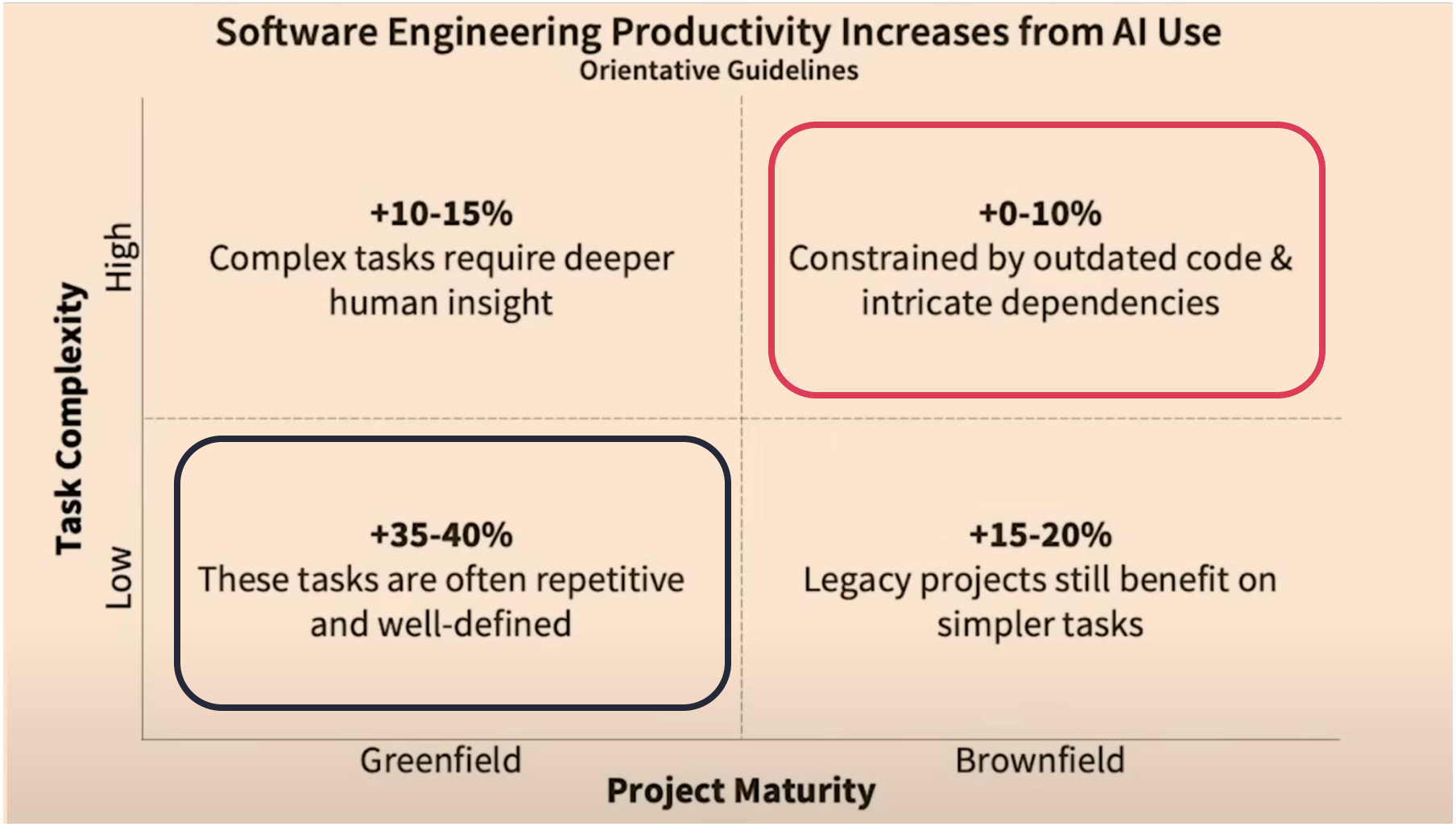
beyond vibe coding:
a call for a spec language

Markus Höfling
Business Engineering @ DevBoost

vibe coding:

"fix one thing, break ten others."

