# AI tools for software teams: The state of play

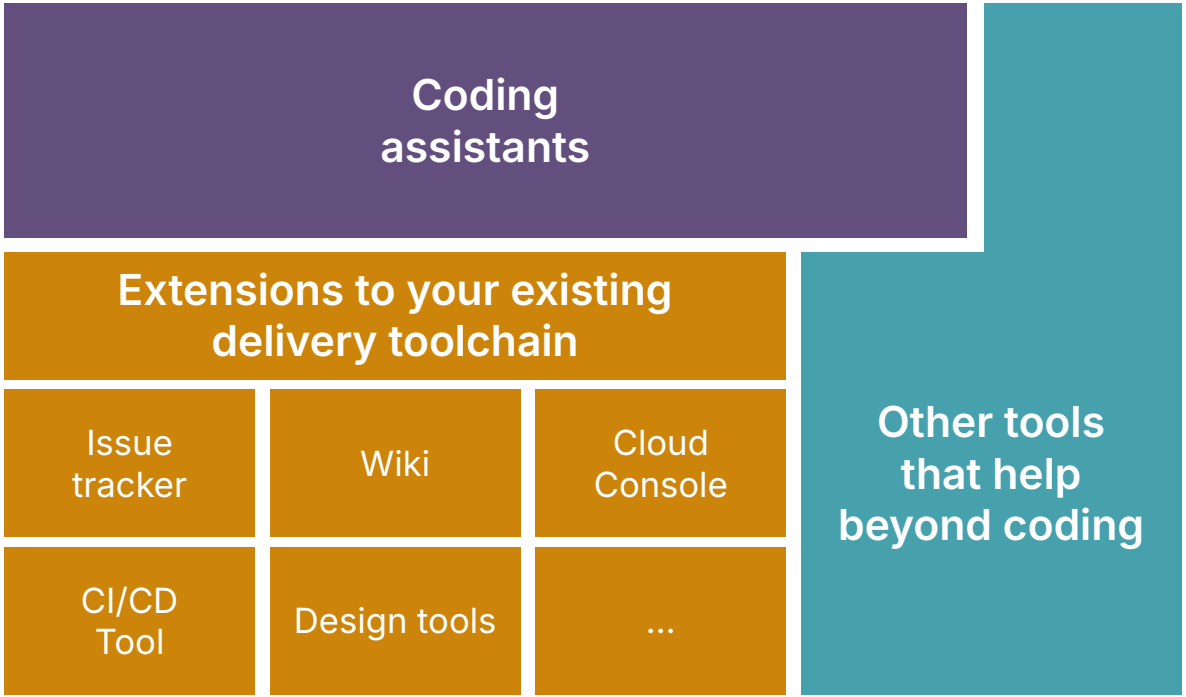
October 2024

Birgitta Böckeler
birgitta.info
@bboeckel@toot.thoughtworks.com

/thoughtworks

© 2024 Thoughtworks

# Big picture: AI tooling archetypes



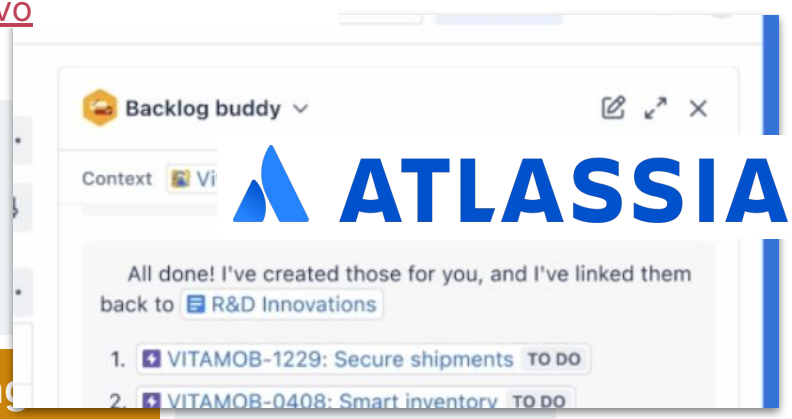

© 2024 Thoughtworks

# Adjusted for size: Tooling maturity and current adoption




© 2024 Thoughtworks

https://www.atlassian.com/software/rovo




ATLASSIAN

**Coding assistants**

**Extensions to your existing delivery toolchain**

| | | |
|---|---|---|
| Issue tracker | Wiki | Cloud Console |
| CI/CD Tool | Design tools | ... |

**Other tools that help beyond coding**

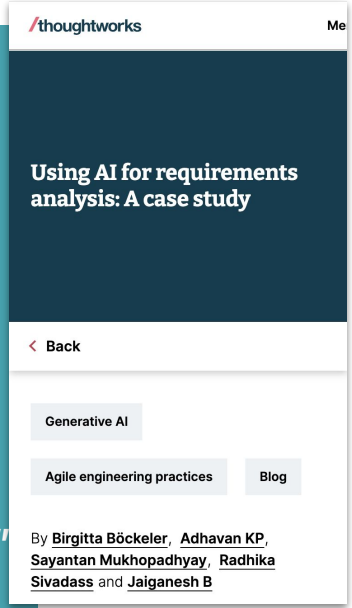
© 2024 Thoughtworks

**Coding assistants**

**Extensions to your existing delivery toolchain**

| | | |
|---|---|---|
| Issue tracker | Wiki | Cloud Console |
| CI/CD Tool | Design tools | ... |

**Other tools that help "beyond coding"**




https://www.thoughtworks.com/insights/blog/generative-ai/using-ai-requirements-analysis-case-study

© 2024 Thoughtworks

# Coding assistants: A tour



© 2024 Thoughtworks

# Inline suggestions

```
// calculate the median of a list of numbers
public static median(values:number[]): number {
```

```
// calculate the median of a list of numbers
public static median(values:number[]): number {
    if(values.length === 0) return 0;

    values.sort((a,b) => a-b);

    var half = Math.floor(values.length / 2);

    if (values.length % 2)
        return values[half];

    return (values[half - 1] + values[half]) / 2.0;
}
```

# Inline suggestions

```
class PromptList():

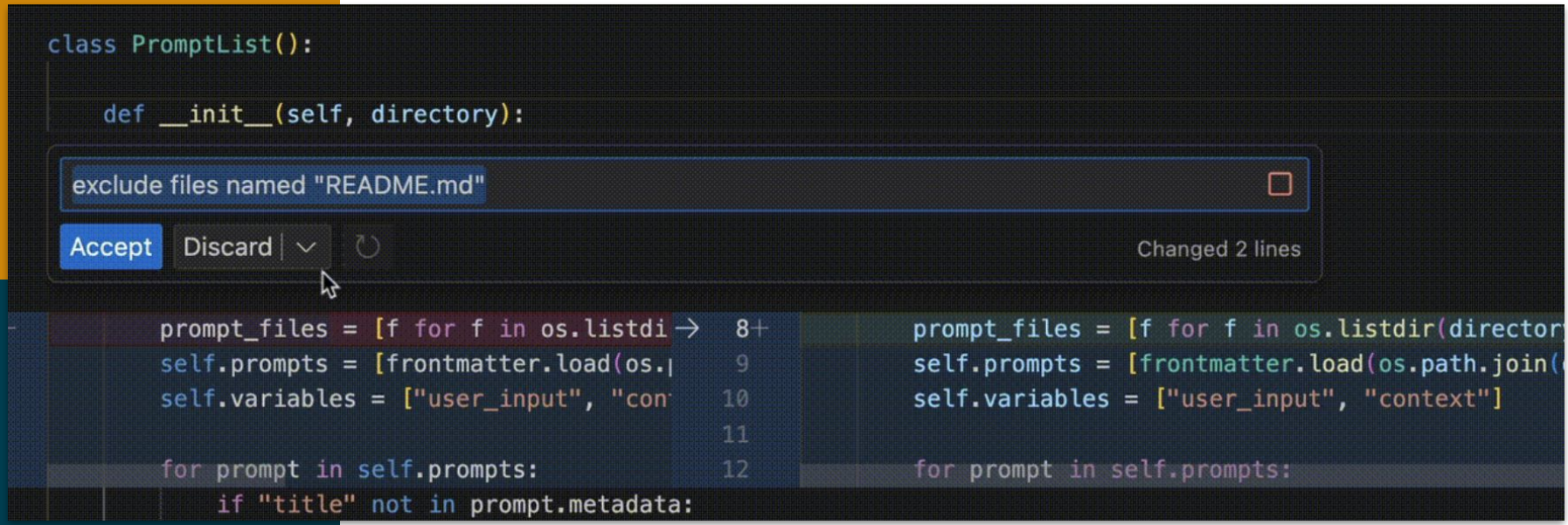
    def __init__(self, directory):
        prompt_files = [f for f in os.listdir(directory) if f.endswith('.md')]
        self.prompts = [frontmatter.load(os.path.join(directory, filename)) for filename in prompt_files]
        self.variables = ["user_input", "context"]

        for prompt in self.prompts:
            if "title" not in prompt.metadata:
                prompt.metadata["title"] = "Unnamed use case"
            if "context" not in prompt.metadata:
                prompt.metadata["context"] = "None"
            if "system" not in prompt.metadata:
                prompt.metadata["system"] = "You are a useful assistant"

    def get_titles(self):
        return [prompt.metadata.get("title", "Unnamed use case") for prompt in self.prompts]
```

