BMad-Method BMAd Code User Guide

This guide will help you understand and effectively use the BMad Method for agile ai driven planning and development.

The BMad Plan and Execute Workflow

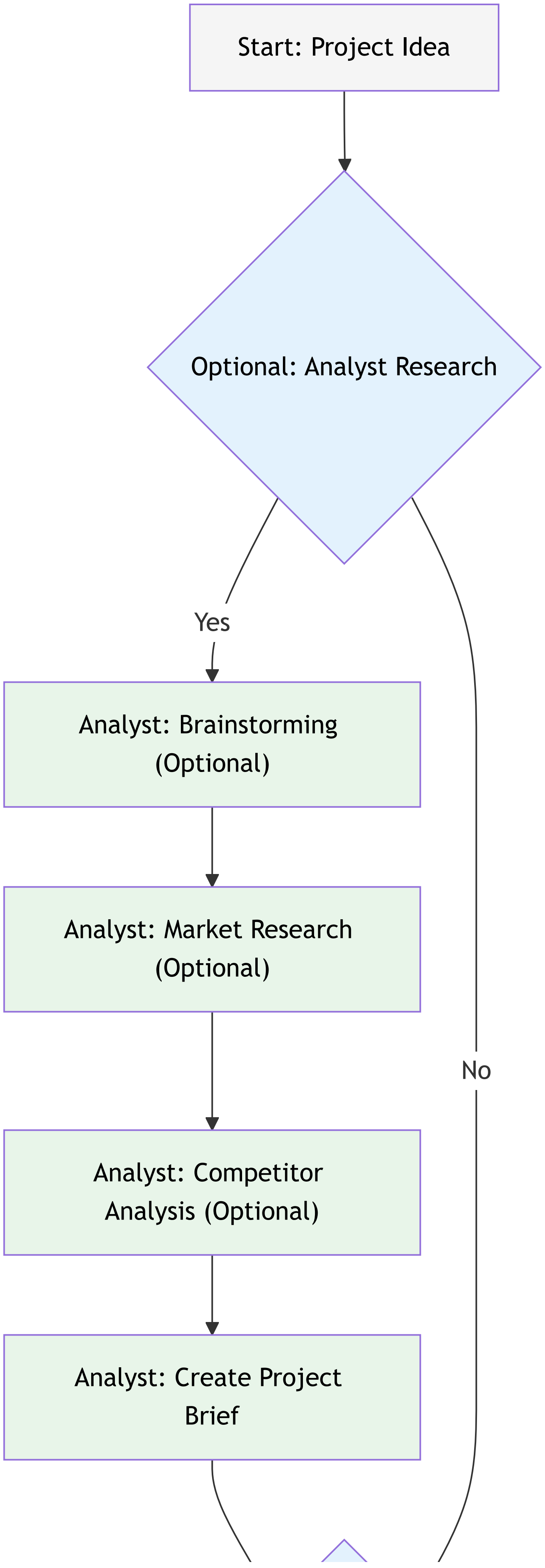
First, here is the full standard Greenfield Planning + Execution Workflow. Brownfield is very similar, but its suggested to understand this greenfield first, even if on a simple project before tackling a brownfield project. The BMad Method needs to be installed to the root of your new project folder. For the planning phase, you can optionally perform it with powerful web agents, potentially resulting in higher quality results at a fraction of the cost it would take to complete if providing your own API key or credits in some Agentic tools. For planning, powerful thinking models and larger context - along with working as a partner with the agents will net the best results.