DUSAN OMERIC (CODEPLAIN, "BEYOND VIBE CODING" 2025)



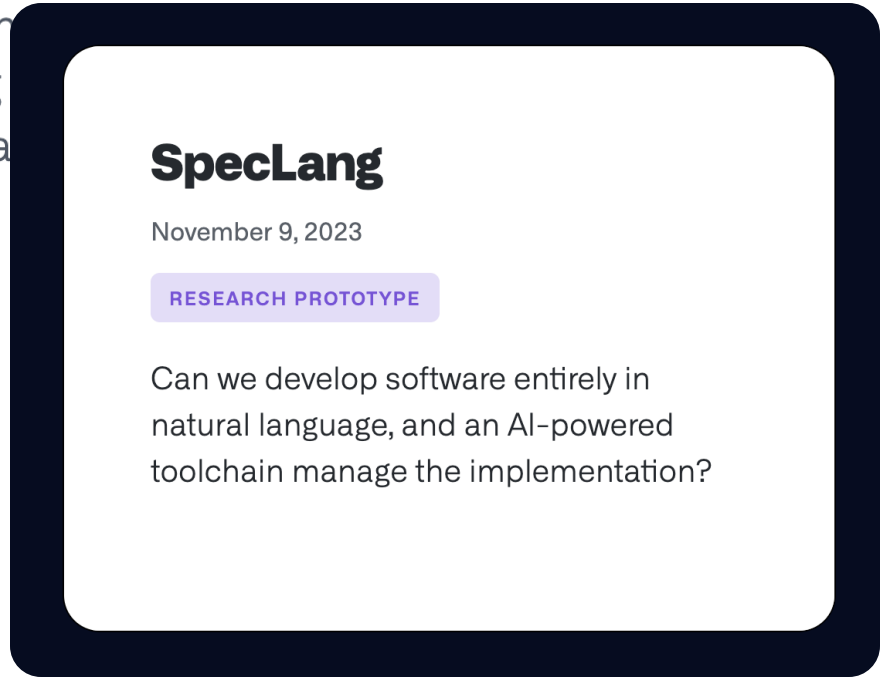
Stanford Study on AI's Impact on Developer Productivity (*https://www.youtube.com/watch?v=tbDDYKRFjhk)*

kiro.dev by AWS

GitHub Next

GitHub Next investigates the future of software development.

We are a team of researchers and engineers at GitHub, exploring things beyond the adjacent possible. We prototype tools an
We identify new approaches to building
tea



Feature / Task Idea

Q&A with LLM to clarify the WHY and WHAT

critical step!
-> may require multiple iterations
due to ambigouity of natural language!

Generate Requirements

```
# Requirements Document

## Introduction

This feature enables administrators and users to customize the logo
displayed on the LibreChat login page. The current login page
displays a default logo, and this feature will provide the ability to
replace it with a custom logo while maintaining proper styling,
responsiveness, and accessibility standards.

## Requirements

### Requirement 1

**User Story:** As an administrator, I want to replace the default
login page logo with a custom logo, so that I can brand the
application according to my organization's identity.

#### Acceptance Criteria

1. WHEN an administrator uploads a custom logo THEN the system SHALL
display the new logo on the login page
2. WHEN no custom logo is configured THEN the system SHALL display the
default LibreChat logo
3. WHEN a custom logo is uploaded THEN the system SHALL validate the
```

Generated Code

```python
def fetch(limit=100):
    headers = {"Authorization": f"Bearer {API_KEY}"}
    cursor = None
    users = []

    while True:
        params = {"limit": limit}
        if cursor:
            params["cursor"] = cursor

        r = requests.get(f"{BASE_URL}/v1/users",
                headers=headers, params=params)
        r.raise_for_status()

        payload = r.json()
        users.extend(payload.get("data", []))
        cursor = payload.get("next_cursor")
        if not cursor:
            break

    return user
```

