

Junie: Your AI Coding Agent in JetBrains IDEs





Autonomous Coding with Safety Controls

Transform Your Development Workflow
with AI-Powered Automation





Four-Hour Hands-On Workshop

What You'll Master Today

Core Concepts

-  **Ask Mode:** Analysis & insights
-  **Code Mode:** Automated changes
-  **Guidelines:** Team conventions
-  **Safety:** Approvals & controls

Hands-On Labs

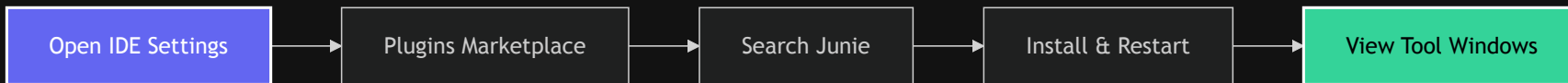
-  **Java:** Spring Boot APIs
-  **Python:** Refactoring & testing
-  **React:** Forms & validation
-  **MCP:** External tool integration

Prerequisites Check

- ✓ **JetBrains IDE** (IntelliJ IDEA, PyCharm, or WebStorm)
- ✓ **JetBrains AI Pro** subscription
- ✓ **Development Environment** (Java 17+ / Python 3.8+ / Node 16+)

💡 **Quick Setup:** We'll install Junie together in the first 5 minutes

Installation in 60 Seconds



Available Models:



GPT-5 (default)



