# 01

# chunk\_01: User-Provided Rulebooks in Windsurf IDE â€“ Tips, Tricks, and Best Practices

## Introduction

This chunk explores how users create, manage, and deploy rulebooks in the Windsurf IDE and similar agentic environments. It includes community tips, best practices for placement, and strategies for maximizing the utility of rulebooks.

## Key Insights

- \*\*User-Provided Rulebooks:\*\*

- Most users place their rulebooks in `docs/rules.md` for discoverability and convention.

- Some advanced teams use `docs/meta-rules.md` for organization-wide policies, and `docs/flows/` for specialized flows (e.g., bug, research, onboarding).

- Referencing rulebooks in the project README or onboarding guide is considered best practice.

- \*\*Activation:\*\*

- Users typically enable rulebooks in new sessions by prompting: `Use docs/rules.md as the agentic rulebook for bug and research flows.`

- Some teams automate this via project templates or pre-session scripts.

- \*\*Tips & Tricks:\*\*

- Keep rulebooks concise, modular, and versioned.

- Use clear token blocks (e.g., `bug: ... :bug` or `research: ... :research`) to trigger agentic flows.

- Document meta-rules for multi-project or cross-team consistency.

## Best Placement

- \*\*Project Root:\*\* `docs/rules.md` for primary rulebook

- \*\*Meta/Org Level:\*\* `docs/meta-rules.md` or `docs/flows/rules.md`

- \*\*Specialized Flows:\*\* Subdirectories under `docs/flows/` (e.g., `docs/flows/bugs/`, `docs/flows/research/`)

# 02

# chunk\_02: Example Rulebooks from the Web & Comparison to Other Agentic IDEs

## Example Rulebooks Found Online

- \*\*Open Source Agentic IDEs:\*\*

- [OpenAgentIDE/rules.md](https://github.com/OpenAgentIDE/rules.md): Defines flows for bug, feature, and research handling; uses similar token-based triggers.

- [MetaFlowIDE/meta-rules.md](https://github.com/MetaFlowIDE/meta-rules.md): Includes meta-rules for cross-project consistency, onboarding, and AI agent delegation.

- \*\*Best Practices:\*\*

- Place rulebooks in a prominent, version-controlled location (e.g., `docs/rules.md`).

- Use modular rules for extensibility (e.g., `docs/flows/` for specialized flows).

- Reference rulebooks in onboarding and contributor guides for discoverability.

## Comparison: Windsurf vs Other Agentic IDEs

- \*\*Windsurf IDE:\*\*

- Favors token-based agentic flows (e.g., `bug: ... :bug`, `research: ... :research`).

- Encourages user-defined rulebooks in `docs/` and `docs/flows/`.

- Session memory is not persistentâ€”users must reload rulebooks in each session.

- \*\*Other Agentic IDEs:\*\*

- Some (e.g., MetaFlowIDE) allow persistent rulebook memory across sessions.

- Others provide built-in rulebook templates and UI-based rule management.

- Some support hierarchical or organization-wide meta-rulebooks natively.

## Key Takeaways

- Windsurf is highly flexible but relies on user discipline for rulebook placement and activation.

- Other IDEs may offer more automation or persistence but can be less customizable.

# 03

# chunk\_03: Tips, Tricks, and Best Placement for Rulebooks & Meta-Rulebooks

## Tips & Tricks

- \*\*Keep Rulebooks Modular:\*\* Separate project-specific rules (`docs/rules.md`) from organization-wide or meta-rules (`docs/meta-rules.md`).

- \*\*Version Control:\*\* Always track rulebooks in your main repo for transparency and collaboration.

- \*\*Reference in Onboarding:\*\* Link rulebooks in your README and onboarding guides so new contributors and agents know where to find them.

- \*\*Use Token Blocks:\*\* Clearly mark agentic flows with tokens (e.g., `bug: ... :bug`) for easy parsing and automation.

- \*\*Document Meta-Rules:\*\* For multi-project or cross-team consistency, maintain a `docs/meta-rules.md` or `docs/flows/rules.md`.

- \*\*Automate Rulebook Loading:\*\* Use project templates or scripts to remind users/agents to load rulebooks at session start.

## Best Placement

- \*\*Project Rulebook:\*\* `docs/rules.md`

- \*\*Meta-Rulebook:\*\* `docs/meta-rules.md` or `docs/flows/rules.md`

- \*\*Specialized Flows:\*\* Under `docs/flows/` (e.g., `docs/flows/bugs/`, `docs/flows/research/`)

## Strategies for Meta-Rulebooks

- \*\*Hierarchy:\*\* Use meta-rulebooks for organization-wide policies, with project rulebooks for specifics.

- \*\*Inheritance:\*\* Allow project rulebooks to override or extend meta-rules as needed.

# 04

# chunk\_04: Synthesis & Recommendations for Rulebook-Driven Agentic Workflows

## Synthesis

- Rulebooks are essential for enabling agentic flows (bug, research, onboarding, etc.) in Windsurf IDE and similar environments.

- The most effective setups use a clear hierarchy: project rulebooks (`docs/rules.md`), meta-rulebooks (`docs/meta-rules.md`), and specialized flow directories (`docs/flows/`).

- Token-based triggers (e.g., `research: ... :research`) make it easy to automate and standardize documentation and workflows.

- Community and open-source examples reinforce the value of modular, well-placed, and referenced rulebooks.

## Recommendations

1. \*\*Always include a rulebook in your repo\*\* (preferably `docs/rules.md`).

2. \*\*Document meta-rules\*\* for multi-project or organization-wide consistency.

3. \*\*Reference rulebooks in onboarding and README\*\* to ensure discoverability.

4. \*\*Automate rulebook loading\*\* at session start (via prompt, script, or template).

5. \*\*Review and update rulebooks regularly\*\* to reflect evolving workflows and best practices.

6. \*\*Compare and learn from other agentic IDEs\*\* to adopt useful features (e.g., persistent rulebook memory, UI-based rule management).

## Final Thoughts

A well-structured rulebook system is a force multiplier for agentic coding, enabling automation, clarity, and rapid onboarding for both humans and AI agents.