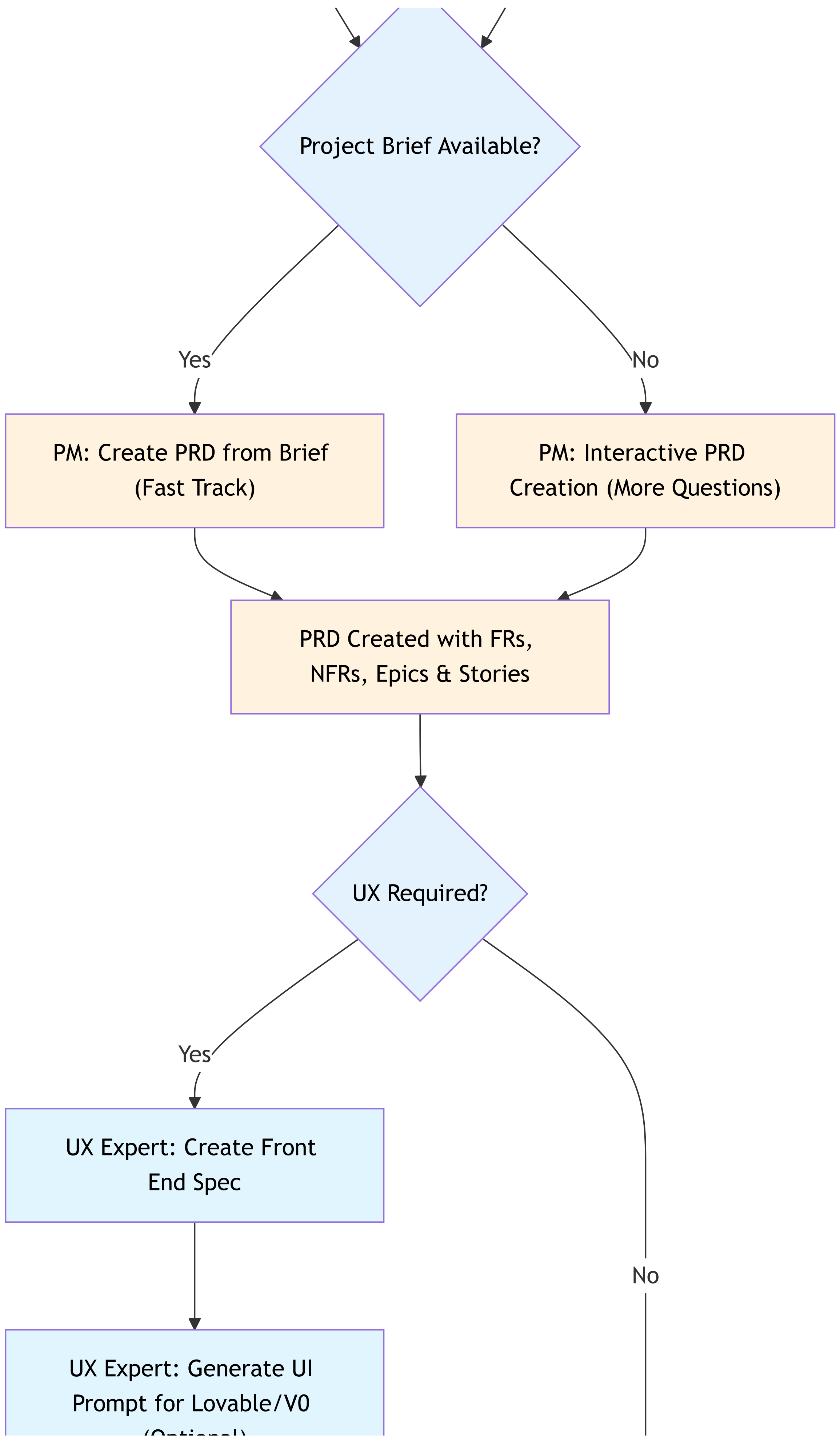
If you are going to use the BMad Method with a Brownfield project (an existing project), review Working in the Brownfield