Claude Sonnet 3.7

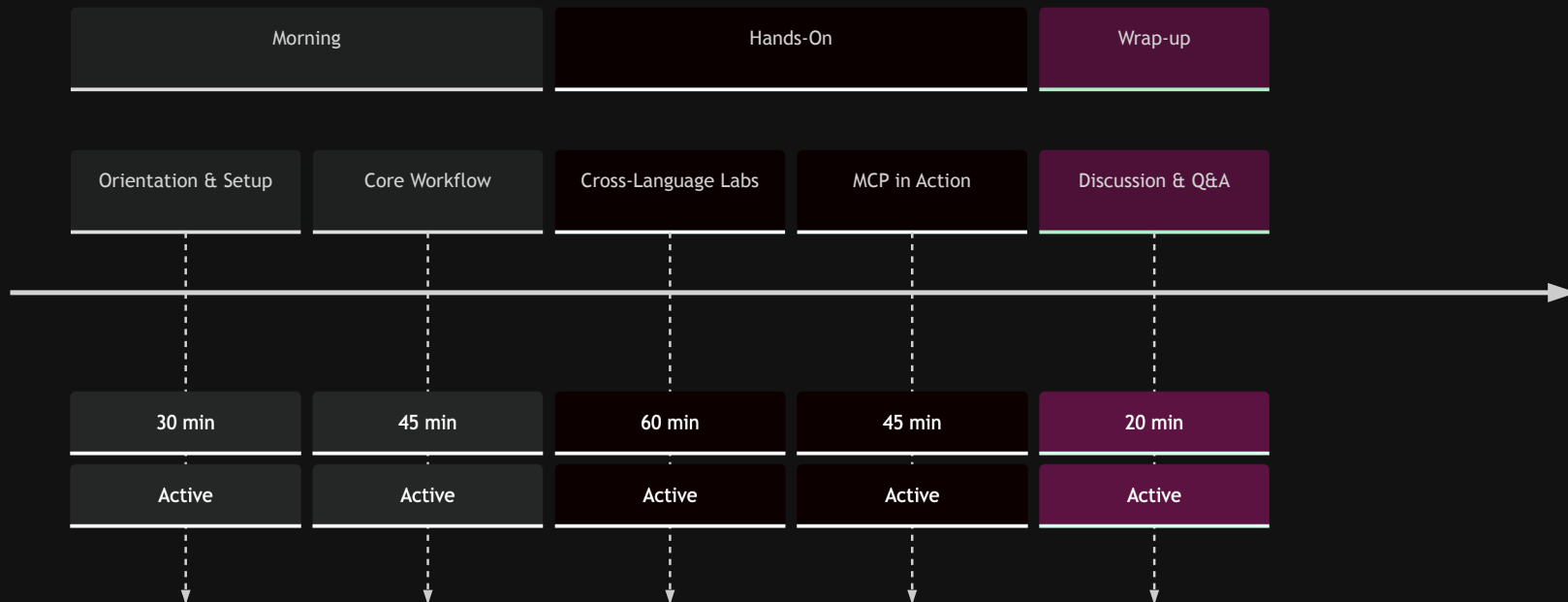


Claude Sonnet 4.0



Today's Journey

Four-Hour Workshop Timeline

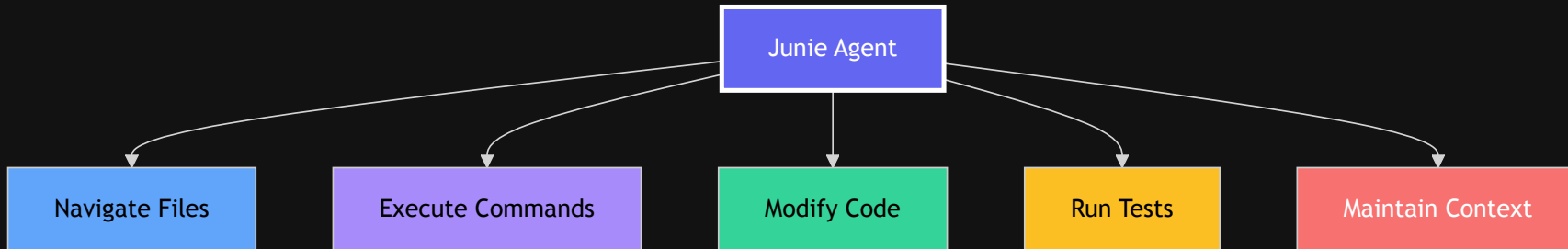


Break after Core Workflow segment



What is Junie?

JetBrains' AI-Powered Coding Agent





Two Operating Modes



Ask Mode

Read-Only Analysis

- Explain complex code
- Analyze architecture
- Find bugs & issues
- Review security
- Check test coverage

Perfect for understanding before changing



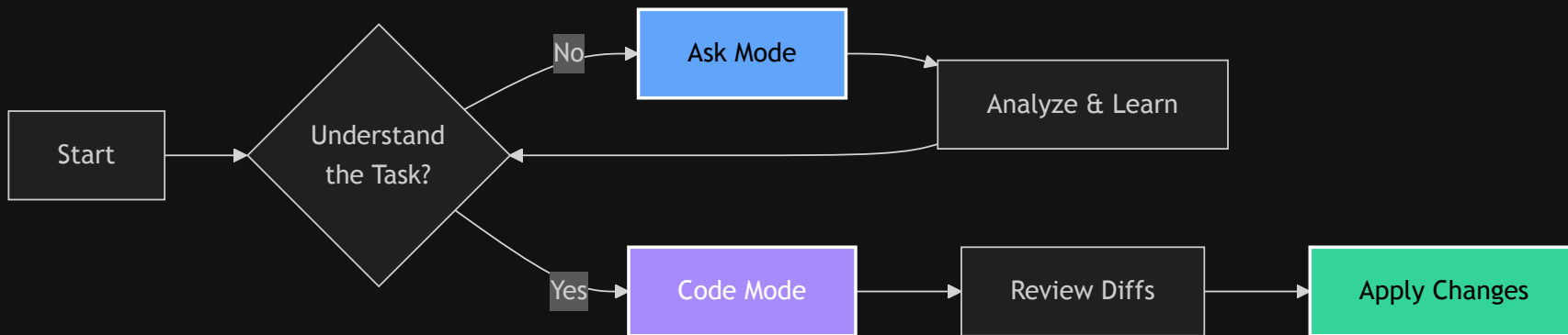
Code Mode

Make Changes

- Implement features
- Fix bugs
- Refactor code
- Generate tests
- Update dependencies

Executes multi-step plans with diffs

💡 Pro Workflow



✨ Always start with Ask mode for complex tasks



Project Guidelines

Location: `.junie/guidelines.md`

```
## Technology Stack
- Framework: Spring Boot 3.2
- Testing: JUnit 5 + AssertJ

## Conventions
- REST: /api/v1/{resource}
- DTOs: Java records
- Tests: Given-When-Then
```