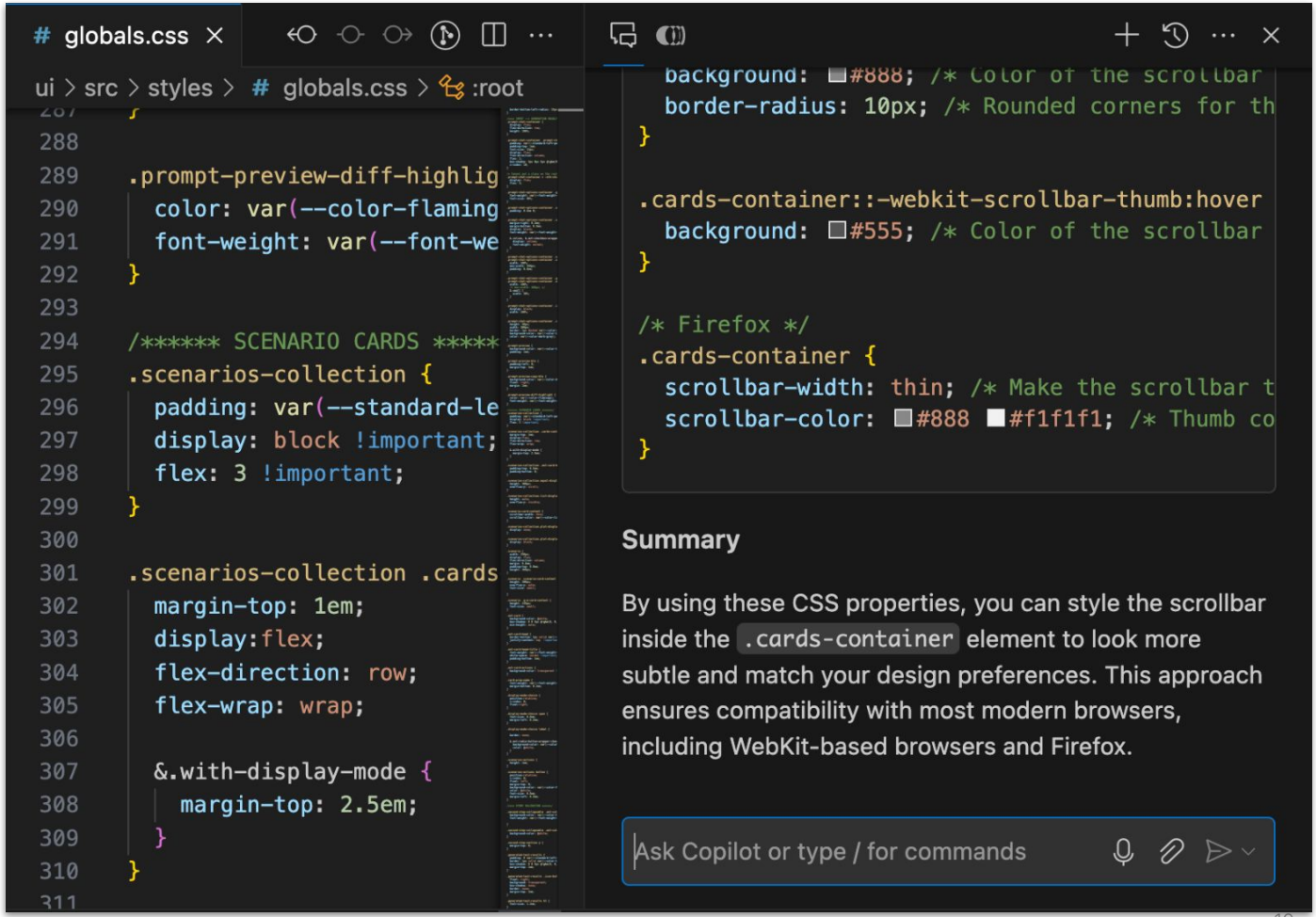
# Integrated chat

# Inline suggestions




# Integrated chat

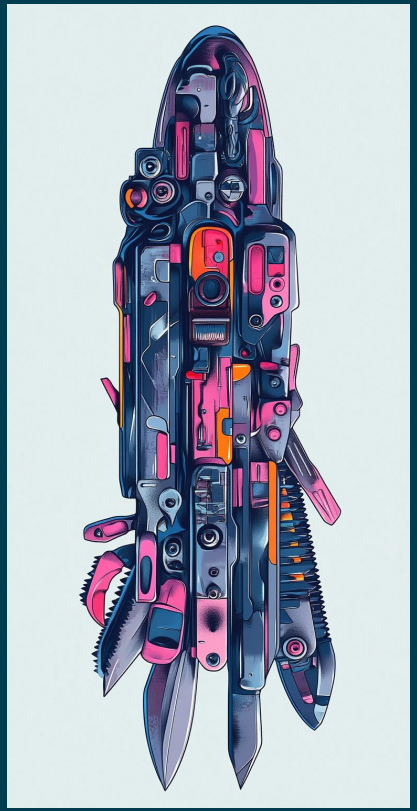
# Integrated chat



# Integrated chat



Midjourney: one big swiss army knife with lots of bizarre tools, vector based, flat areas of color



Creating new classes and components

Create SQL queries

Understanding a cron expression

Generate stubs

Creating new classes and components

Create test data

Transform a script into a more user-friendly CLI

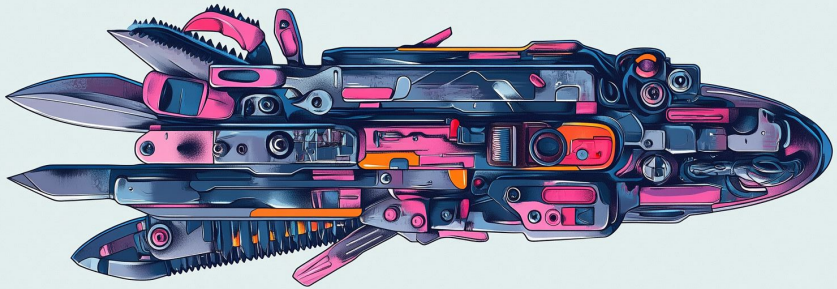
Build a browser extension to test a tricky bug

Find out if a technology has a “hard to Google” feature

Build a visualisation

## Coding assistants

# Core features



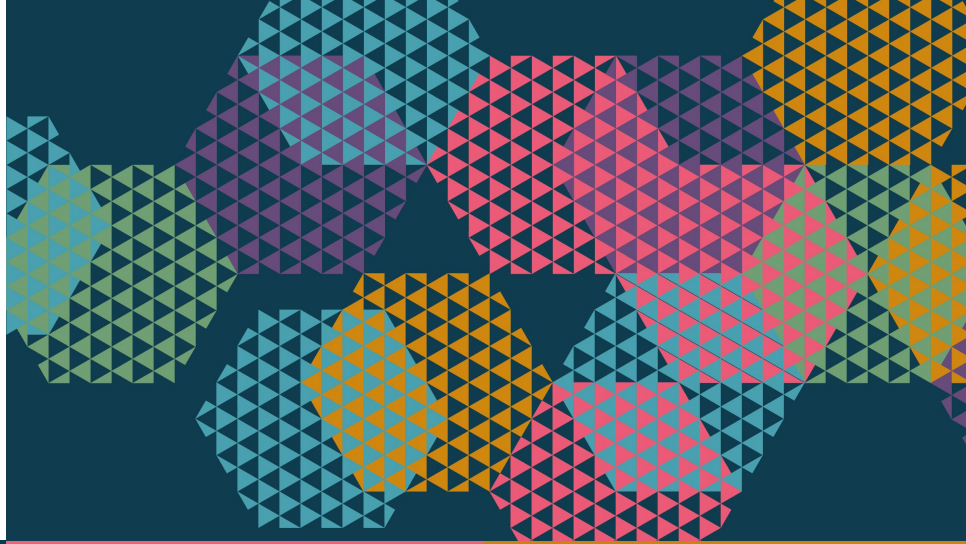
Inline suggestions

Integrated chat

© 2024 Thoughtworks

Coding assistants

# Model choices



Provided by vendor

Self-hosting

Bring your own key

Local models

© 2024 Thoughtworks

| Provided by vendor | Self-hosting | Bring your own key | Local models |
| --- | --- | --- | --- |
| How useful is the model for coding tasks? | | Where does my data go? | |
| How much money do I want to spend on maintenance? | | ~~Can I fine-tune the model with my own code?~~ | |