SpecLang

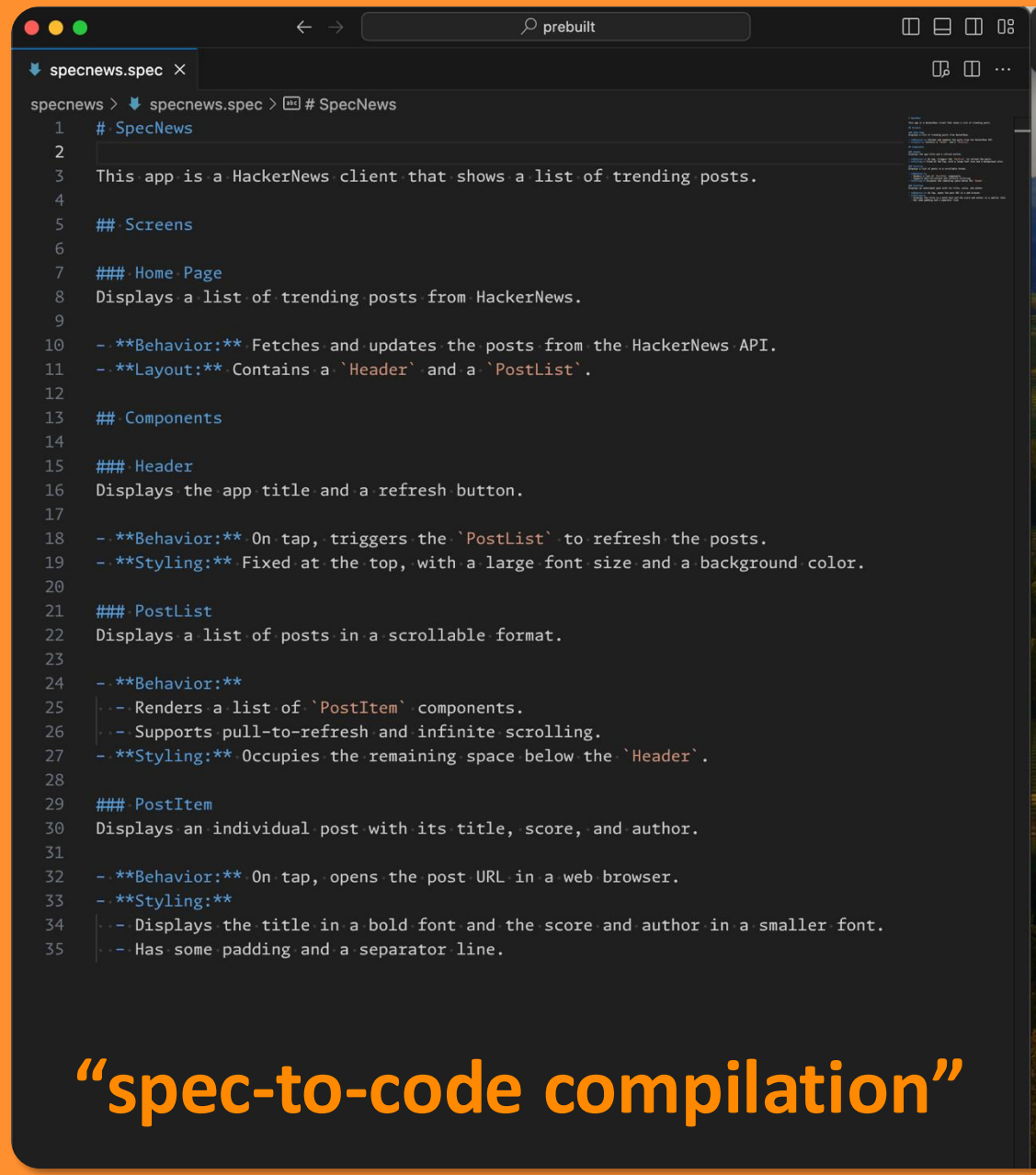
Can we develop software entirely in natural language, and let an AI-powered toolchain manage the implementation?

WHAT'S IT FOR?

End-to-end development in natural language

STAGE

RESEARCH PROTOTYPE



“spec-to-code compilation”

https://githubnext.com/projects/speclang/

*** plain

***plain — a specification language that combines the efficiency of natural language with the control and precision of code.

"To move **beyond vibe coding** […], developers must be able to **fix any issue entirely within the specifications** — **without** ever **touching the underlying code**."

*currently under development by members of **Codeplain**, **SpecLang** & **former member of original GitHub Copilot Team**

** first **release** is planned for **fall 2025**

https://www.codeplain.ai/

codeplain

spec-driven, production-ready code generation

*codeplain is an ai-powered code generation service that produces production-ready software from specifications written in ***plain language**.

JOIN WAITLIST

*** plain

Specification in ***plain

intercom-client.plain

```
{% include "python-console-app-template.plain" %}

***Definitions:***

- The App interacts with Intercom API (The API).
- The List of Users is the list of Intercom API users.

***Non-Functional Requirements:***

- The resource [intercom-openapi.json](https://github.com/Codeplain-ai/intercom-openapi.json) describes The API.

***Test Requirements:***

- The Conformance Tests should use the real server of The API - do not use mock server.
- The resource [intercom-credentials.json](../credentials/intercom-credentials.json) contains credentials that can be used for testing.

***Functional Requirements:***

- The App should take credentials json as the only positional argument.
- Authenticate with The API.
- Fetch The List of Users from The API.
- Print The List of Users to the console.
```