✗ Without Guidelines

Inconsistent patterns

Multiple review cycles

Style conflicts

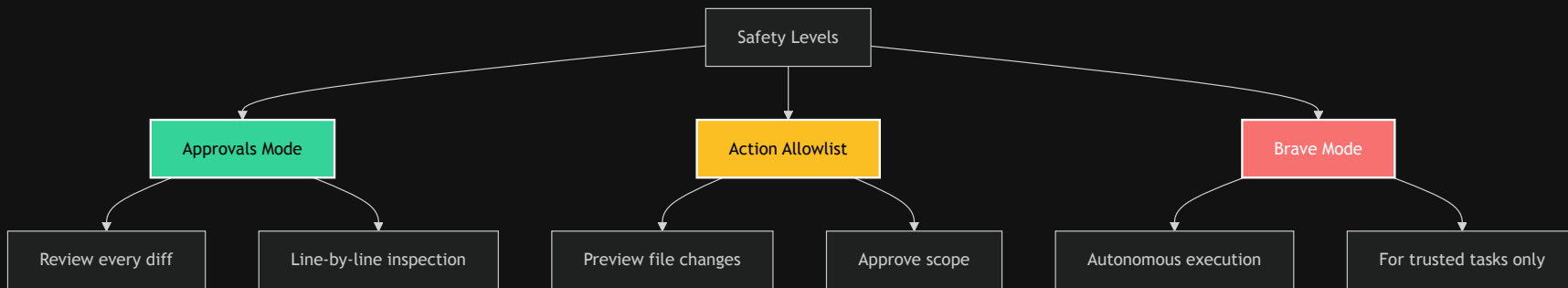
✓ With Guidelines

Consistent output

70% fewer reviews

Team alignment

Safety Controls



⚡ When to Use Each Mode

Mode	Good For	Not For
Approvals	<ul style="list-style-type: none">• Critical business logic• First-time tasks• Learning Junie	<ul style="list-style-type: none">• Repetitive formatting• Simple refactors
Allowlist	<ul style="list-style-type: none">• Known file sets• Defined scope• Team reviews	<ul style="list-style-type: none">• Exploratory changes• Unknown impact
Brave	<ul style="list-style-type: none">• Test generation• Formatting• Documentation	<ul style="list-style-type: none">• Production code• Database changes• First attempts

MCP: Model Context Protocol

Available Tools

context7

- Real-time library docs
- 1000+ libraries
- Version compatibility
- No API key needed!

Playwright

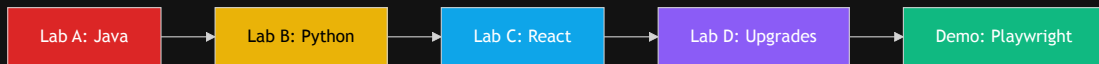
- E2E test generation
- Browser automation
- Accessibility-first
- TypeScript output

Configuration

```
{
  "mcpServers": {
    "context7": {
      "command": "npx",
      "args": ["@upstash/context7"]
    },
    "playwright": {
      "command": "npx",
      "args": ["@playwright/mcp-server"]
    }
  }
}
```

Settings → Junie → MCP → Edit mcp.json

Lab Overview



Timing:

- Lab A: 45-60 min
- Lab B: 30-45 min
- Lab C: 30-45 min
- Lab D: 20-30 min
- Demo: 20-30 min



Java

Spring Boot
REST APIs
JUnit + AssertJ



Python

PEP 8
Type hints
pytest



React

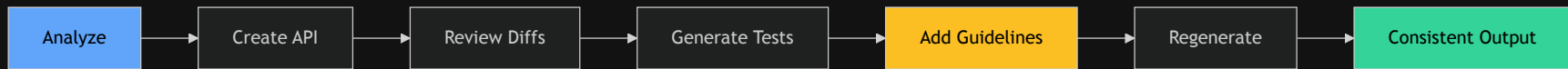
TypeScript
Forms
Testing Library



MCP

context7
Upgrades
Migration

Lab A: Spring Boot Journey



Without Guidelines

- Inconsistent patterns
- Various styles
- More review cycles

With Guidelines

- Consistent code
- Team standards
- Faster approval

Lab B: Python Transformation

Before 🤖

```
def calc(x,y,op):  
    if op=="add": return x+y  
    elif op=="sub": return x-y
```

- No type hints
- Poor naming
- No tests
- No docs

After ✨

```
def calculate(  
    x: float,  
    y: float,  
    operation: str  
) -> Optional[float]:  
    """Calculate result.  
  
    Args:  
        x: First number  
        y: Second number  
        operation: Operation  
  
    Returns:  
        Calculation result  
    """
```



