

- **Testing Agent** for automated test generation, execution, and validation.
  - **Documentation Agent** for producing detailed developer and end-user documentation.
- Ensures **consistency and interoperability** across outputs from different agents.

#### 4. Automated Code Generation

- Generates **complete, modular, and maintainable codebases** with directory structures, configuration files, and environment setups.
- Produces **clean, optimized code** following standard practices (linting, formatting, naming conventions).
- Provides compatibility with **multiple programming languages and frameworks** such as Python, JavaScript (Node.js, React), and Java.

#### 5. Testing and Error Handling

- Auto-generates **unit tests, integration tests, and functional tests** for each module.
- Performs **real-time debugging and error correction** by re-running LLM agents on faulty outputs.
- Enhances software reliability by validating generated code against requirements.

#### 6. Documentation Support

- Creates **inline code documentation** for better readability and maintainability.
- Generates **project-level documentation** including API references, setup guides, and usage manuals.
- Produces **user-friendly guides** to facilitate easier onboarding of developers or stakeholders.

#### 7. Integration with Development Ecosystem

- Provides **Git integration** for automatic commit and version control.
- Supports **CI/CD integration** for continuous deployment pipelines.
- Enables **Docker-based containerization** for deployment-ready environments.