Coding assistants

# Context Orchestration



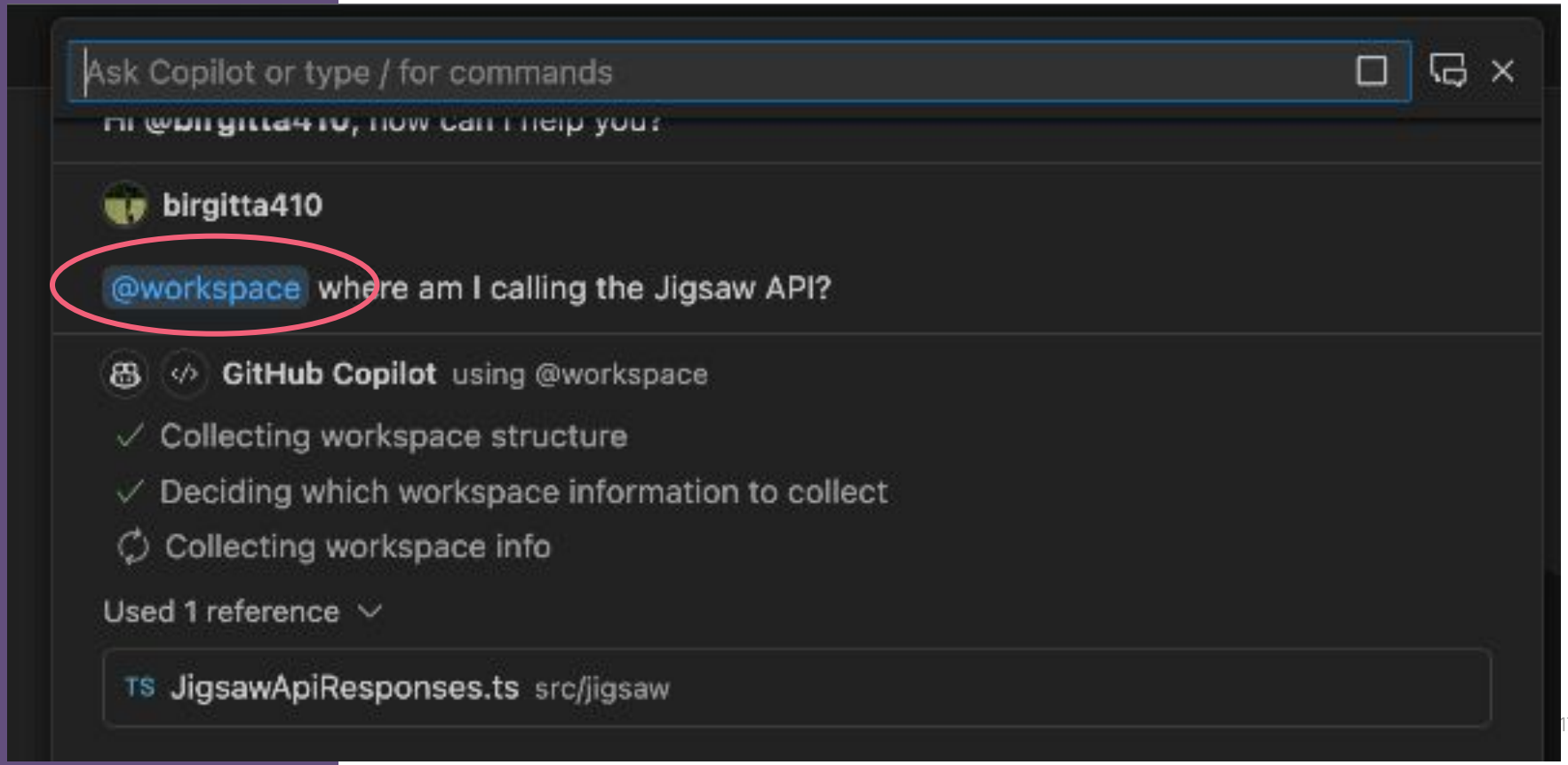
Current codebase

All your codebases

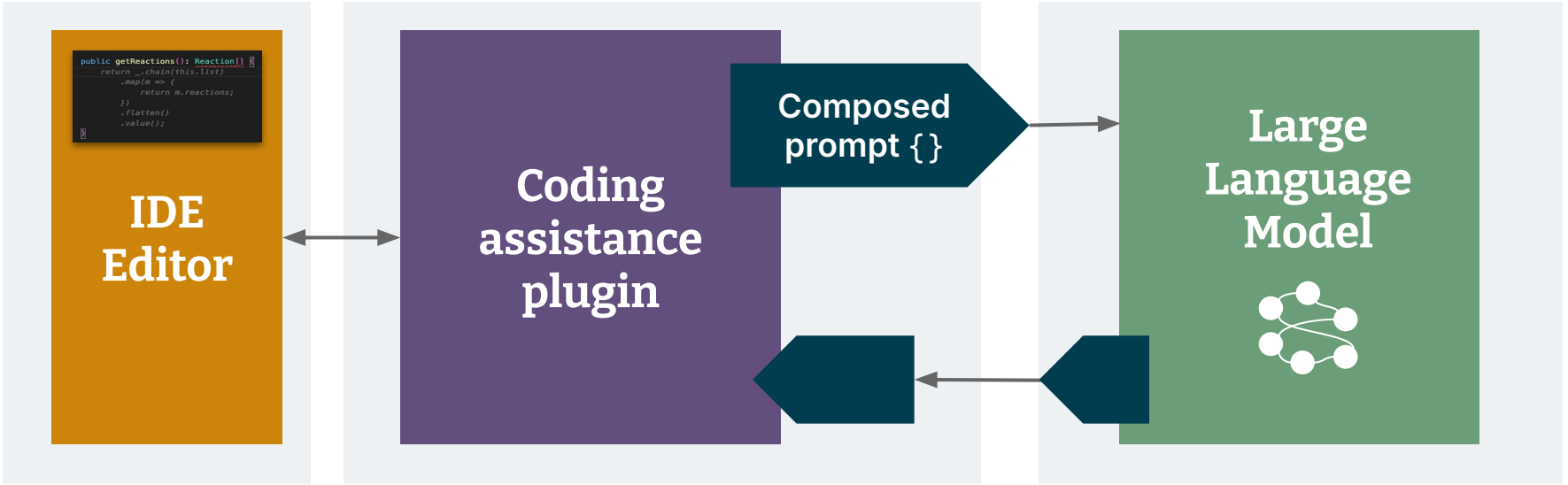
Other information sources

Your instructions

# Current codebase

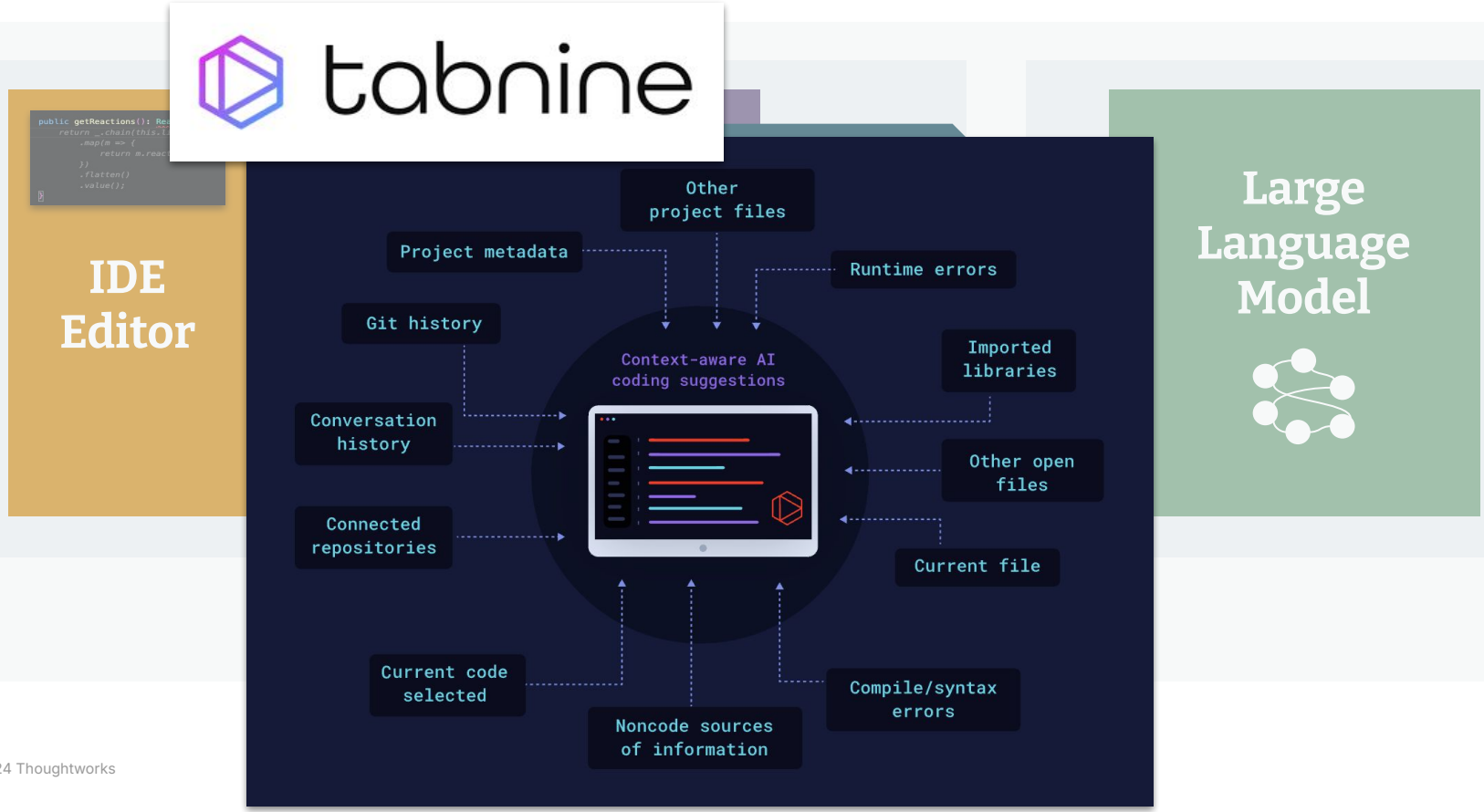


# AI tools are more than the model




© 2024 Thoughtworks