Lab C: React Forms

```
interface RegistrationForm {  
  email: string;    // valid email  
  password: string; // min 8, special char  
  confirm: string;  // must match  
  terms: boolean;   // required  
}
```

React Hook Form
Form state

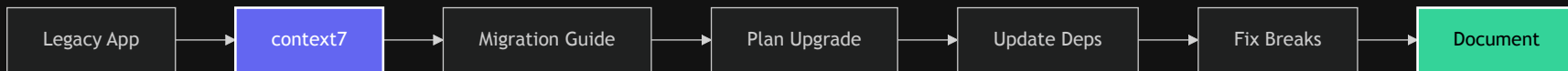
Zod
Validation

Testing Library
90% coverage

Accessibility
ARIA compliant



Lab D: Smart Upgrades with context7



React 17

Legacy



React 18

Modern

Demo: Playwright Magic

Generate E2E Tests in 5 Minutes

```
test('successful login', async ({ page }) => {  
  const loginPage = new LoginPage(page);  
  await loginPage.goto();  
  await loginPage.login('user@example.com', 'pass');  
  
  await expect(page).toHaveURL('/dashboard');  
  await expect(page.getByRole('heading', { name: 'Welcome' })))  
    .toBeVisible();  
});
```

✓ Page Objects

✓ Accessibility

✓ TypeScript

✓ Retry Logic

⚔ Junie vs The Competition

Feature	Junie	Copilot	Cursor	Codeium
Multi-file edits	✅ Full	❌	✅	⚠
Test execution	✅ Yes	❌	❌	❌
Rollback	✅ Yes	❌	⚠	❌
MCP tools	✅ Yes	❌	❌	❌
IDE native	✅ Full	⚠	❌	⚠

Common Patterns

TDD Flow

1. Ask: "What tests?"
2. Code: "Generate tests"
3. Code: "Implement"

Red → Green → Refactor

Safe Refactoring

1. Generate tests
2. Identify issues
3. Refactor safely

Tests stay green

Upgrades

1. context7 check
2. Update deps
3. Fix breaks

Systematic migration



CI/CD Integration

```
name: Junie Guidelines Check
on: [pull_request]

jobs:
  guidelines-compliance:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Check guidelines
        run: |
          # Validate against .junie/guidelines.md
          # Ensure consistency
```

