

BMAD-METHOD 技术规格文档

版本: 1.0

日期: 2025-09-11

1. 简介

本文件旨在提供BMAD-METHOD™ (Breakthrough Method of Agile AI-Driven Development) 的全面技术规格。BMAD-METHOD™ 是一个突破性的AI代理编排框架，其核心理念是通过专门的AI代理来模拟完整的敏捷开发团队，从而使单个开发者或小型团队也能获得类似完整团队的支持能力。该框架旨在通过结构化、规范驱动的开发流程，解决AI辅助开发中常见的规划不一致和上下文丢失问题。

BMAD-METHOD™ 的两个关键创新是：

- 代理规划 (Agentic Planning):** 专门的代理（如分析师、项目经理、架构师）与用户协作，创建详细、一致的产品需求文档 (PRD) 和架构文档。通过先进的提示工程和“人在环路”的优化，这些规划代理能够生成远超通用AI任务生成的综合性规范。
- 上下文工程开发 (Context-Engineered Development):** Scrum Master代理将详细的规划文档转化为超详细的开发故事，其中包含了开发代理所需的一切信息——完整的上下文、实现细节和架构指导，这些信息都直接嵌入到故事文件中。这种方法确保了开发代理在开始工作时，能够完全理解要构建什么、如何构建以及为什么构建。

2. 系统架构

BMAD-METHOD™ 的生态系统围绕 `bmad-core` 目录构建，该目录是整个系统的核心。`tools` 目录则提供了处理和打包核心组件以适应不同环境的工具。

2.1. 核心组件

`bmad-core` 目录包含了定义代理能力的所有资源。

2.1.1. 代理 (Agents)

- 位置:** `bmad-core/agents/`
- 描述:** 代理是系统的基本构建模块。每个 `.md` 文件定义了一个AI代理的角色、能力和依赖关系。例如，`bmad-master.md`，`pm.md`，`dev.md` 分别定义了主代理、项目经理代理和开发代理。
- 结构:** 每个代理文件都包含一个YAML头部，用于指定其角色、身份、依赖项和启动说明。依赖项是代理可以使用的任务、模板、清单和数据文件的列表。

- **启动说明:** 代理可以包含启动序列，用于从 `docs/` 文件夹加载项目特定的文档，如编码标准、API规范或项目结构文档，从而在激活时提供即时的项目上下文。
- **文档集成:** 代理可以引用和加载项目 `docs/` 文件夹中的文档，作为任务、工作流程或启动序列的一部分。用户还可以将文档直接拖到聊天界面中以提供额外的上下文。

2.1.2. 代理团队 (Agent Teams)

- **位置:** `bmad-core/agent-teams/`
- **描述:** 团队文件（例如 `team-all.yaml`）定义了为特定目的（如“全栈开发”或“仅后端”）捆绑在一起的代理和工作流程的集合。这为Web UI环境创建了一个更大、预打包的上下文。
- **结构:** 团队文件列出了要包含的代理。它可以使用通配符，例如 `"**"` 来包含所有代理。

2.1.3. 工作流程 (Workflows)

- **位置:** `bmad-core/workflows/`
- **描述:** 工作流程是YAML文件（例如 `greenfield-fullstack.yaml`），为特定项目类型定义了规定的步骤序列和代理交互。它们作为用户和 `bmad-orchestrator` 代理的战略指南。
- **结构:** 工作流程定义了复杂和简单项目的序列，列出了每个步骤涉及的代理、它们创建的工件以及从一个步骤转移到下一个步骤的条件。它通常包含一个用于可视化的Mermaid图。

2.1.4. 可重用资源 (Reusable Resources)

这些文件夹包含了由代理根据其依赖关系动态加载的模块化组件：

- **templates/** : 包含PRD、架构规范和用户故事等常见文档的Markdown模板。
- **tasks/** : 定义了执行特定、可重复操作的说明，例如“shard-doc”或“create-next-story”。
- **checklists/** : 为产品负责人（PO）或架构师等代理提供质量保证清单。
- **data/** : 包含核心知识库（`bmad-kb.md`）、技术偏好（`technical-preferences.md`）和其他关键数据文件。

3. 工作流程

BMAD-METHOD™ 的工作流程分为两个主要阶段：规划阶段和开发阶段。

3.1. 规划工作流程 (Web UI 或强大的IDE代理)

在开发开始之前，BMAD-METHOD™ 遵循一个结构化的规划工作流程，理想情况下在Web UI中完成以提高成本效益。

1. **创建文档:** 与PO（产品负责人）代理协作，创建产品需求文档 (PRD) 和架构文档。
2. **文档对齐:** PO确认PRD和架构文档之间的一致性。