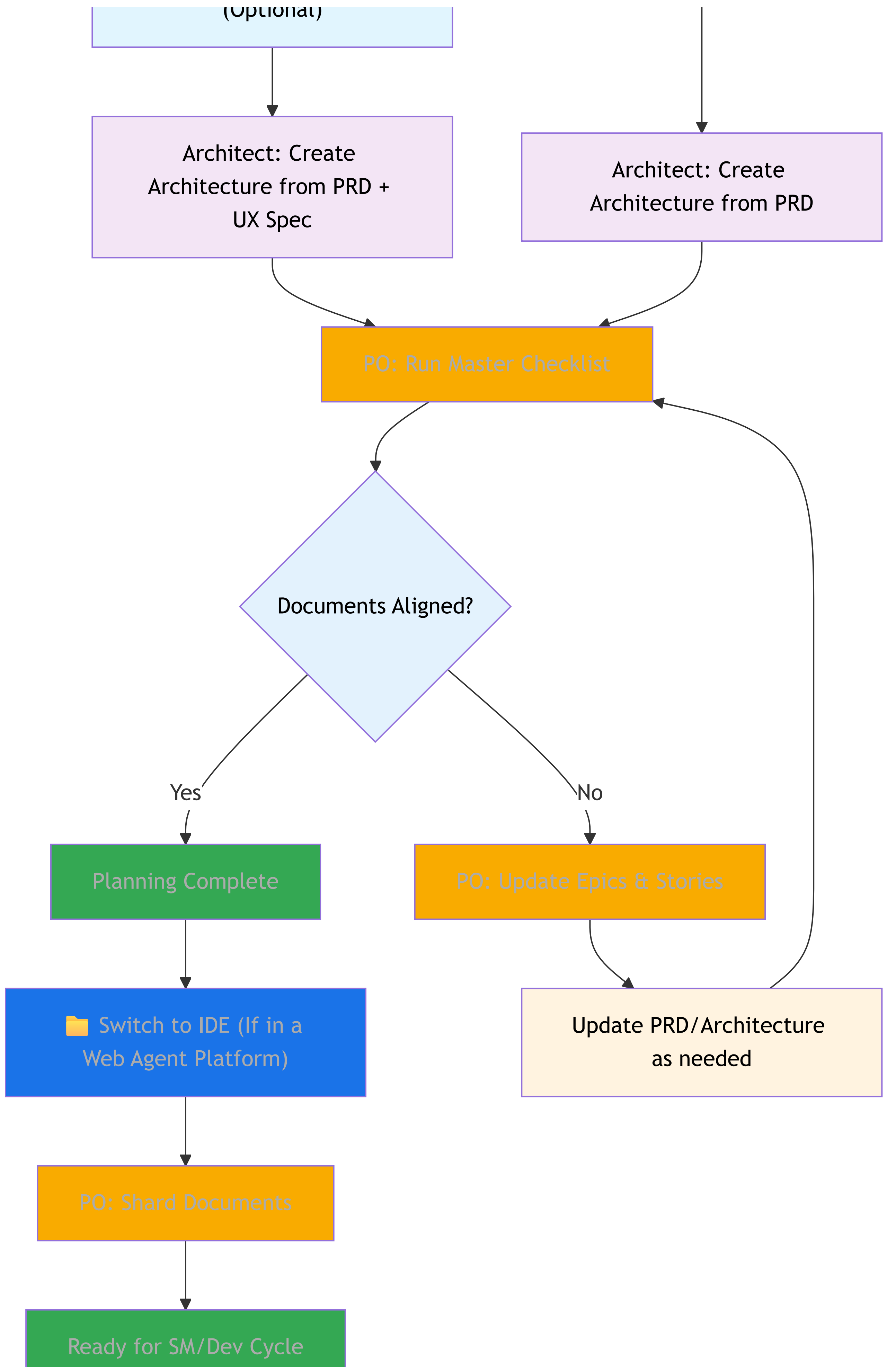
If you do not see the diagrams that following rendering, you can install Markdown All in One along with the Markdown Preview Mermaid Support plugins to VSCode (or one of the forked clones). With these plugin's, if you right click on the tab when open, there should be a Open Preview option, or check the IDE documentation.

The Planning Workflow (Web UI or Powerful IDE Agents)

Before development begins, BMad follows a structured planning workflow that's ideally done in web UI for cost efficiency:


Start: Project Idea
Optional: Analyst Research
Yes
No
Analyst: Brainstorming (Optional)
Analyst: Market Research (Optional)
Analyst: Competitor Analysis (Optional)
Analyst: Create Project Brief


Project Brief Available?
Yes
No
PM: Create PRD from Brief (Fast Track)
PM: Interactive PRD Creation (More Questions)
PRD Created with FRs, NFRs, Epics & Stories
UX Required?
Yes
No
UX Expert: Create Front End Spec
UX Expert: Generate UI Prompt for Lovable/V0 (Optional)


(Optional)
Architect: Create Architecture from PRD + UX Spec
Architect: Create Architecture from PRD
PO: Run Master Checklist
Documents Aligned?
Yes
No
Planning Complete
PO: Update Epics & Stories
Update PRD/Architecture as needed
📁 Switch to IDE (If in a Web Agent Platform)
PO: Shard Documents
Ready for SM/Dev Cycle