Pre-commit
Local validation

CI Pipeline
Automated checks

PR Review
Team alignment

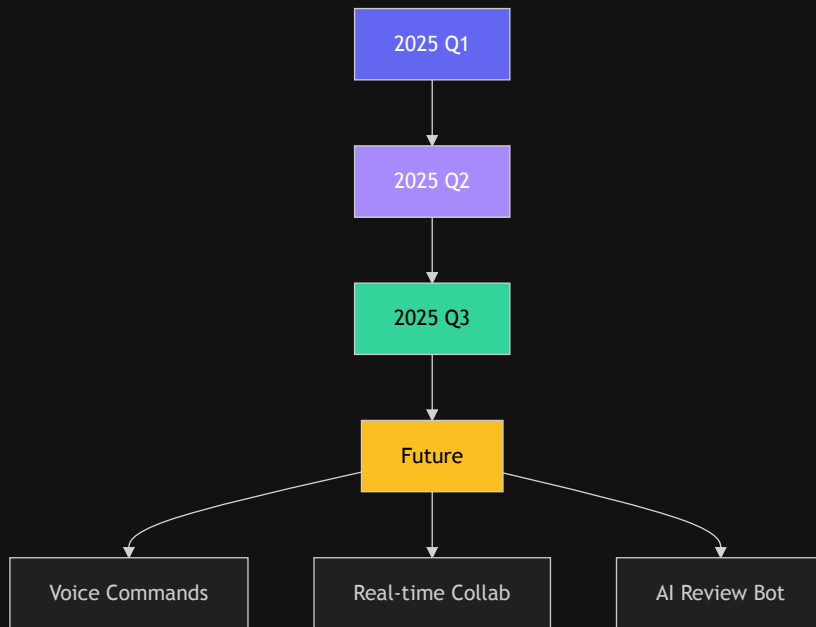
Future Roadmap

Q1 2025

- 🎯 GPT-5 as default
- 🔄 Multi-repo support
- 💻 VS Code beta

Q2 2025

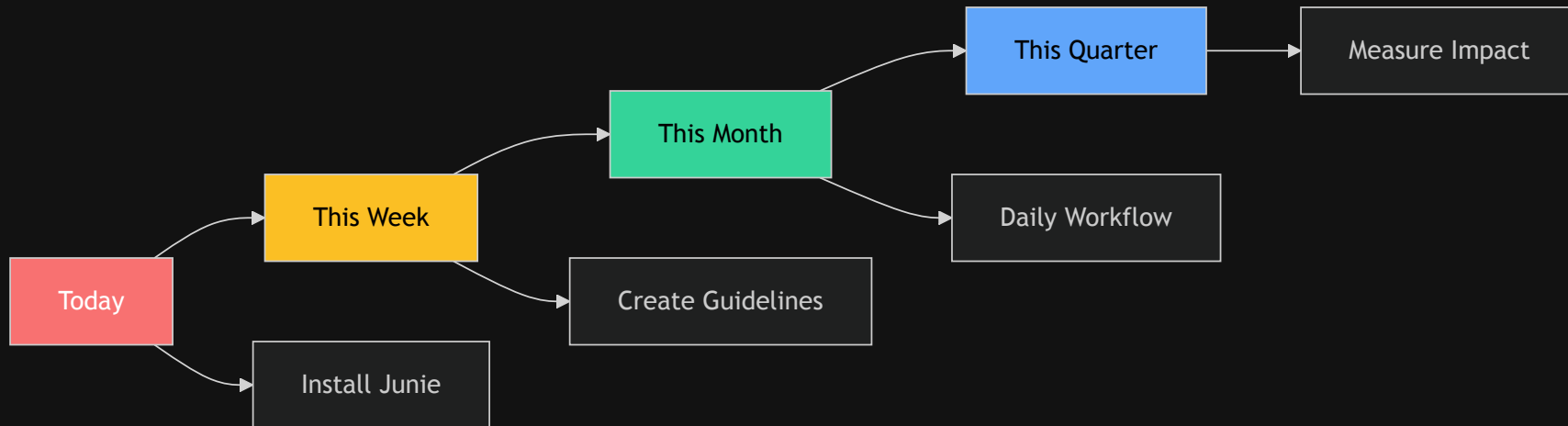
- 🐙 GitHub PR reviews
- 🛍️ MCP marketplace
- 👥 Team guidelines



✨ Key Takeaways

- ✓ **Ask before Code** - Understand first, implement second
- ✓ **Guidelines drive consistency** - Encode your team's DNA
- ✓ **Build trust gradually** - Approvals → Allowlist → Brave
- ✓ **MCP extends capabilities** - Leverage external tools
- ✓ **Safety first** - Review diffs, maintain control

Your Next Steps



 Start with Approvals Mode → Build Trust → Scale Up

Discussion Time

Let's Explore Together:

 What would you trust Junie to do unattended?

 Where do you always want human review?




 How could guidelines help your team?

 Which MCP tools fit your workflow?



 How will AI agents change development?

Resources Hub




Documentation

-  [Getting Started](#)
-  [Guidelines Guide](#)
-  [MCP Settings](#)




Tutorials

-  [IntelliJ Playbook](#)
-  [PyCharm Playbook](#)

GitHub

-  [Guidelines Catalog](#)
-  [Context7 MCP](#)
-  [This repository](#)

Community

-  [JetBrains AI Slack](#)
-  [Stack Overflow: `jetbrains-junie`](#)
-  [Support team](#)

 Thank You!

AI amplifies good practices

 Remember:

**Invest in your guidelines today,
reap consistency forever!**


Questions? Let's explore together! 

About Ken Kousen


Ken Kousen

President, Kousen IT, Inc.

 ken.kousen@kousenit.com

 github.com/kousen

 [@talesfromthejarside](https://twitter.com/talesfromthejarside)

 kousenit.substack.com

 linkedin.com/in/kenkousen

 bsky.app/profile/kousenit.com

AI agents aren't here to replace developers—
they're here to make us **better** developers.

Bonus: Quick Reference

Keyboard Shortcuts

- `Ctrl+Alt+J` - Open Junie
- `Ctrl+Enter` - Execute
- `Esc` - Cancel operation
- `Ctrl+Z` - Undo changes

Common Commands

"Analyze this file"

"Add tests with 90% coverage"

"Refactor to best practices"

"Fix all type errors"

MCP Tools

```
{  
  "context7": "Library docs",  
  "playwright": "E2E tests",  
  "search": "Web search",  
  "custom": "Your tools"  
}
```

 Print this slide for your desk!