# AI tools are more than the model




© 2024 Thoughtworks

# Context of all your codebases

Copilot Enterprise

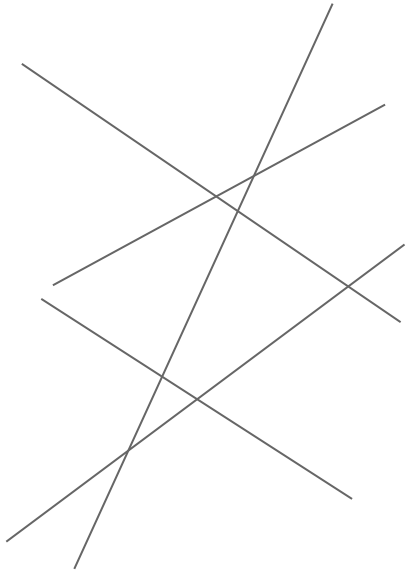
Customize Copilot to your organization and infuse AI throughout the software development lifecycle.

Cody

codeium

tabnine

...



GitHub

ATLASSIAN Bitbucket

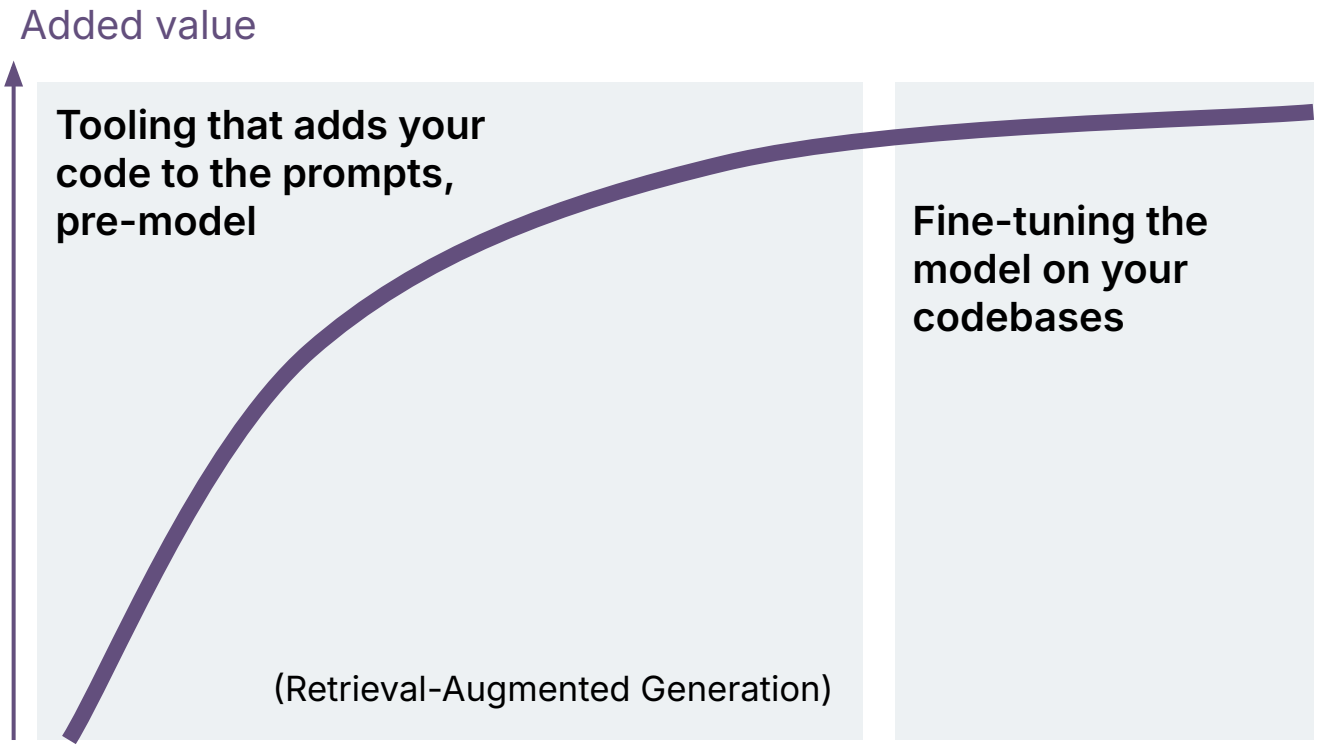
GitLab

...