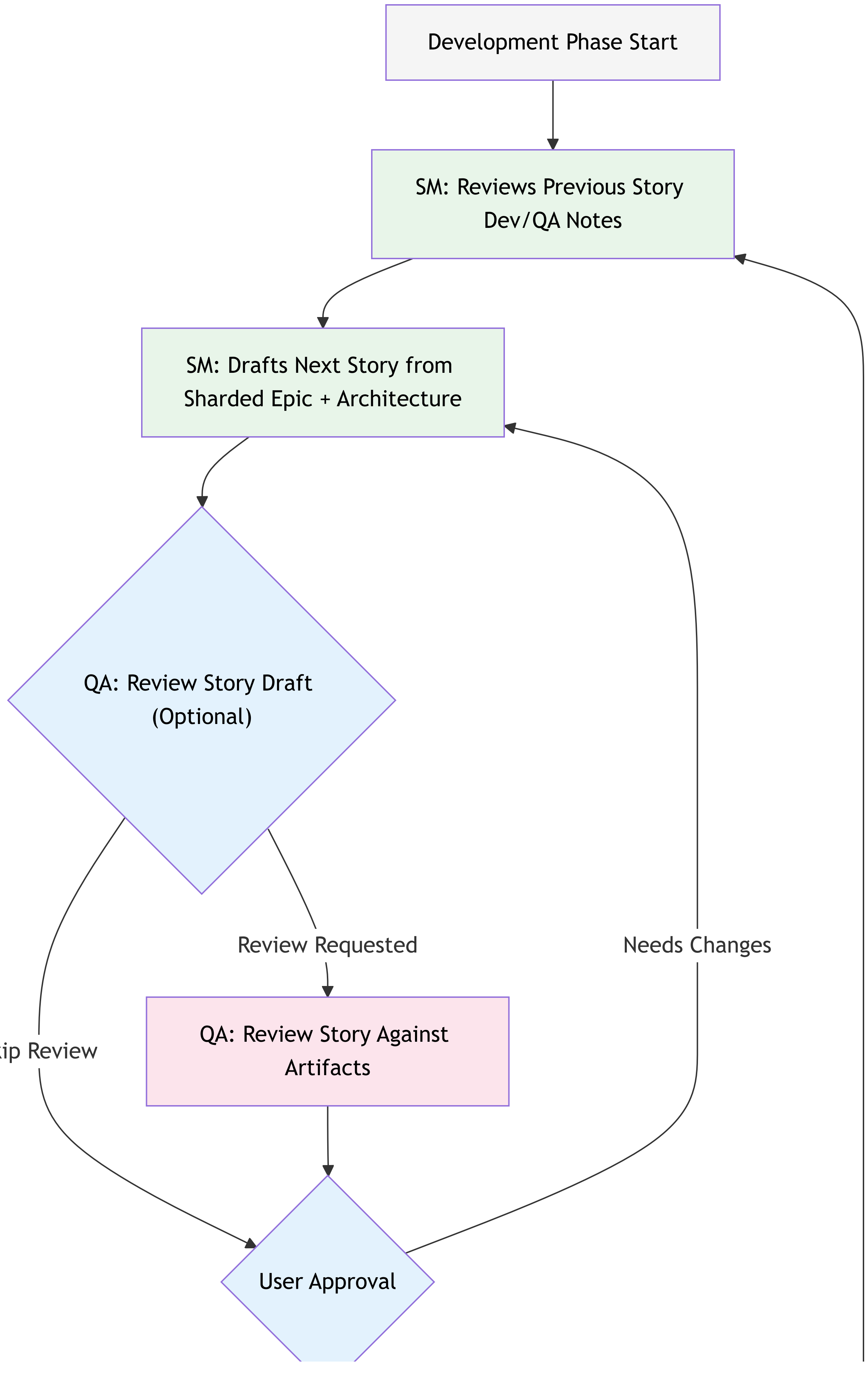
Web UI to IDE Transition

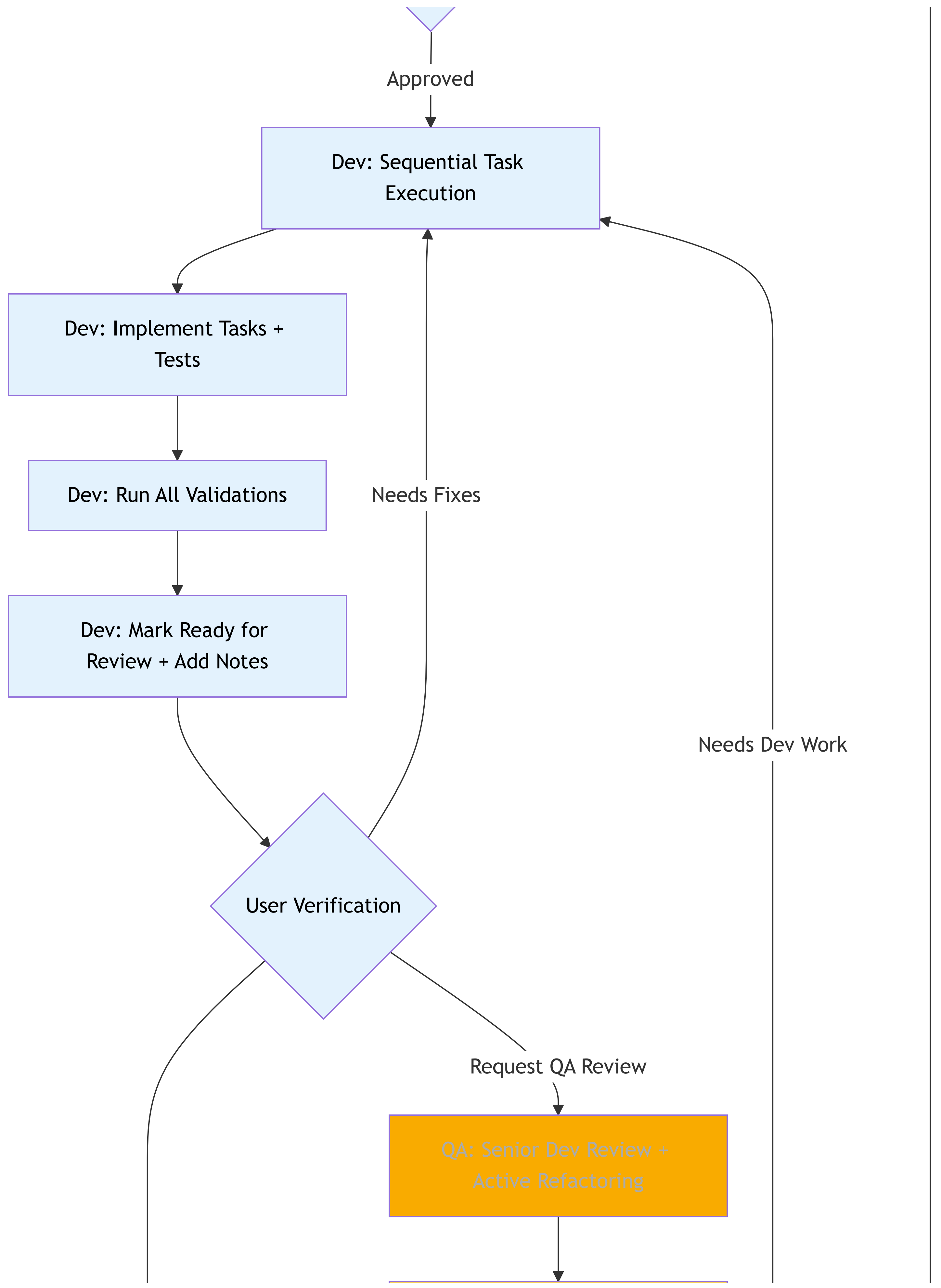
Critical Transition Point: Once the PO confirms document alignment, you must switch from web UI to IDE to begin the development workflow:

