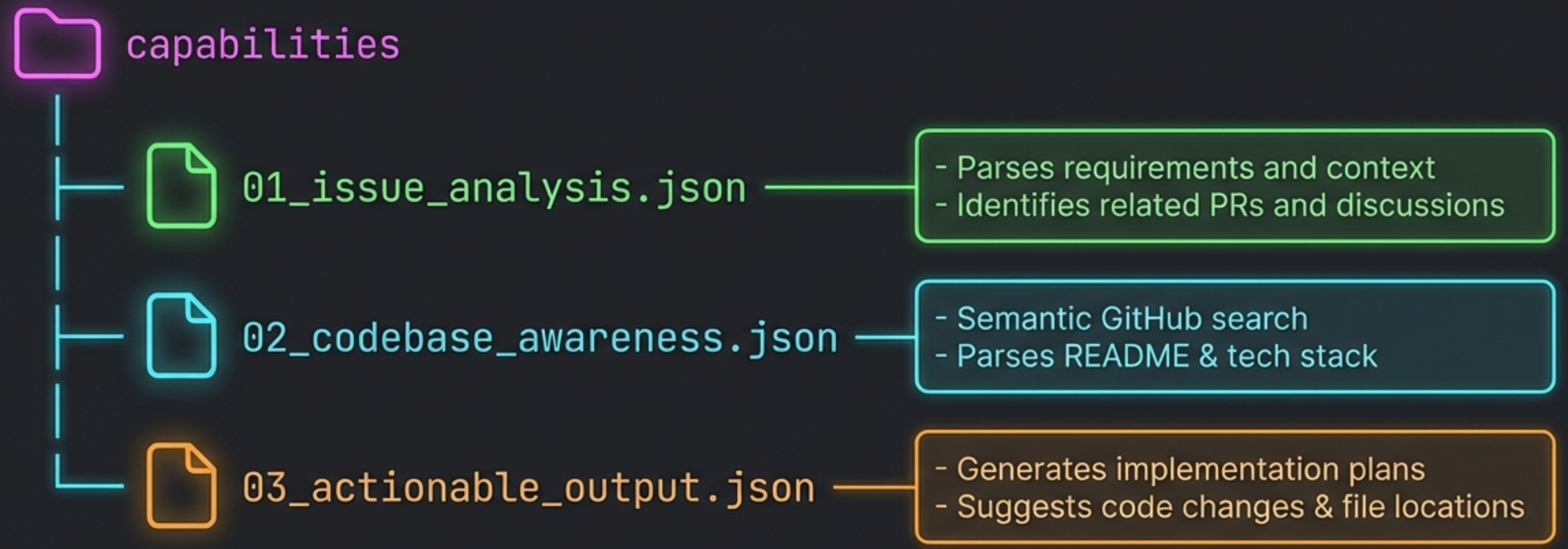
STACK

Built on Python 3.10+, CrewAI, and GitHub API integration.

OBJECTIVE

To enable developers to point an ‘AI dev team’ at any GitHub repository and instantly generate structured help and solutions.

```
> tree ./capabilities
```



> ls -R /structure

scripts/

main.py Single entry point

github_crew.py Multi-agent engine

example_github_issue.py Lightweight scanner

crew_runner/

Tools for git ops, logging, safety checks

docs/

Quickstart, guides, troubleshooting

> cat .env

```
GITHUB_TOKEN      = ghp_your_token_here
GITHUB_REPO       = owner/repo
GITHUB_PROJECT_ID = PVT_your_project_v2_id  # optional
OPENAI_API_KEY    = your_openai_key_here    # optional
```

File is git-ignored for security.

Project ID enables V2 Board integration.

> run setup.sh

REQUIREMENTS

- ✓ Python 3.10+ (Recommended)
- ✓ GitHub Account
- ✓ Personal Access Token (PAT)
- ✓ Scopes: repo, read:org

TERMINAL

```
01 $ git clone git@github.com:Neotro
02 onProductions/ai-dev-team.git
03 $ cd ai-dev-team
04 $ python3 -m venv .venv
05 $ source .venv/bin/activate
06 $ pip install -r requirements.txt
07
08
09
```

```
> python scripts/main.py
```