Generated Python code

intercom-client.py

https://www.codeplain.ai/

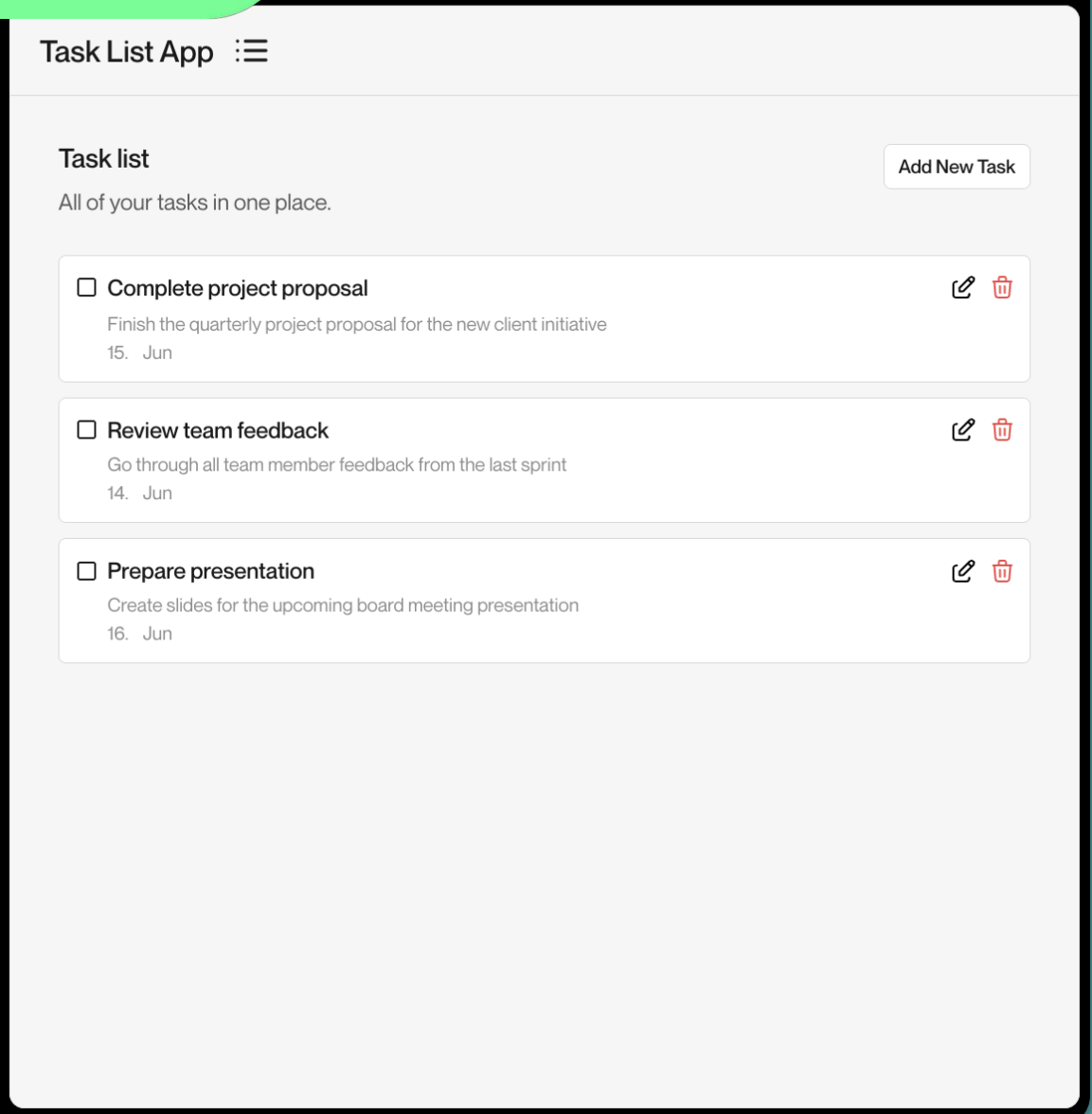
*** plain

Specification in ***plain

task_manager.plain

```
{% include "typescript-react-app-template.plain" %}

***Definitions:***

- The User is the user of The App.
- The Task describes an activity that needs to be done by The User. The Task has the following attributes
    - Name - (required) - a short description of The Task. The name must be at least 3 characters long.
    - Notes - additional details about The Task.

***Test Requirements:***

- End-to-end tests of The App should be implemented in React using Jest.

***Functional Requirements:***

- Show The Task List. The details of the user interface are provided in [task_list_ui_specification](https://github.com/codeplain/task-manager.png).
- The User should be able to add The Task. The details of the user interface are provided [new_task_modal_specification](https://github.com/codeplain/new-task.png).
- The User should be able to delete The Task.
- The User should be able to edit The Task.
- The User should be able to mark The Task as completed.
```

Generated App



https://www.codeplain.ai/

Dimension	Vibe coding	Spec-driven with ***plain
Time to first run	Very fast	Fast (testing adds time)
Source of truth	Codebase	The specification
Tests	Requires explicit prompt	Automated test generation and execution
Change safety	Manual regressions	Auto-tested
Collaboration	Review code after generation	Review the spec before generation
Maintainability	Declines as complexity grows	Stays maintainable
Barrier to entry	Lower (freeform text)	Higher (learn ***plain primitives)

https://blog.codeplain.ai/p/from-vibes-to-specs