

Spec-driven AI coding: Spec-kit, BMAD, Agent OS and Kiro

4 min read · Oct 18, 2025



Tim Wang

Follow



Listen



Share



More

For AI-driven development, **Spec-kit**, **BMAD**, and **OpenSpec** are toolkits that enforce a structured, spec-driven approach to minimize the inconsistencies of pure “vibe coding”. **AgentOS**, by Builder Methods, is a framework and server that enables the management of multi-agent workflows across different AI coding tools.

Here is a comparison of Spec-kit, BMAD, OpenSpec, and AgentOS.

Spec-kit

Flexible and lightweight: Adds a structured process to your existing AI coding assistant without requiring a major workflow overhaul.

Developer-orchestrated: The human developer uses the Spec-kit CLI to guide a single AI assistant through a structured workflow.

A command-line interface (CLI) toolkit that uses templates and slash commands with any AI coding assistant.

Gated four-phase process: specify, plan, tasks, and implement.

High Flexibility: Being AI-tool agnostic, it can be used with any AI coding assistant.

Developers who want to add structure to their existing workflow for small to medium projects.

BMAD Method

Comprehensive and opinionated: Mimics a full agile development team with specialized AI agents for planning, developing, and quality assurance.

Multi-agent simulation: Specialized AI agents, such as Analyst, Architect, and QA, collaborate and hand off artifacts to each other.

An NPM package and workflow system with a CLI and web UI that integrates with various IDEs and AI models.

Two phases: **Agentic Planning** (by Analyst, PM, and Architect agents) and **Context-Engineered Development** (by Scrum Master and Dev agents).

Medium Flexibility: While tool-agnostic, its effectiveness depends on how well the underlying AI model interprets its structured files.

Large, complex projects that need robust planning, documentation, and a clear separation of development roles.

OpenSpec

Minimalist and token-efficient: Prioritizes simplicity and portability by defining specifications in plain Markdown files.

Simple and single-agent: A developer uses a few slash commands to guide a single AI assistant through a three-step process.

A lightweight CLI that uses plain Markdown files and a minimal set of slash commands.

Three steps: **Propose** a change, **Review** tasks, and **Apply** the spec.

Medium Flexibility: Supports Markdown files, making it portable, but currently has integrations with a limited number of AI agents.

Individual developers or small teams who want a simple, fast, and token-efficient method.

AgentOS (Builder Methods)

Universal agentic framework: Provides the foundational technology to orchestrate multi-agent workflows across different AI coding tools.

Flexible multi-agent: Creates and manages multiple agents that can be customized to work together through conversation and predefined workflows.

A model context protocol (MCP) server that enables communication and orchestration among agents.

Customizable workflows that capture a project's standards, tech stack, and codebase to guide AI agents.

High Flexibility: Offers a foundational framework for building custom agents and workflows beyond just software development.

Teams that need to build and orchestrate multi-agent systems and require advanced customization.

How to choose

- **For a lightweight, developer-centric approach:** If you are a developer or a small team and want to improve the quality of AI-generated code by adding more structure, **Spec-kit** is the ideal starting point. It's straightforward, uses your existing tools, and is backed by [GitHub](#).
- **For large, enterprise-scale projects:** If you need a comprehensive, highly structured, and multi-agent approach that mimics a full agile team, the **BMAD Method** provides a complete framework for robust planning, documentation, and execution.
- **For maximum simplicity and portability:** If your primary goal is speed and efficiency, and you prefer a minimalist approach using plain Markdown files, **OpenSpec** offers a very lightweight solution with minimal overhead.
- **For customized, multi-agent systems:** If you need to build and manage complex multi-agent workflows that can integrate with different AI coding tools and are not limited to a specific development methodology, **AgentOS** from Builder Methods provides the underlying server and framework.

Comparisons with Kiro

As a full Integrated Development Environment (IDE), **Kiro** is distinct from the other options, which are either toolkits or frameworks. Instead of adding structure on top of an existing tool, Kiro is a self-contained, agentic environment designed around a spec-driven development (SDD) workflow from the ground up.