One-Command Initialization

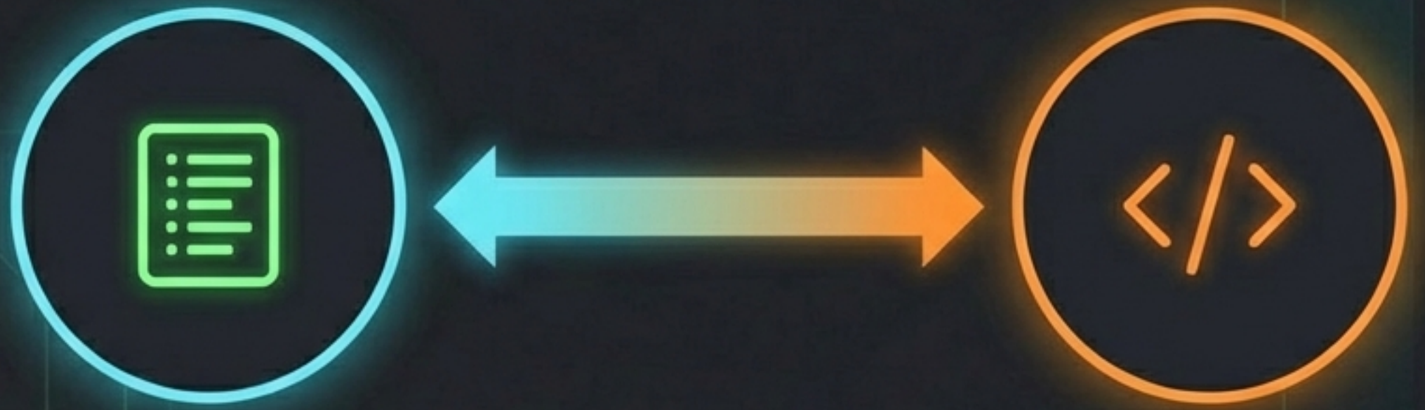
Designed to reduce friction: **Clone, Run, Analyze.**

```
> python scripts/github_crew.py
```

Multi-Agent Execution Mode

```
python3 scripts/github_crew.py <issue_id> <owner/repo>
```

Agent 1:
GitHub Issue Manager



Agent 2:
Code Analyst

Reads issue, structures problem.

Analyzes code, proposes plans.

```
> python scripts/example_github_issue.py
```

Single-Agent Analysis Mode

Use Case: Single Issue

```
$ .../example_github_issue.py
owner/repo 123
```

Quick analysis of a specific issue ID.

Use Case: Repository Scan

```
$ .../example_github_issue.py
owner/repo
```

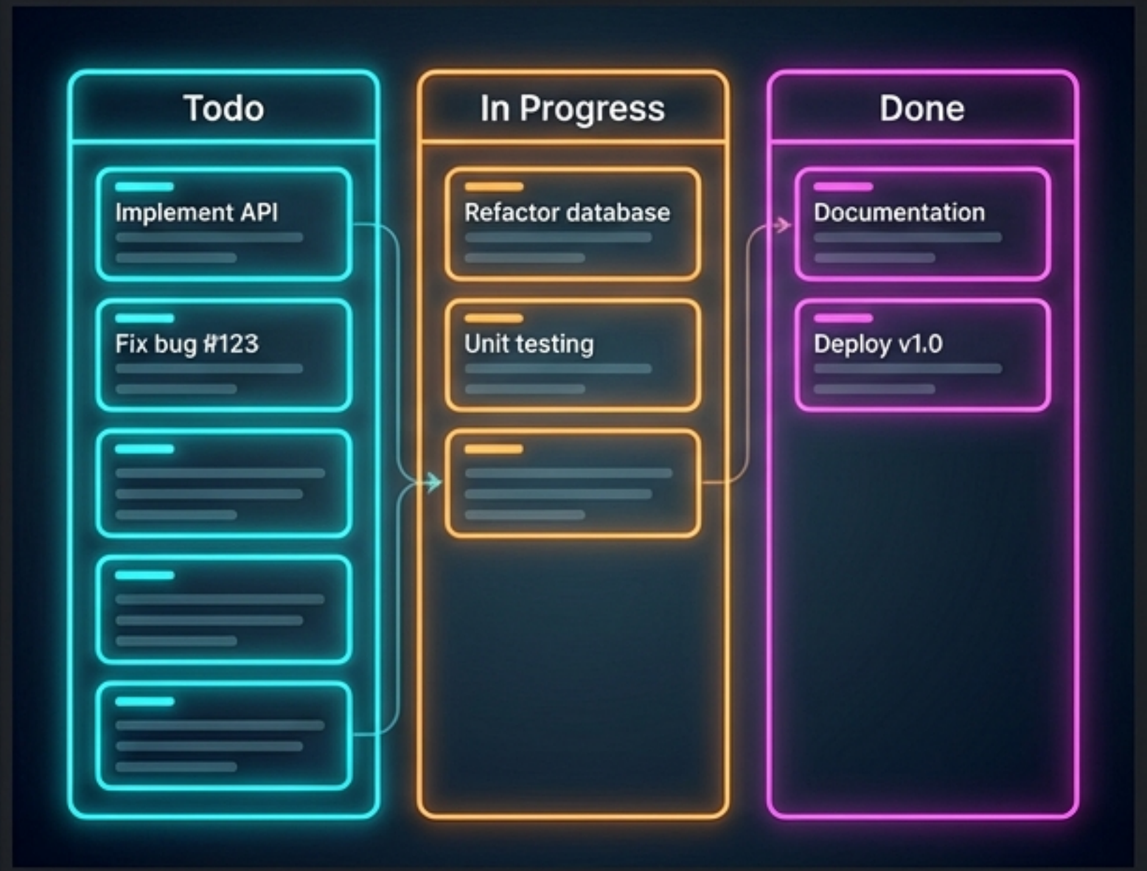
Scanning top open issues.

