

README

- [Custom GitHub Copilot Agents](#)
 - [What are Custom Agents?](#)
 - [Available Agents](#)
 - [@ArchitectureReviewer](#)
 - [@BacklogGenerator](#)
 - [@TestStrategist](#)
 - [Using Agents](#)
 - [In VS Code](#)
 - [Example Usage](#)
 - [Agent Design Principles](#)
 - [Governance](#)
 - [Creating New Agents](#)
 - [Workshop Context](#)

Custom GitHub Copilot Agents

This directory contains custom agent definitions for GitHub Copilot in VS Code. These agents are specialized AI assistants that provide consistent, role-based guidance for common development workflows.

What are Custom Agents?

Custom agents are chat participants (invoked with @agentName) that have:

- **Specific roles and responsibilities** (e.g., architecture reviewer, test strategist)
- **Defined constraints and guidelines** for consistent behavior
- **Structured output formats** tailored to their purpose
- **Team-aligned practices** encoded as reusable workflows

Available Agents

@ArchitectureReviewer

Purpose: Review code and design for architectural concerns

Use when: Evaluating structural decisions, boundaries, dependencies

Output: Structured analysis with recommendations

@BacklogGenerator

Purpose: Generate user stories and backlog items from requirements

Use when: Planning features, breaking down work

Output: Formatted user stories with acceptance criteria

@TestStrategist

Purpose: Propose test strategies and identify test scenarios

Use when: Planning test coverage, identifying edge cases

Output: Test plan with categorized scenarios

Using Agents

In VS Code

1. Open Copilot Chat
2. Click the **agent dropdown** to select an agent
3. Provide your request or context
4. Use **Agent Mode** for multi-step workflows

Example Usage

Select "Architecture Reviewer" from dropdown:
analyze the task management domain model

Select "Backlog Generator" from dropdown:
create user stories for notification feature

Select "Test Strategist" from dropdown:
propose test scenarios for order validation

Agent Design Principles

- Role-based, not task-based** - Agents represent specialists, not single actions
- Consistent output format** - Predictable structure aids automation
- Bounded scope** - Clear responsibilities prevent scope creep
- Human-in-the-loop** - Agents advise; humans decide

Governance

- Agent definitions are **version-controlled** in this repository
- Changes require **pull request review** (treat as code)
- See [Agent Governance Guide](#) for details

Creating New Agents

See [Agent Design Guide](#) for instructions on:

- Defining agent responsibilities
- Writing effective instructions
- Testing and iterating on agent behavior
- Submitting agents for team use

Workshop Context

These agents are used in **Part 2: Advanced GitHub Copilot** workshop. See:

- [Agent Catalog](#) - Full agent registry
 - [Lab 06](#) - Hands-on introduction
 - [Lab 09](#) - Build your own agent
-

Related Resources:

- [GitHub Copilot Custom Agents Documentation](#)
- [VS Code Copilot Agent API](#)