

# GitHub Agentic Workflows

TECHNICAL PREVIEW



MONA LISA  
Mascot  
GitHub

MADE WITH ♥ BY

GitHub Next & Microsoft Research

@pelikhan @dsyme @eaftan @mrjf @mnkieferr @moussaka @lpcox

# Agenda



From Automation  
to Intelligence



GitHub Agent  
Workflows



Demo



Q&A

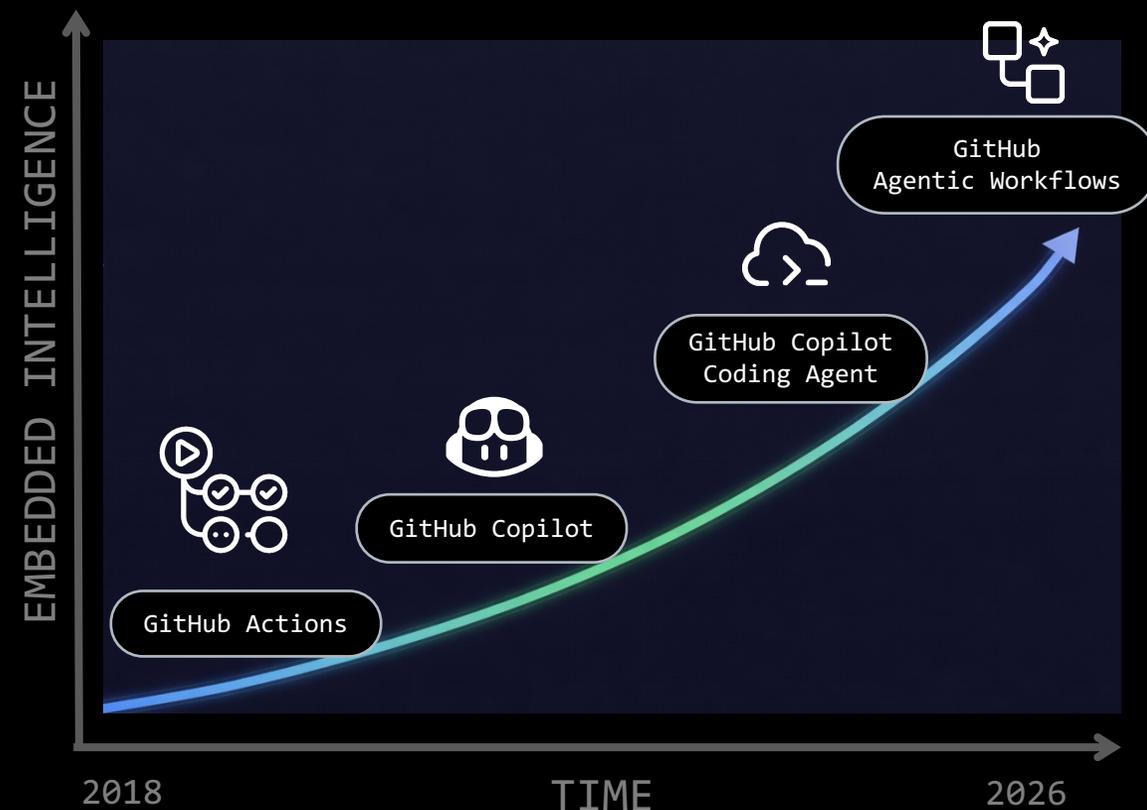




# From Automation to Intelligence

Embedding intelligence into trusted workflows:

- Automation is the foundation  
2018: Launch of GitHub Actions [🔗](#)
- Copilot assists in the editor  
2021: Launch of GitHub Copilot [🔗](#)
- Software Engineering (SWE) agents act on tasks  
2025: Launch of GitHub Copilot Coding Agent [🔗](#)
- Agentic workflows with built-in intelligence  
2026: Launch of GitHub Agentic Workflows [🔗](#)





# GitHub Agentic Workflows

Safely combine GitHub Actions & SWE agents using our Command Line Interface (CLI).

## Prerequisites:

- AI Account - Assumes GitHub Copilot by default
- GitHub Repository - A repository where you have *write* access
- GitHub Actions enabled
- GitHub CLI (gh) v2.0.0+
- Operating System: Linux, macOS, or Windows with WSL

## Setup:

Install the GitHub Agentic Workflows CLI [🔗](#)

```
gh extension install github/gh-aw
```

```
-zsh 1

GitHub Agentic Workflows

GitHub Agentic Workflows from GitHub Next

Common Tasks:
gh aw init # Set up a new repository
gh aw new my-workflow # Create your first workflow
gh aw compile # Compile all workflows
gh aw run my-workflow # Execute a workflow
gh aw logs my-workflow # View execution logs
gh aw audit <run-id> # Debug a failed run
```



# GitHub Agentic Workflows

Use frontmatter to define when a workflow runs and what it is allowed to read and write.

## Required:

Trigger configuration: `on`

## Optional:

- AI configuration: `engine`
- Execution: `jobs`, `concurrency`, `if`
- Capabilities: `tools`, `imports`, `env`
- Metadata: `name`, `description`, `label`, `source`
- Security & Access: `permissions`, `network`, `safe-outputs`
- Runtime: `runs-on`, `timeout-minutes`, `environment`

See docs for a full frontmatter reference. [🔗](#)

```
issue-clarifier.md ×
.github > workflows > issue-clarifier.md > abc # Issue Clarifier
1 ---
2 on: # Trigger: when to run
3 |   issues:
4 |     | types: [opened]
5 |
6 permissions: read-all # Security: read-only by default
7
8 safe-outputs: # Allowed write operations
9 |   add-comment:
10 ---
11 # Issue Clarifier
12
13 Analyze the current issue and ask for additional details
14 if the issue is unclear.
15
```



# GitHub Agentic Workflows

Write simple markdown to declare your automation needs:

- Write instructions in natural language.