Context of your codebase

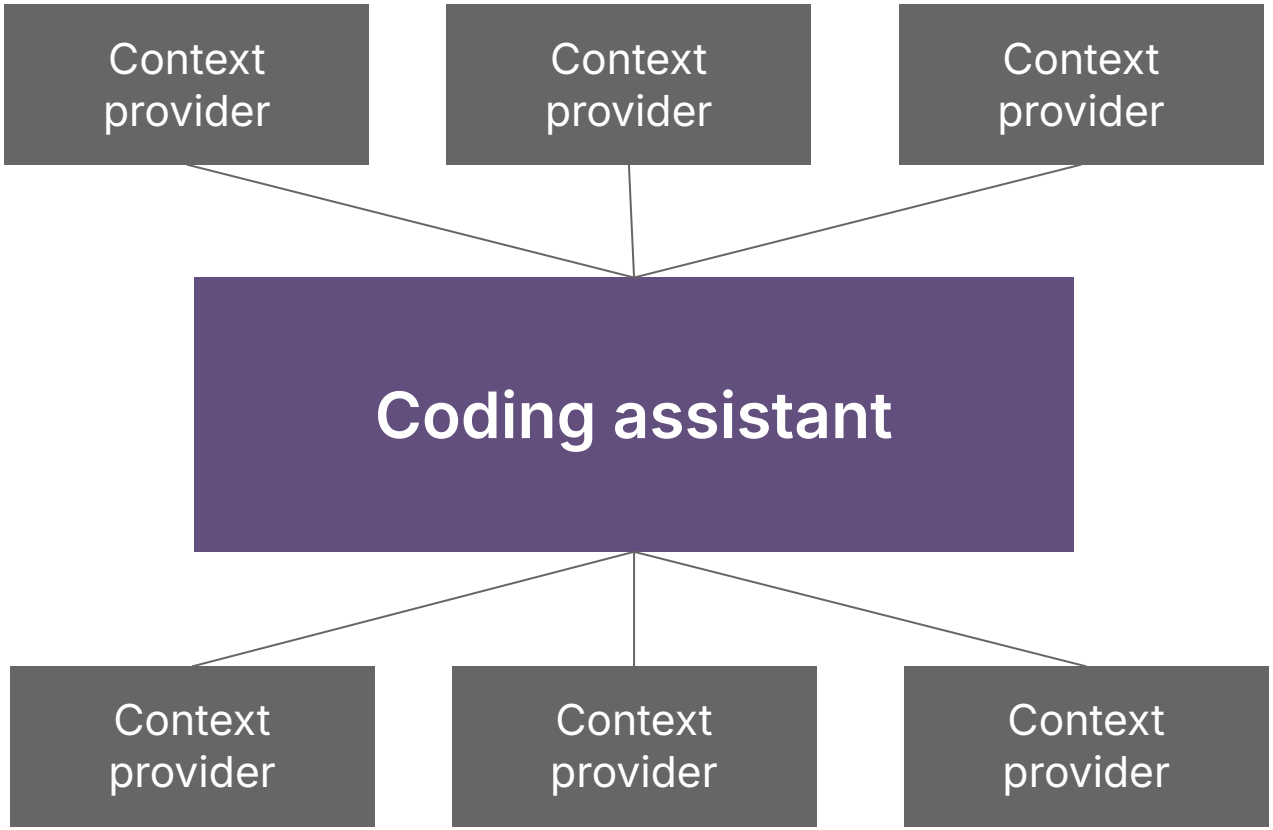
Context of all your codebases

# What about fine-tuning?

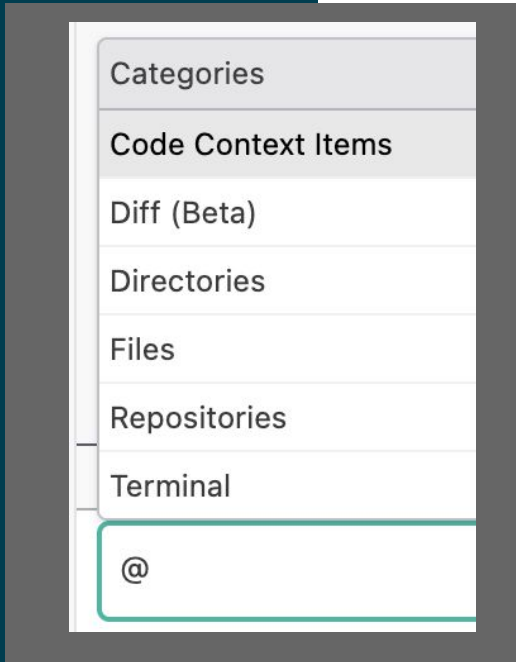



*Graph inspired by Tabnine*

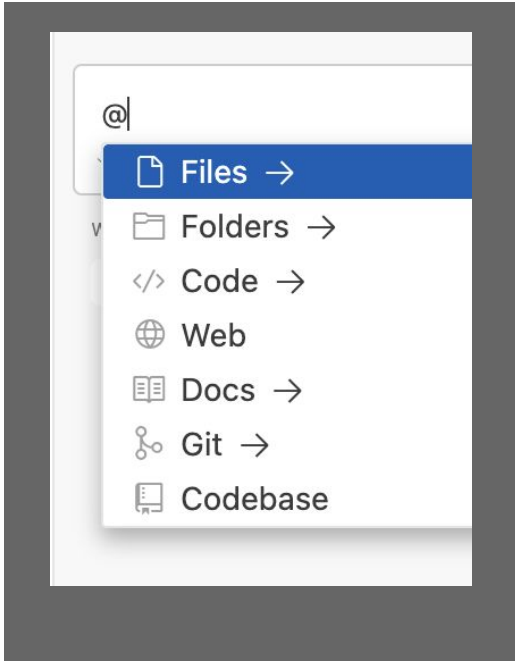
# Other information sources



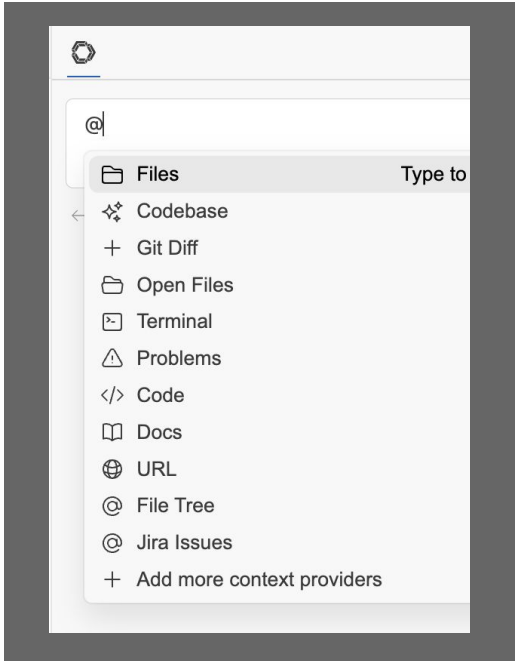
# Other information sources




codeium



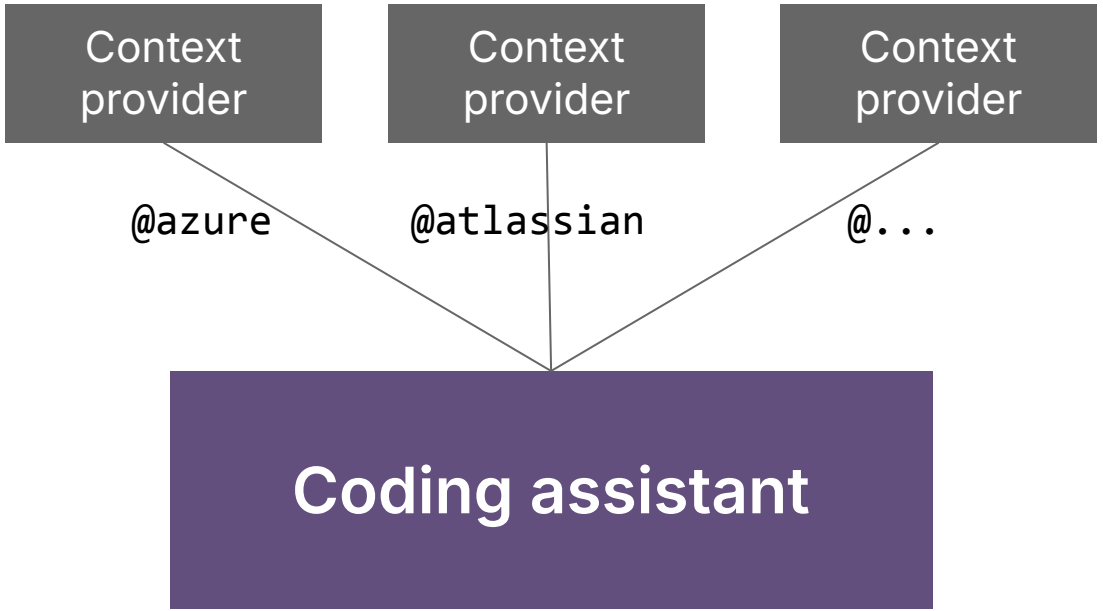

CURSOR




Continue

Other information sources

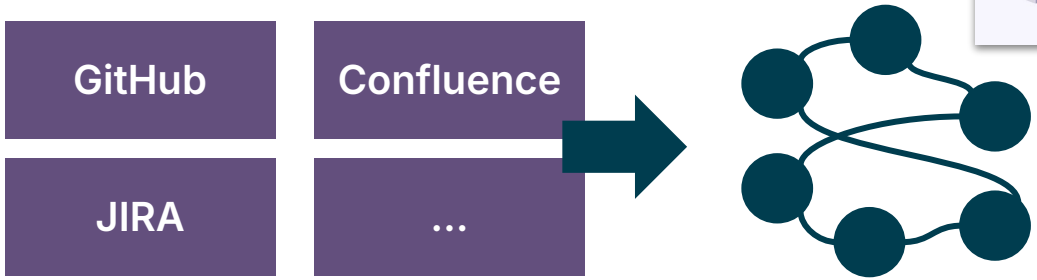
# A new ecosystem is emerging

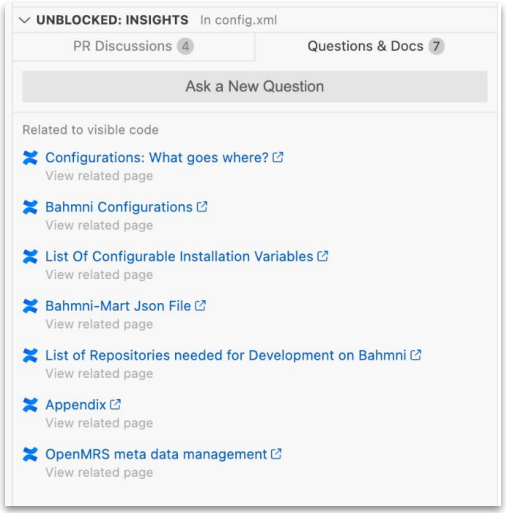


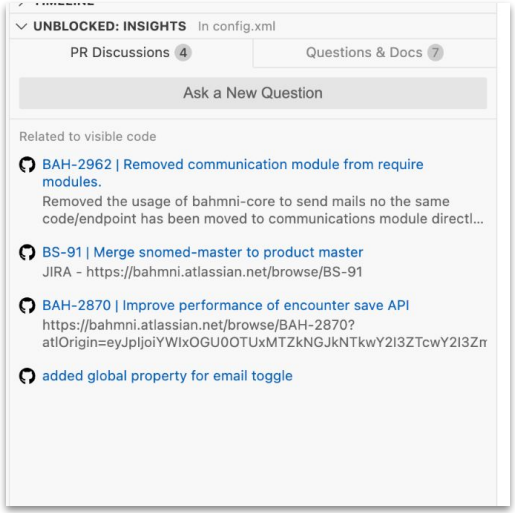
# Other information sources

# Or: Go for “all in one”

Unblocked