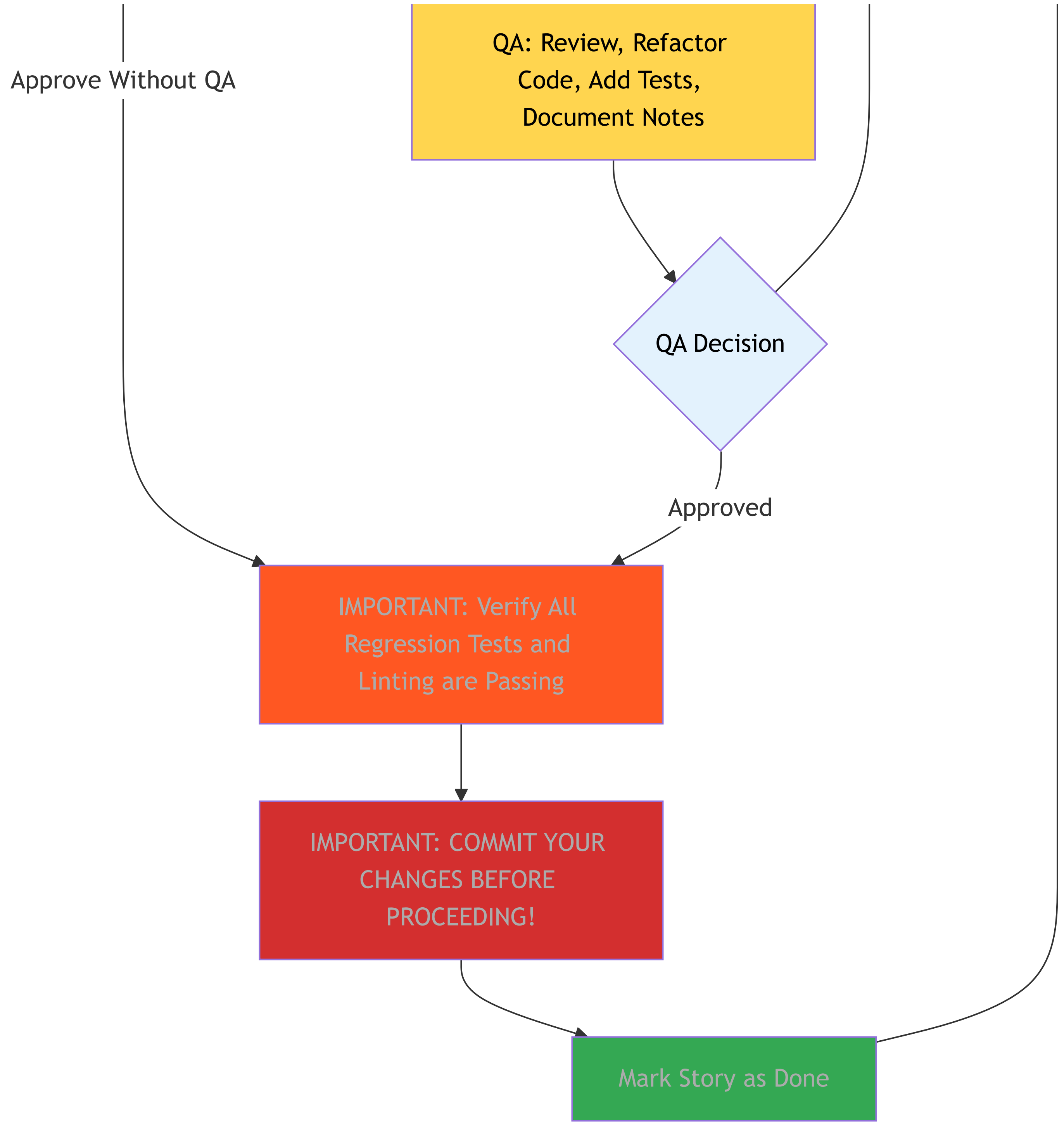
1. **Copy Documents to Project**: Ensure `docs/prd.md` and `docs/architecture.md` are in your project's docs folder (or a custom location you can specify during installation)
2. **Switch to IDE**: Open your project in your preferred Agentic IDE
3. **Document Sharding**: Use the PO agent to shard the PRD and then the Architecture
4. **Begin Development**: Start the Core Development Cycle that follows

The Core Development Cycle (IDE)

Once planning is complete and documents are sharded, BMad follows a structured development workflow:


Development Phase Start
SM: Reviews Previous Story Dev/QA Notes
SM: Drafts Next Story from Sharded Epic + Architecture
QA: Review Story Draft (Optional)
Review Requested
Skip Review
QA: Review Story Against Artifacts
Needs Changes
User Approval


Approved
Dev: Sequential Task Execution
Dev: Implement Tasks + Tests
Dev: Run All Validations
Dev: Mark Ready for Review + Add Notes
User Verification
Needs Fixes
Needs Dev Work
Request QA Review
QA: Senior Dev Review + Active Refactoring