Markdown file \*.md

- Compile to validate and generate an executable workflow.

Yaml file \*.lock.yml



```
issue-clarifier.md ×
.github > workflows > issue-clarifier.md > abc # Issue Clarifier
1 ---
2 on: # Trigger: when to run
3 | issues:
4 | | types: [opened]
5
6 permissions: read-all # Security: read-only by default
7
8 safe-outputs: # Allowed write operations
9 | add-comment:
10 ---
11 # Issue Clarifier
12
13 Analyze the current issue and ask for additional details
14 if the issue is unclear.
15
```



# GitHub Agentic Workflows

With Agentic Workflows, your CI/CD can:

- **Understand context**  
Read the repository, issues & pull requests, etc.
- **Make decisions**  
Choose what to do based on the context not just predefined conditions.
- **Stay on course**  
Understand concepts not just GitHub primitives.

Supported AI engines:

[Open](#)

mnkiefer opened 24 minutes ago

Please add better setup instructions.

Right now it's not clear how to get started, what prerequisites are needed, or what the expected workflow is.

Create sub-issue



github-actions bot 20 minutes ago – with [GitHub Actions](#)

Thanks for raising this! The request for better setup instructions is clear, but to write something accurate and useful, a few more details would help:

1. **What does this project do?** A brief description of the project's purpose would help frame the setup instructions appropriately.



# GitHub Agentic Workflows

Agentic Workflows put security first:

- **agent**

Has *read-only* permissions and only proposes actions.

- **detection**

Threat detection analyses scans the agent output for security issues.

- **safe outputs**

Run proposed actions in isolated containers.

Learn more by reading about our Security Architecture. [🔗](#)

## Security Architecture

GitHub Agentic Workflows implements a defense-in-depth security architecture that protects against untrusted Model Context Protocol (MCP) servers and compromised agents. This document provides an overview of our security model and visual diagrams of the key components.

### Security Model

Agentic Workflows (AW) adopts a layered approach that combines substrate-enforced isolation, declarative specification, and staged execution. Each layer enforces distinct security properties under different assumptions and constrains the impact of failures above it.

### Threat Model

We consider an adversary that may compromise untrusted user-level components, e.g., containers, and may cause them to behave arbitrarily within the privileges granted to them. The adversary may attempt to:



# GitHub Agentic Workflows

Agentic Workflows are the building blocks of Continuous AI:

Peli's Agent Factory [↗](#)

- **Reactive Intelligence (ChatOps · IssueOps)**  
AI responds to issues, PRs & comments to triage, etc.
- **Continuous Improvement (DailyOps · TaskOps)**  
Scheduled AI-driven analysis and incremental upgrades.
- **Insight & Governance (DataOps · ProjectOps)**  
Analyze repository data and automatically update project state.
- **Orchestration (MultiRepoOps · CentralRepoOps)**  
Coordinate work across repositories with centralized oversight.

And many more! Check out our "Design Patters". [↗](#)

## Welcome to Peli's Agent Factory

Jan 12, 2026



Don Syme



Peli de Halleux



Mara Kiefer

Welcome, welcome, WELCOME to Peli's Agent Factory!

Imagine a software repository where AI agents work alongside your team - not replacing developers, but handling the repetitive, time-consuming tasks that slow down collaboration and forward progress.

Peli's Agent Factory is our exploration of what happens when you take the design philosophy of "**let's create a new automated agentic workflow for that**" as the answer to almost every opportunity that arises! What happens when you **max out on automated agentic workflows** - when you make and use dozens of specialized, automated AI agentic workflows and use them in practice.





# GitHub Agentic Workflows

Explore our agentic collection for inspiration.

📖 README   ❤️ Code of conduct   ⚖️ MIT license   ⚖️ Security

## ✨ The Agentic

A sample family of reusable [GitHub Agentic Workflows](#).

### 📁 Available Workflows

#### Triage Workflows

- 📁 [Issue Triage](#) - Triage issues and pull requests

#### Fault Analysis Workflows

- 📁 [CI Doctor](#) - Monitor CI workflows and investigate failures automatically
- 📁 [CI Coach](#) - Optimize CI workflows for speed and cost efficiency

Agentic

### Code Review Workflows

- ✅ [Contribution Guidelines Checker](#) - Review pull requests for compliance with contribution guidelines
- 🤖 [Grumpy Reviewer](#) - On-demand opinionated code review by a grumpy but thorough senior developer

### Research, Status & Planning Workflows

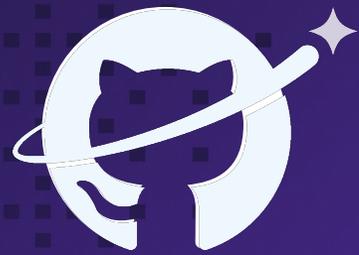
- 📖 [Weekly Research](#) - Collect research updates and industry trends
- 📊 [Weekly Issue Summary](#) - Weekly issue activity report with trend charts and recommendations
- 👥 [Daily Repo Status](#) - Assess repository activity and create status reports
- 👥 [Daily Team Status](#) - Create upbeat daily team activity summaries with productivity insights
- 📅 [Daily Plan](#) - Update planning issues for team coordination

### Dependency Management Workflows

- 📦 [Dependabot PR Bundler](#) - Create pull requests to bundle together as many dependabot updates as possible
- 📦 [Dependabot Issue Bundler](#) - Create issues that group together dependabot updates related to the same ecosystem



# Demo



**Thank you**

