

Agile Software Development with BMad Code BMAD

AI Assist Generated Content

What is BMAD?

The **Breakthrough Method for Agile AI-Driven Development (BMAD)** is a framework that restructures software development around a collaborative "agentic pipeline" of specialized AI agents. It shifts the human developer's role from "coder" to "director" or "product owner."

Instead of a single human using one large language model (LLM) with a single prompt, BMAD uses a "team" of distinct AI personas—such as an **Analyst, Product Manager (PM), Architect, and Developer**—that work sequentially. Each agent adds a layer of detailed planning and context, passing its work to the next agent.

The primary goal of BMAD is to solve the "context loss" problem of AI development. By having AI agents build the Product Requirements Document (PRD), user stories, and technical architecture *before* any code is written, the final "developer" agent receives a perfectly detailed, machine-readable specification. This results in more complex, accurate, and robust applications built with AI, drastically reducing the human effort of iterative prompting and debugging.

Detailed Steps and Workflows for Using the BMAD Framework

The BMAD framework is not a single, rigid process but a series of "planning tracks" you choose based on your project's needs. The core workflow, however, involves "Agentic Planning."

The Core Philosophy: From Coder to Director

First, you must change your mindset. Your role is not to write code. Your role is to:

- Have the initial vision.
- Clearly define the problem.
- Review and critique the AI agents' work.
- Give "go" or "no-go" decisions at each stage.

You are guiding your AI team to a solution.

The "Team" of AI Agents (The "Who")

A BMAD implementation (like those built on GitHub or Replit) uses a "CORE" (Collaboration Optimized Reflection Engine) to manage different AI agents. Key agents include:

- **Human (You):** The project owner and final decision-maker.
- **Analyst Agent:** Your market researcher. It analyzes the problem, competitors, and user needs.
- **Product Manager (PM) Agent:** Takes the Analyst's report and creates a formal **Product Requirements Document (PRD)**, including epics, user stories, and acceptance criteria.
- **Architect Agent:** Takes the PM's PRD and designs the technical solution. It chooses the tech stack, defines the data models, outlines the system architecture, and specifies API endpoints.
- **Developer Agent:** The "coder." It receives the highly-detailed PRD and Architect's specs and writes the actual code.
- **(Specialized) Archaeologist Agent:** Used in "Brownfield" projects to analyze existing, legacy code and create documentation *before* new features are added.

The Main BMAD Workflow: Agentic Planning (For New Products)

This is the most comprehensive workflow, used for building a new product or platform from scratch.

Step 1: Initiation (Human)

You provide the initial project idea, a core problem statement, and define what is in and out of scope.

Step 2: Analysis (Analyst Agent)

1. **Action:** You run the Analyst agent.
2. **Output:** The agent provides a detailed analysis of the problem, target market, and potential pitfalls.
3. **Human Review:** You review the analysis. You might ask the agent to refine its research or "challenge" its assumptions.

Step 3: Product Definition (PM Agent)

- **Action:** You feed the Analyst's report to the PM agent and ask it to create a PRD.
- **Output:** The PM agent generates a full PRD with user stories (e.g., "As a user, I want to...") and acceptance criteria.
- **Human Review:** This is a critical step. You read the PRD as if you were the product owner. Does it miss any features? Are the user stories clear? You ask the PM agent for revisions until the PRD is perfect.

Step 4: System Architecture (Architect Agent)



1. **Action:** You feed the *approved PRD* to the Architect agent.
2. **Output:** The Architect agent generates a complete technical specification, including tech stack (e.g., React, Python, PostgreSQL), data flow diagrams, and API definitions.
3. **Human Review:** You (or a human technical lead) review the architecture. Is it scalable? Is it secure? You can ask the agent to "justify" its choice of database or "consider" an alternative framework.


Step 5: Code Generation (Developer Agent)

- **Action:** Once the PRD and Architecture are locked, you feed them *both* to the Developer agent.
- **Output:** Because the prompt is now incredibly detailed and contextualized, the agent writes the complete, structured codebase, including file structures, comments, and tests.
- **Human Review:** You review the final code, which should be 90-99% complete, and deploy it.

BMAD Planning Tracks (Workflows)

BMAD is not one-size-fits-all. You choose your workflow based on the task.

Workflow Track	When to Use It	Example Agent Flow
 Quick Flow Track	Bug fixes, small features, or rapid prototyping.	Human → Developer Agent
 BMad Method Track	The full workflow for new products, features,	Human → Analyst → PM →

	or platforms.	Architect → Developer
 Brownfield Track	Adding features to an existing, "legacy" codebase.	Human → Archaeologist Agent (to map code) → PM → Architect → Developer

By using this structured, multi-agent approach, the BMAD framework aims to create a more disciplined and scalable method for developing software with AI.

This video provides a high-level overview of [the BMAD method and its role as an AI co-pilot](#) for software development.