# Much of the coming improvements will come from tooling, not models

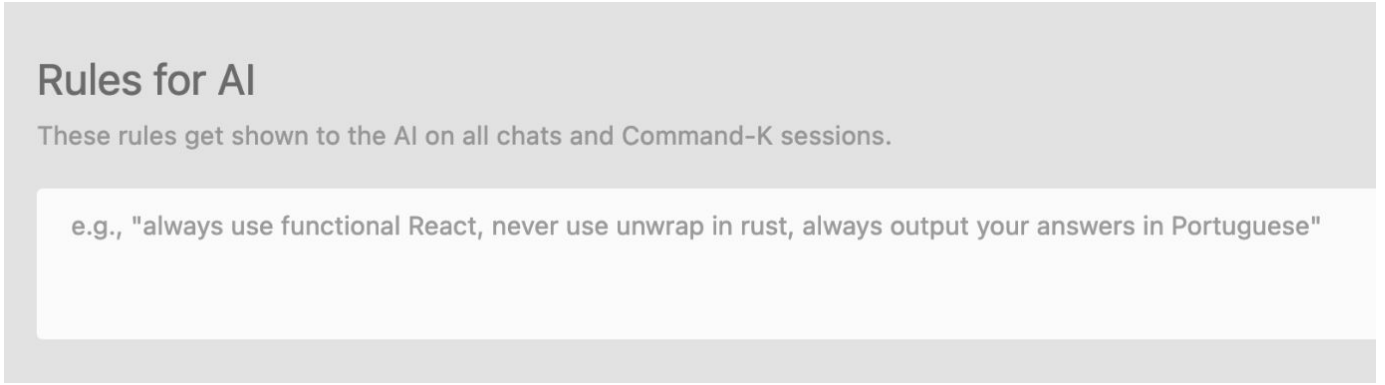


© 2024 Thoughtworks

# Custom instructions

# Workspace settings

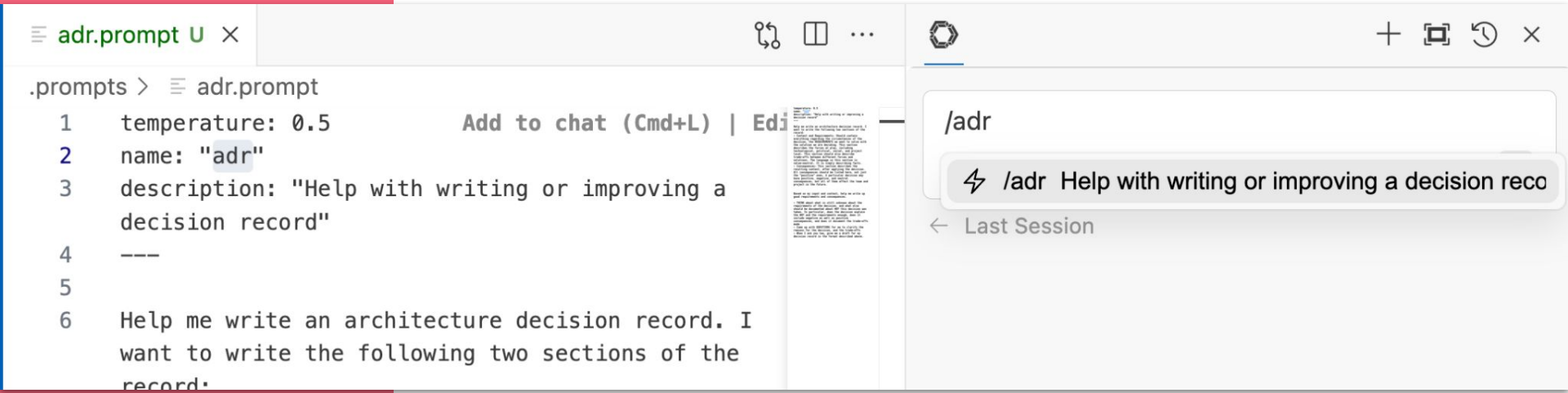


Custom instructions

# Team prompts

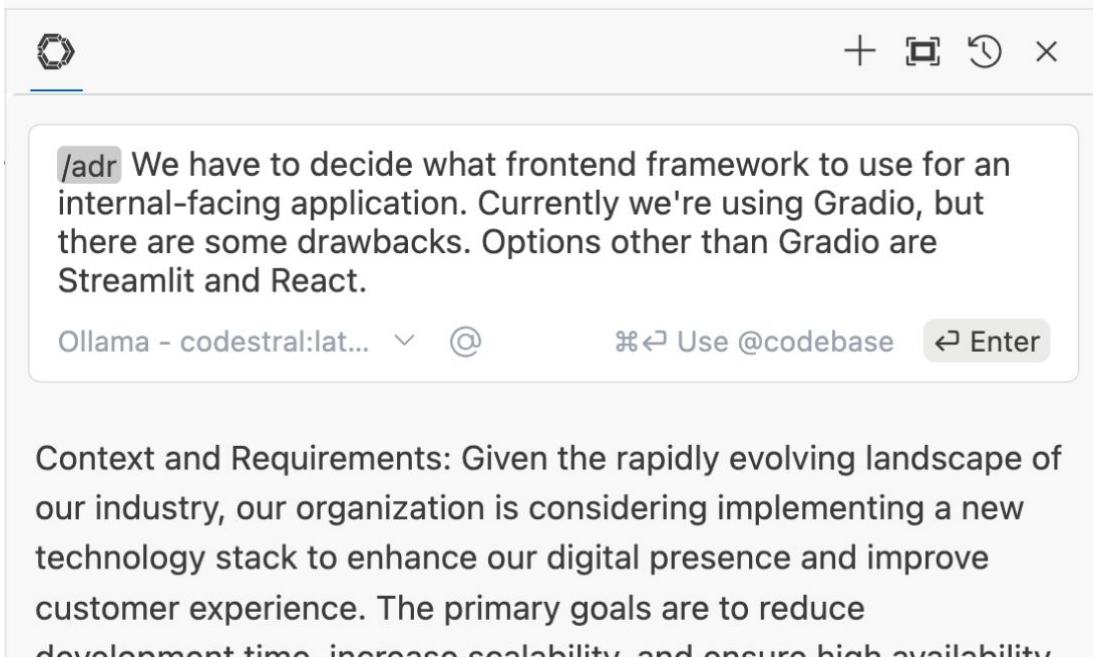
Continue



Custom instructions

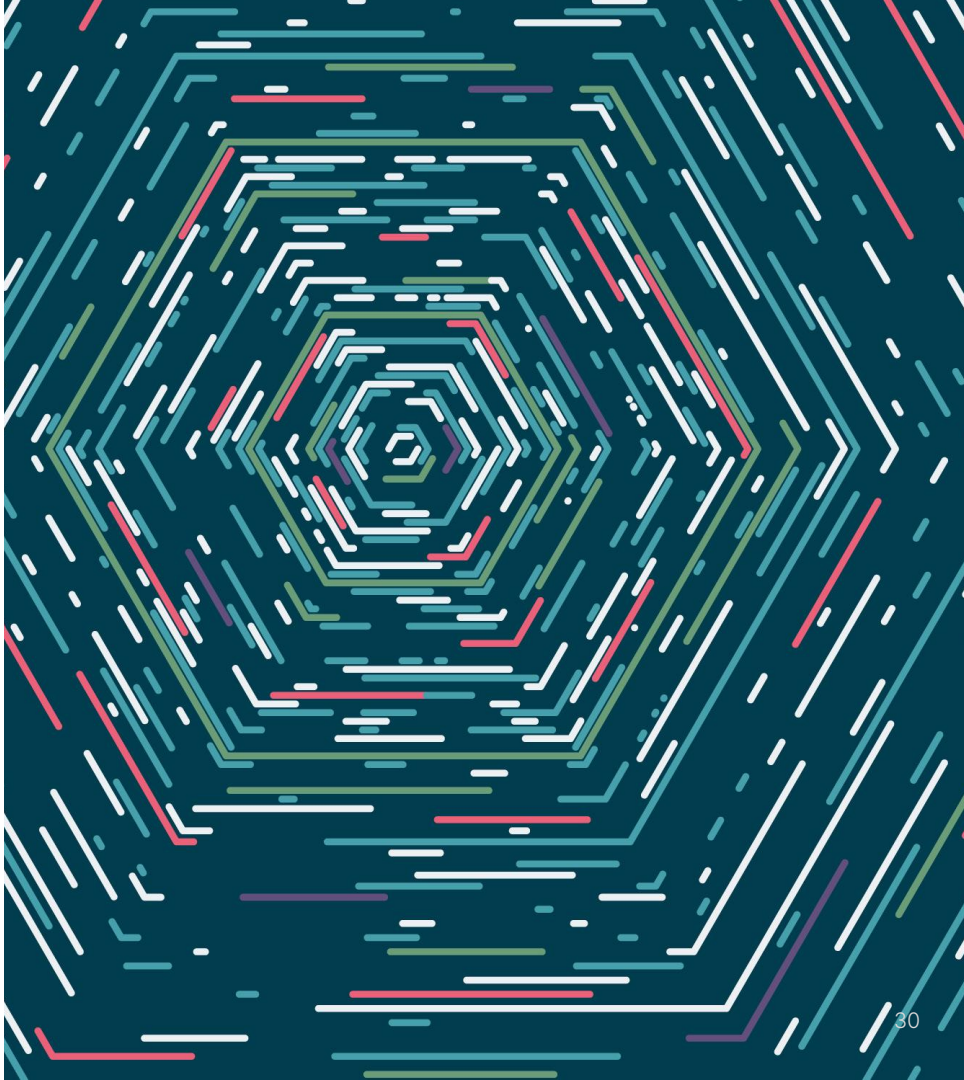
# Team prompts

Continue



## Coding assistants

# The next frontiers



© 2024 Thoughtworks

# Testing

There are coding assistants that specialise in test generation and improvement

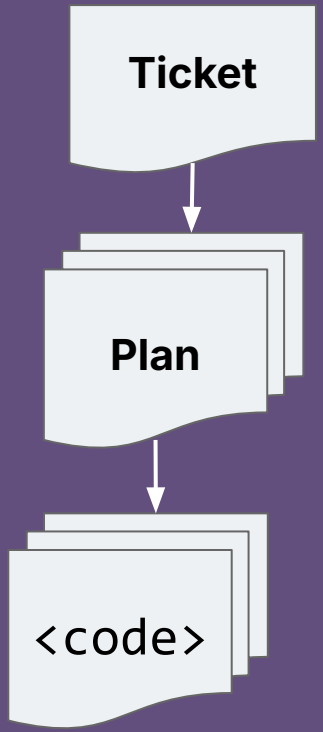



AI is great for brainstorming testing scenarios and coming up with test data.