A lightweight, fast alternative to the full multi-agent crew.

> import projects_v2

GitHub Projects V2 Integration

JetBrains Mono



JetBrains Mono

Logic

- Connects via `GITHUB_PROJECT_ID`
- Prioritizes based on status/priority fields
- Focuses AI on active board items

Helper Tools

- `test_project_access.py`
- `test_graphql_projects.py`
- `test_project_fields.py`

> import safety

Workflow Protection & Verification

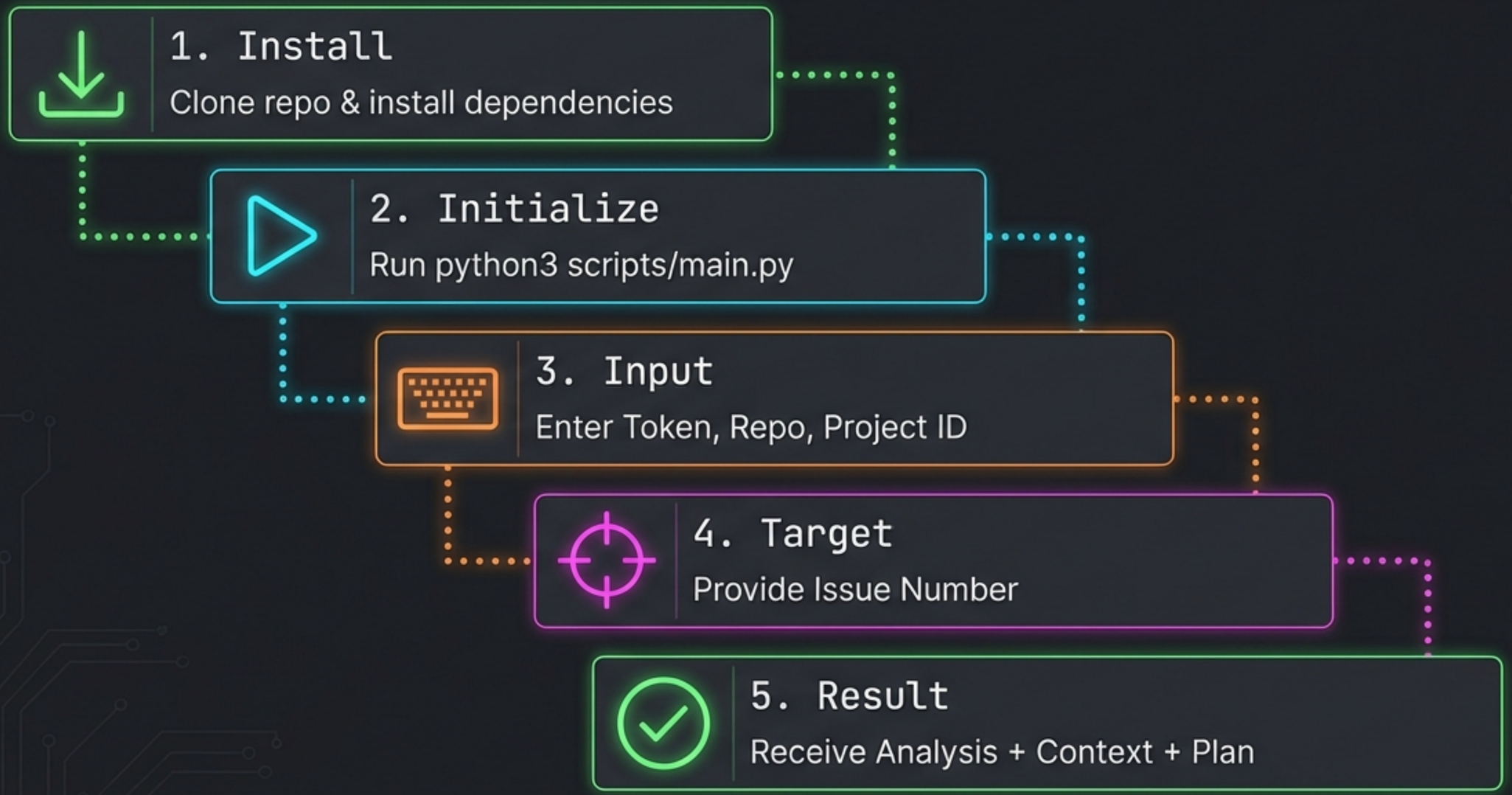
Verification
Coverage & testing helpers included

Non-Destructive
Changes expressed as patches/diffs

Documentation available in /docs for troubleshooting.

> sh demo_scenario.sh

Typical User Workflow



```
> git status

On branch main

Changes to be committed:
 + Automated Context: Eliminates manual code diving
 + Structured Output: Consistent implementation plans
 + Scalability: Works on any GitHub repo
```

> exit(0)

$ _

Repository: github.com/NeotronProductions/ai-dev-team
Status: Process completed successfully.