3. **切换到IDE:** 将 `docs/prd.md` 和 `docs/architecture.md` 复制到项目的 `docs` 文件夹中，并在IDE中打开项目。
4. **文档分片:** 使用PO代理将PRD和架构文档分片为Epics和Stories。

3.2. 核心开发周期 (IDE)

规划完成后，BMAD-METHOD™ 遵循一个结构化的开发工作流程：

1. **起草故事:** Scrum Master (SM) 代理起草下一个开发故事。
2. **实现故事:** 开发 (Dev) 代理实现该故事。
3. **QA审查:** QA代理审查已实现的故事。
4. **循环:** 重复此过程，直到所有故事都完成。

4. 安装与配置

4.1. 系统要求

- **Node.js:** v20 或更高版本
- **npm:** v9 或更高版本
- **Git:** 已安装并配置

4.2. 安装

通过以下命令进行交互式安装：

Shell

```
npx bmad-method install
```

此命令将处理新安装、升级和扩展包的安装。

4.3. 配置

BMAD-METHOD™ 的核心配置文件是 `technical-preferences.md`，位于 `bmad-core/data/` 目录中。该文件作为一个持久的技术配置文件，影响所有项目中的代理行为。用户可以根据自己的技术栈、设计模式、编码标准等偏好来自定义此文件。

5. API 和集成

BMAD-METHOD™ 主要通过命令行界面 (CLI) 和IDE集成进行交互，而不是提供传统的RESTful API。其核心交互模型基于文件系统和AI代理的调用。

5.1. 代理交互

在IDE中，可以通过特定的语法与代理进行交互。例如，在某些IDE（如Cursor）中，使用 `@` 符号来调用代理：

Plain Text

```
@pm Create a PRD for a task management app
@architect Design the system architecture
@dev Implement the user authentication
```

在其他IDE（如Claude Code）中，可能使用斜杠命令：

Plain Text

```
/pm Create user stories
/dev Fix the login bug
```

5.2. Codex 集成

BMAD-METHOD™ 与OpenAI Codex集成，提供了两种安装模式：

- **Codex (仅本地):** `.bmad-core/` 目录被忽略，适用于本地开发。
- **Codex Web Enabled:** `.bmad-core/` 目录被跟踪，以便可以将其提交以用于Codex Web。

安装后，会在项目根目录生成 `AGENTS.md` 文件，其中包含了与Codex交互的详细说明。

5.3. 扩展包

BMAD-METHOD™ 的自然语言框架可以应用于任何领域。扩展包提供了针对特定领域的专用AI代理，例如创意写作、商业策略、健康与保健、教育等。用户还可以创建自己的扩展包来扩展核心功能。

6. 性能和安全性

6.1. 性能优化

BMAD-METHOD™ 通过以下方式优化性能：

- **依赖系统:** 每个代理只加载其需要的资源，保持精简的上下文。
- **模块化设计:** 资源在代理之间共享以保持一致性，同时避免重复加载。
- **上下文管理:** 建议在每个故事实现后压缩对话并开始新的对话，以防止上下文过载影响性能。

6.2. 安全性

BMAD-METHOD™ 的安全性主要依赖于底层AI平台（如Gemini、Claude、ChatGPT）的安全措施。该框架本身不处理敏感数据，而是专注于提供结构化的提示和工作流程。

7. 版本控制和发布

7.1. 版本管理

BMAD-METHOD™ 使用语义版本控制 (Semantic Versioning)。当前版本为 v4.43.1（截至2025年9月）。

7.2. 更新机制

用户可以通过以下命令轻松更新：

Shell

```
npx bmad-method install
# 或者
git pull
npm run install:bmad
```

更新过程会自动检测现有安装，只更新已更改的文件，并为用户的自定义修改创建 `.bak` 备份文件。

8. 故障排除和支持

8.1. 常见问题

- **图表不渲染:** 安装 Markdown All in One 和 Markdown Preview Mermaid Support 插件到 VSCode。
- **代理无响应:** 检查依赖项是否正确加载，确保所需的文件存在于相应的目录中。
- **上下文过载:** 定期压缩对话并开始新的对话以保持性能。

8.2. 社区支持

- **Discord社区:** 提供实时帮助和讨论
- **GitHub Issues:** 报告错误和功能请求
- **GitHub Discussions:** 一般讨论和问答

9. 结论

BMAD-METHOD™ 代表了AI辅助开发的重大进步，通过结构化的代理系统和工作流程，解决了传统AI开发中的关键问题。其模块化设计、扩展性和对多种IDE的支持使其成为现代软件开发团队的强大工具。随着AI技术的不断发展，BMAD-METHOD™ 将继续演进，为开发者提供更高效、更一致的开发体验。

作者: Manus AI

文档版本: 1.0

最后更新: 2025-09-11