Approve Without QA
QA: Review, Refactor Code, Add Tests, Document Notes
QA Decision
Approved
IMPORTANT: Verify All Regression Tests and Linting are Passing
IMPORTANT: COMMIT YOUR CHANGES BEFORE PROCEEDING!
Mark Story as Done

Installation

Optional

If you want to do the planning in the Web with Claude (Sonnet 4 or Opus), Gemini Gem (2.5 Pro), or Custom GPT's:

1. Navigate to `dist/teams/`
2. Copy `team-fullstack.txt` content

3. Create new Gemini Gem or CustomGPT
4. Upload file with instructions: "Your critical operating instructions are attached, do not break character as directed"
5. Type `/help` to see available commands

IDE Project Setup

```
# Interactive installation (recommended)
npx bmad-method install
```

Special Agents

There are two bmad agents - in the future they will be consolidated into the single bmad-master.

BMad-Master

This agent can do any task or command that all other agents can do, aside from actual story implementation. Additionally, this agent can help explain the BMad Method when in the web by accessing the knowledge base and explaining anything to you about the process.

If you dont want to bother switching between different agents aside from the dev, this is the agent for you.

BMad-Orchestrator

This agent should NOT be used within the IDE, it is a heavy weight special purpose agent that utilizes a lot of context and can morph into any other agent. This exists solely to facilitate the team's within the web bundles. If you use a web bundle you will be greeted by the BMad Orchestrator.

How Agents Work

Dependencies System

Each agent has a YAML section that defines its dependencies:

```yaml
dependencies:
  templates:
    - prd-template.md
    - user-story-template.md
  tasks:
    - create-doc.md
    - shard-doc.md
  data:
    - bmad-kb.md
```

Key Points:

- Agents only load resources they need (lean context)
- Dependencies are automatically resolved during bundling
- Resources are shared across agents to maintain consistency

Agent Interaction

In IDE:

```
# Some Ide's, like Cursor or Windsurf for example, utilize manual rules so interaction is done
@pm Create a PRD for a task management app
@architect Design the system architecture
@dev Implement the user authentication

# Some, like Claude Code use slash commands instead
/pm Create user stories
/dev Fix the login bug
```

Interactive Modes

- **Incremental Mode**: Step-by-step with user input
- **YOLO Mode**: Rapid generation with minimal interaction

IDE Integration

IDE Best Practices

- **Context Management**: Keep relevant files only in context, keep files as lean and focused as necessary

- **Agent Selection**: Use appropriate agent for task
- **Iterative Development**: Work in small, focused tasks
- **File Organization**: Maintain clean project structure

Technical Preferences System

BMad includes a personalization system through the `technical-preferences.md` file located in `.bmad-core/data/` - this can help bias the PM and Architect to recommend your preferences for design patterns, technology selection, or anything else you would like to put in here.

Using with Web Bundles

When creating custom web bundles or uploading to AI platforms, include your `technical-preferences.md` content to ensure agents have your preferences from the start of any conversation.

Core Configuration

The `bmad-core/core-config.yaml` file is a critical config that enables BMad to work seamlessly with differing project structures, more options will be made available in the future. Currently the most important is the devLoadAlwaysFiles list section in the yaml.

Developer Context Files

Define which files the dev agent should always load:

```yaml
devLoadAlwaysFiles:
  - docs/architecture/coding-standards.md
  - docs/architecture/tech-stack.md
  - docs/architecture/project-structure.md
```

You will want to verify from sharding your architecture that these documents exist, that they are as lean as possible, and contain exactly the information you want your dev agent to ALWAYS load into it's context. These are the rules the agent will follow.

As your project grows and the code starts to build consistent patterns, coding standards should be reduced to just the items that the agent makes mistakes at still - must with the better models, they will look at surrounding code in files and not need a rule from that file to guide them.

Getting Help

- **Discord Community**: Join Discord
- **GitHub Issues**: Report bugs
- **Documentation**: Browse docs
- **YouTube**: BMadCode Channel

Conclusion

Remember: BMad is designed to enhance your development process, not replace your expertise. Use it as a powerful tool to accelerate your projects while maintaining control over design decisions and implementation details.