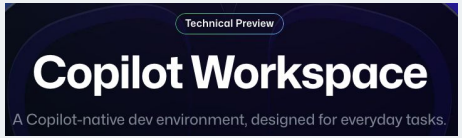
It’s risky to “just let AI write all the tests”. We need to rethink our testing practices.

Generating unit tests for your old and untested codebase won’t work.

# Solving larger problems



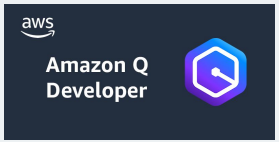

# Agents



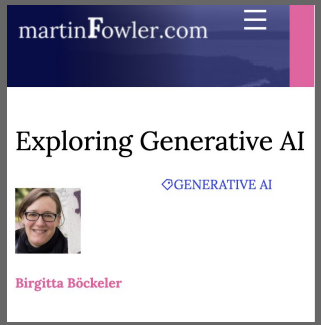







# They don't work — yet?







https://martinfowler.com/articles/exploring-gen-ai.html#memo-10

# Code review and refactoring

**Coding assistants today are much better at adding code than changing and improving code**

9 January 2024

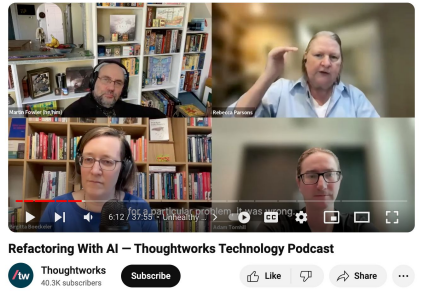
## Refactoring vs Refuctoring:

Advancing the state of AI-automated code improvements

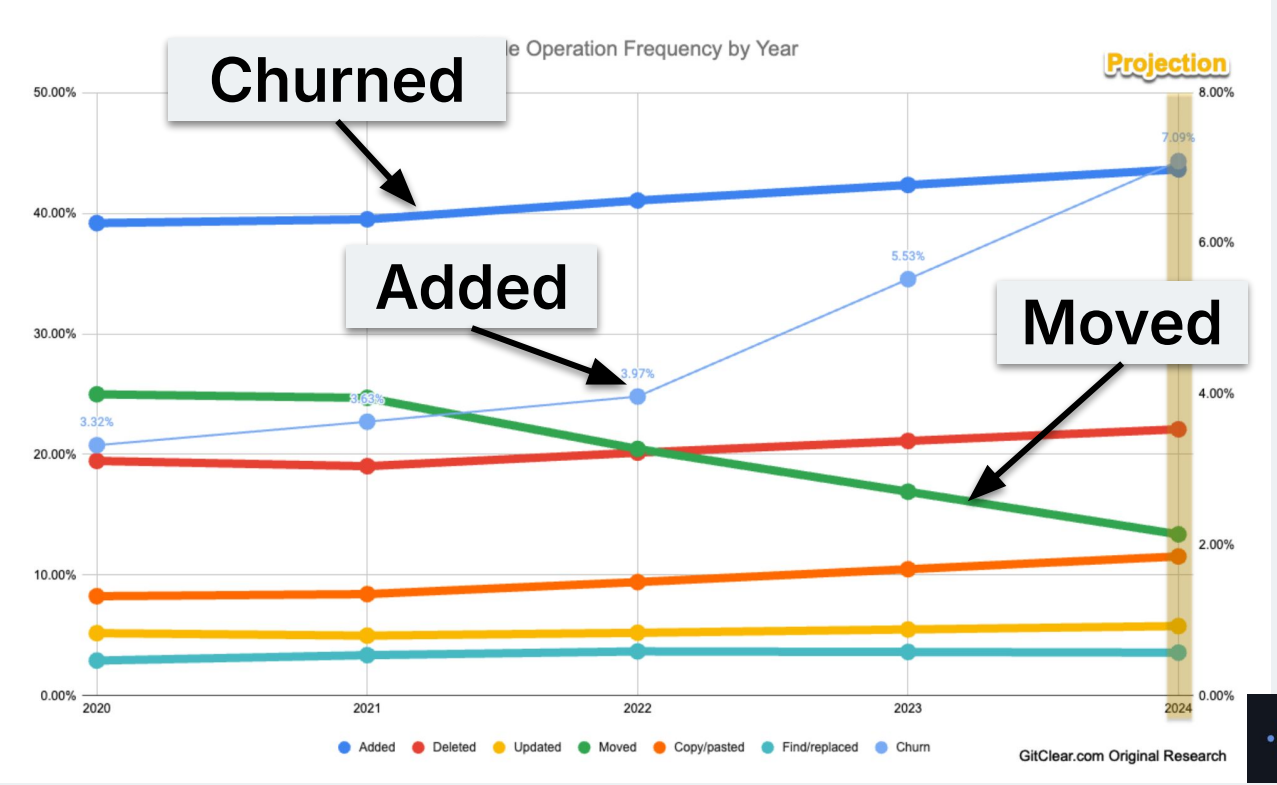
By Adam Tornhill, Markus Borg, PhD & Enys Mones, PhD

*"We find that existing AI solutions only deliver functionally correct refactorings in **37%** of the cases."*

https://codescene.com/hubfs/whitepapers/Refactoring-vs-Refuctoring-Advancing-the-state-of-AI-automated-code-improvements.pdf



# Is your codebase getting bigger faster?

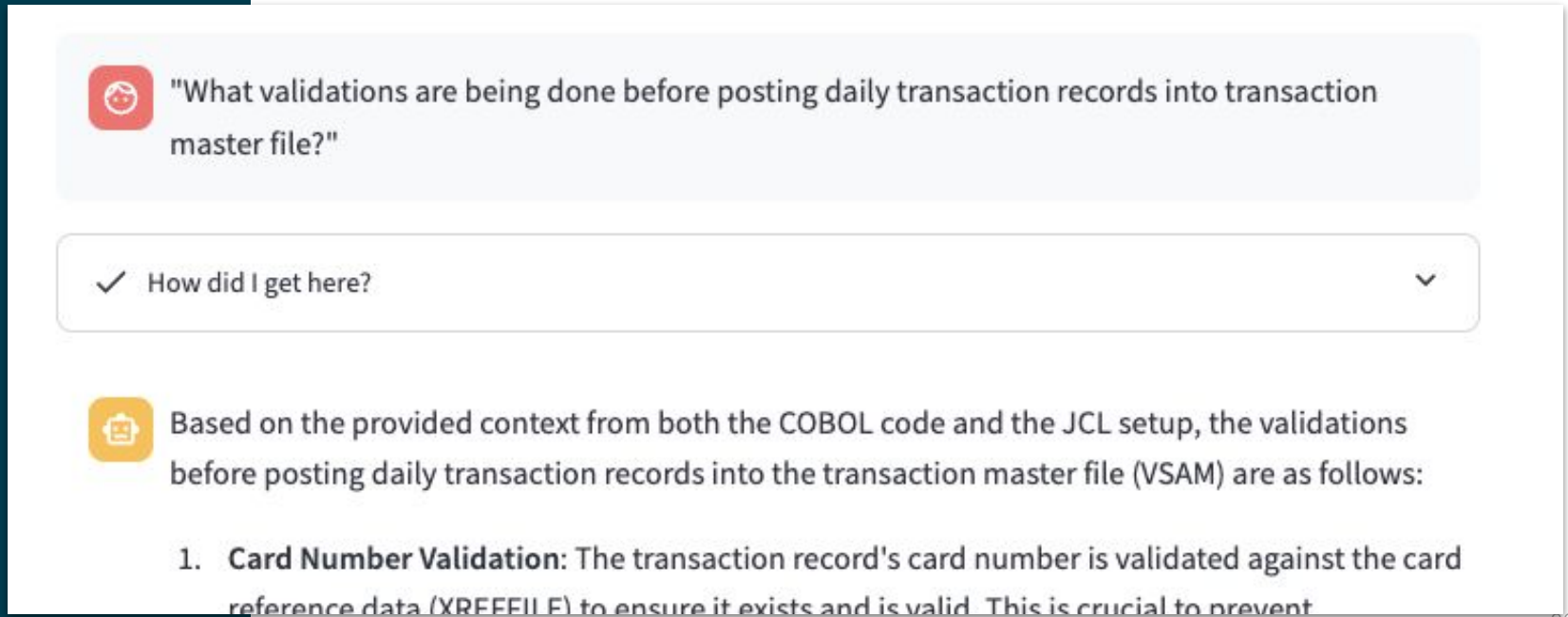



https://gitclear-public.s3.us-west-2.amazonaws.com/Coding-on-Copilot-2024-Developer-Research.pdf