Editor vs. Tooling: Kiro is a complete IDE built on the Code OSS foundation (the same as VS Code), meaning it includes all standard editor features while natively integrating its spec-driven agents. In contrast, Spec-kit, BMAD, and OpenSpec are toolkits that you use within your existing editor or command-line environment. AgentOS is a server and framework, not an editor.

- **Workflow integration:** Kiro's SDD process is deeply embedded into its core functionality. When you start a new feature, Kiro's agents automatically generate requirements, design documents, and task lists, which guide the subsequent coding and testing phases. Other tools like Spec-kit offer a structured process, but Kiro automates more of the spec generation and management.
- **Multi-agent vs. Single-agent focus:** Kiro's agents operate within the IDE to perform complex, multi-file tasks and automate repetitive actions through "hooks," such as updating tests or docs when files are saved. While BMAD also uses a multi-agent approach, Kiro's agents are integrated into the IDE itself rather than simulating an external development team. Spec-kit and OpenSpec are focused on guiding a single AI assistant.
- **Control and context management:** In Kiro, the agents are explicitly designed to reason about the codebase and the spec, giving them a high degree of context awareness. The developer maintains control by reviewing and approving the generated artifacts and code changes. In contrast, using a separate toolkit requires the developer to actively manage the context being fed to the AI.
- **Use case:** Kiro is designed for developers who want a fully integrated, opinionated, and automated agentic experience for going from concept to a production-ready feature. The other tools are better suited for developers who prefer to add a structured workflow to their existing setup, with varying degrees of flexibility and complexity.

Ai Coding

Spec Driven



Follow



Written by Tim Wang

73 followers · 13 following

Coding life @ <https://timwang.au>

No responses yet



Bgerby

What are your thoughts?


More from Tim Wang

WireGuard VPN setup—tricks for IPv6

I had struggled for a while when I tried to configure WireGuard VPN Server on my ASUS GT-AX11000 Router. According to the manual, it is a...

Jul 25, 2023  19



 In Level Up Coding by Tim Wang

Bad Code Smells: The Silent Killers of Swift Elegance

Understanding code smells is essential for maintaining clean, efficient, and manageable Swift codebases. This tech blog explains various...

Mar 27  225  3




 In Level Up Coding by Tim Wang

Mastering Swift's dynamic and @_dynamicReplacement(for:) for Dynamic Behaviour

Swift is primarily a statically dispatched language, but it does offer dynamic dispatch capabilities for times when flexibility trumps...

Mar 30  7



 In Stackademic by Tim Wang

The beauty of Swift Language—Macro

A macro in computer languages is a rule or pattern that can expand the input into output and it is a great way to reuse the code in macro...

Dec 21, 2023  76  1



See all from Tim Wang

Recommended from Medium



Vishal Mysore

BMAD: The AI Development Secret No One's Talking About (Actually... Everyone is)

Over the last few weeks, I've been experimenting with BMAD (Breakthrough Method for Agentic Development) , a new framework that claims to...

Sep 4  11




 Agen.cy

20+ Genius Ways Power Users Are Using Claude Code Right Now

Here are 10+ ways power users are using Claude Code 🧵


4d ago 🖱️ 14



 In Coding Nexus by Code Coup

OpenSpec: A Spec-Driven Workflow for AI Coding Assistants (No API Keys Needed)

I've been using AI coding tools for a while—Claude, Cursor, even Copilot. They're fast, sometimes magical... but they also love to...


 In Towards AI by Vadim Brilliantov

Most AI Agents Are Flashy Demos That Never Scale

Focus On Building The Ones That Do

6d ago 🖱 281

🔖⁺ ⋮

 In AI Software Engineer by Joe Njenga

Kimi CLI is Here — Another Hot Claude Code Alternative (Does it Level Up?)

Kimi CLI has just been released and it's going to take center stage among AI terminal coding tools.

★ 2d ago 🖱️ 27 💬 2



Steve Kaplan AI 🧑🏻💻

The BMAD Method: How I Build AI Apps 10x Faster Than Traditional Dev Teams

From Zero to SaaS in 4 Hours Using Claude Code and Strategic AI Orchestration

Aug 18 🖱️ 11 💬 1



See more recommendations