# Documentation

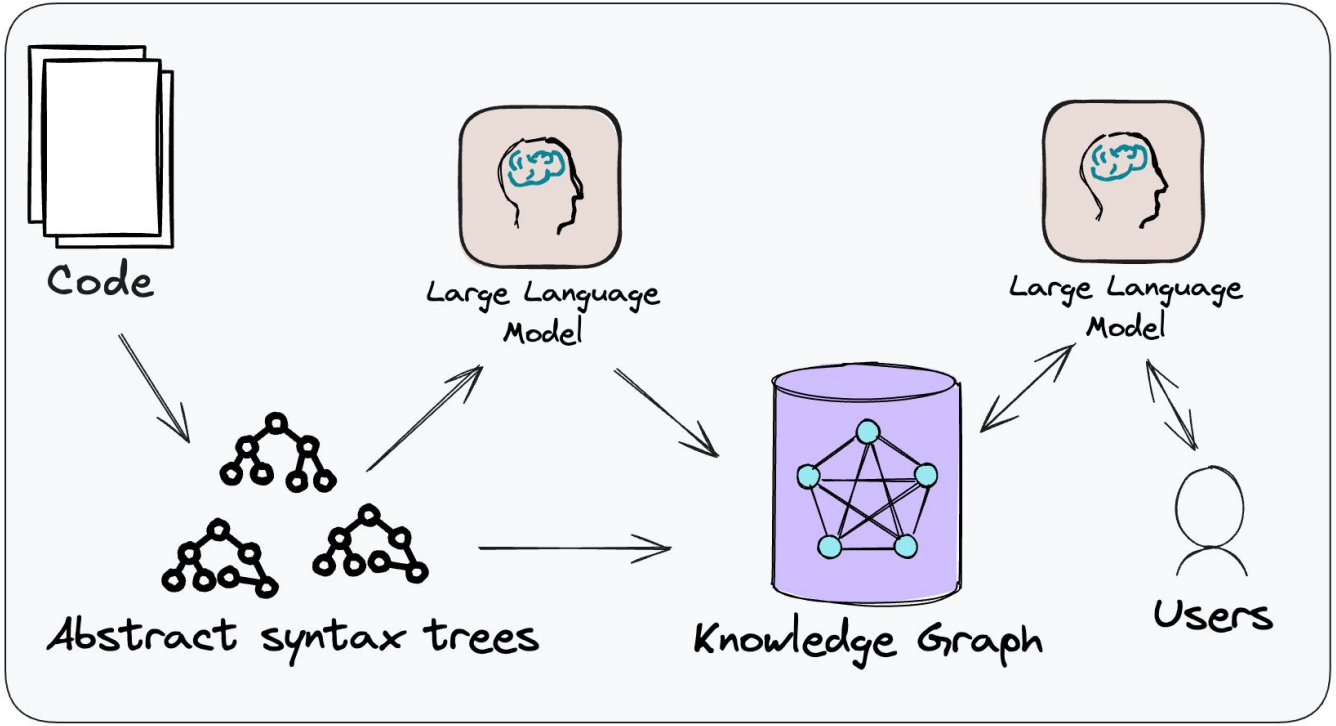
~~Documentation~~

# Unlocking “code as knowledge”



~~Documentation~~

# Unlocking “code as knowledge”




“Legacy Modernization meets GenAI”
Alessio Ferri, Shodhan Sheth, Tom Coggrave
https://martinfowler.com/articles/legacy-modernization-gen-ai.html

# Debugging

CURSOR

"Looks like this is related to..."

"To debug this issue, we need to locate the relevant pom.xml file and..."

*"Searching for pom.xml..."*

```
[WARNING]
[WARNING] Some problems were encountered while building
[WARNING] 'dependencyManagement.dependencies.dependency.
[WARNING] 'dependencyManagement.dependencies.dependency.
```

undefinedPlease help me debug this code. Only debug the latest erro

USED

No context

Based on the output, it seems that the latest error is related to duplicate declarations of the version `${junitVersion}` for the `org.junit.jupiter:junit-jupiter-params` dependency. To debug this issue, we need to locate the relevant `pom.xml` file and find the duplicate declaration.

Let's search for the `pom.xml` file in the `openmrs-core` directory and check the relevant lines.
*Searching for `pom.xml` ...*

There are multiple `pom.xml` files in the `openmrs-core` directory. To debug the issue, we need to check the `pom.xml` file in the root directory. Let's read the contents of that file and find the duplicate declaration of the `org.junit.jupiter:junit-jupiter-params` dependency.

*Reading `pom.xml:1` ...*

I have found the `pom.xml` file in the root directory of `openmrs-core`. Let's take a look at the relevant section where the duplicate declaration of the `org.junit.jupiter:junit-jupiter-params` dependency is mentioned.

Lines 64-75 of `pom.xml`:

```
<!-- OpenMRS Dependencies -->
<dependency>
    <groupId>org.openmrs.tools</groupId>
    <artifactId>openmrs-tools</artifactId>
```

# Cast of main characters

## The leader




## The original




## The open source leader




## "The other 2"







## The developer favorite




## The librarian




## The IDE expert




## The confidence booster

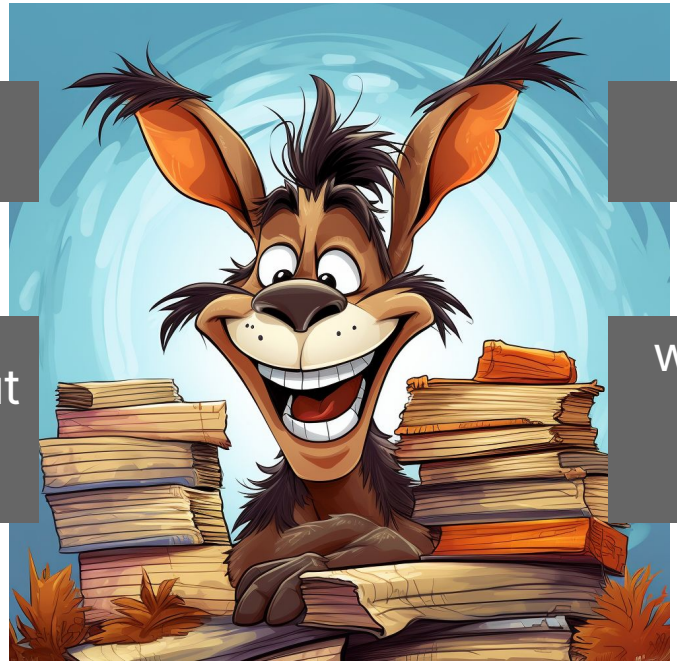



© 2024 Thoughtworks

# What all characters have in common

eager to help

stubborn



**very** well-read, but inexperienced

won’t admit when it doesn’t know something

Midjourney: eager stubborn donkey happy books computer; cartoon, vector based, flat areas of color

https://martinfowler.com/articles/exploring-gen-ai.html#memo-08

© 2024 Thoughtworks

# Final thoughts

Understand the basics of how it works

Embrace that "AI is terrible software" (Ethan Mollick)

(Apart from your org's guardrails) Don't worry too much about tool choice, you can always switch

As a profession, we have to reflect on our practices

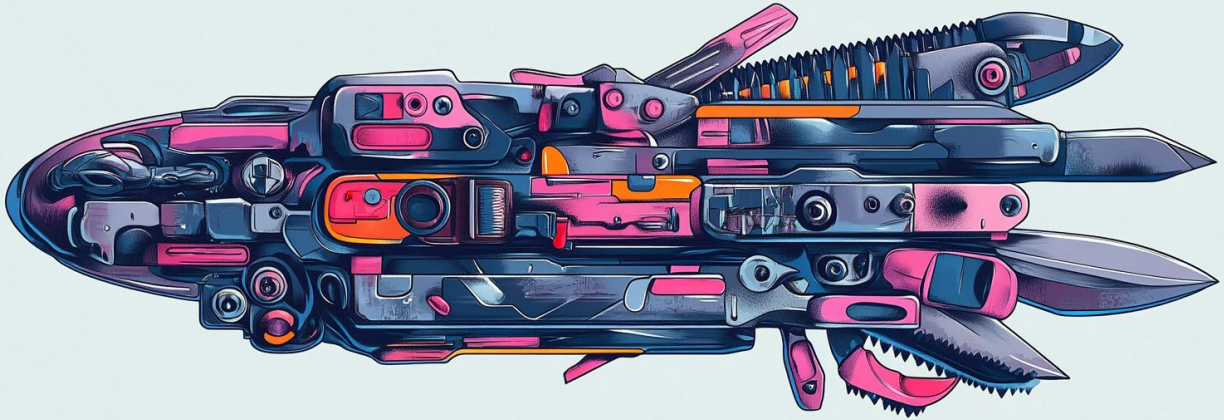
© 2024 Thoughtworks

# All links from this talk

https://gist.github.com/birgitta410/487080bcb95c0d67e2c4c3cd20e2d306

© 2024 Thoughtworks

# Thank you!



birgitta.info

@bboeckel@toot.thoughtworks.com

https://gist.github.com/birgitta410/487080bcb95c0d67e2c4c3cd20e2d306

/thoughtworks

© 2024 Thoughtworks