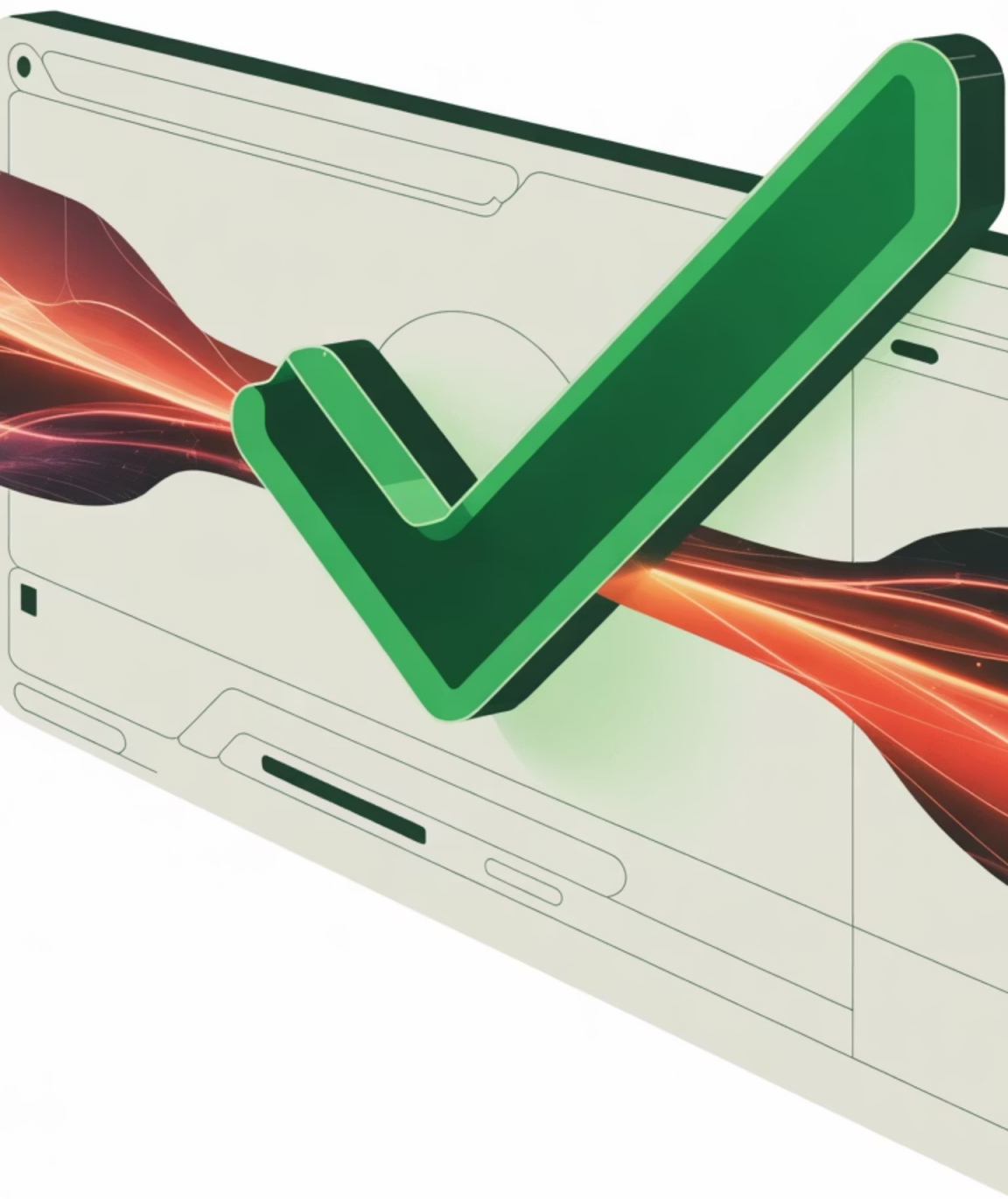


# AI Experiments

To Make Informed Decisions





**AI Says It's OK**  
**So, Ship It!**



# Quality at Speed

**Pascal Dufour**

**AI development Consultancy**

**Agile Test Consultancy**



# How AI Experiments Support Our Decision-Making

## What is an AI experiment?

- The same as an experiment, now with AI as a deciding factor.

## What is decision making?

- When information leads to a choice.





### Tech consulting

Onze architecten vertalen jouw businessvraag snel naar een werkbare oplossing.

### Softwareontwikkeling

Onze eigen teams in Portugal en China bouwen onder Nederlandse regie, met slimme inzet van AI om development te versnellen.

### IT-operations & support

We zorgen voor de operationele continuïteit van alles wat we bouwen, van hosting en monitoring tot onderhoud en support.

### Procesoptimalisatie met AI

Ontdek hoe procesoptimalisatie met AI je bedrijf sneller en slimmer laat groeien.



# "Insurance" Company

**DELA**  
voor elkaar

 **Active Insurances**

6M

 **Funerals Handled**

50K+

 **Cremations**

50K+

 **Total Employees**

3000

 **IT Staff**

250



# Organizational Design Differences

## Insurance Company

### Scaled Agile Framework (SAFe) / Scrum Hybrid

- PI planning
- IT teams for IT
- Business IT teams  
("feature teams")
- ...

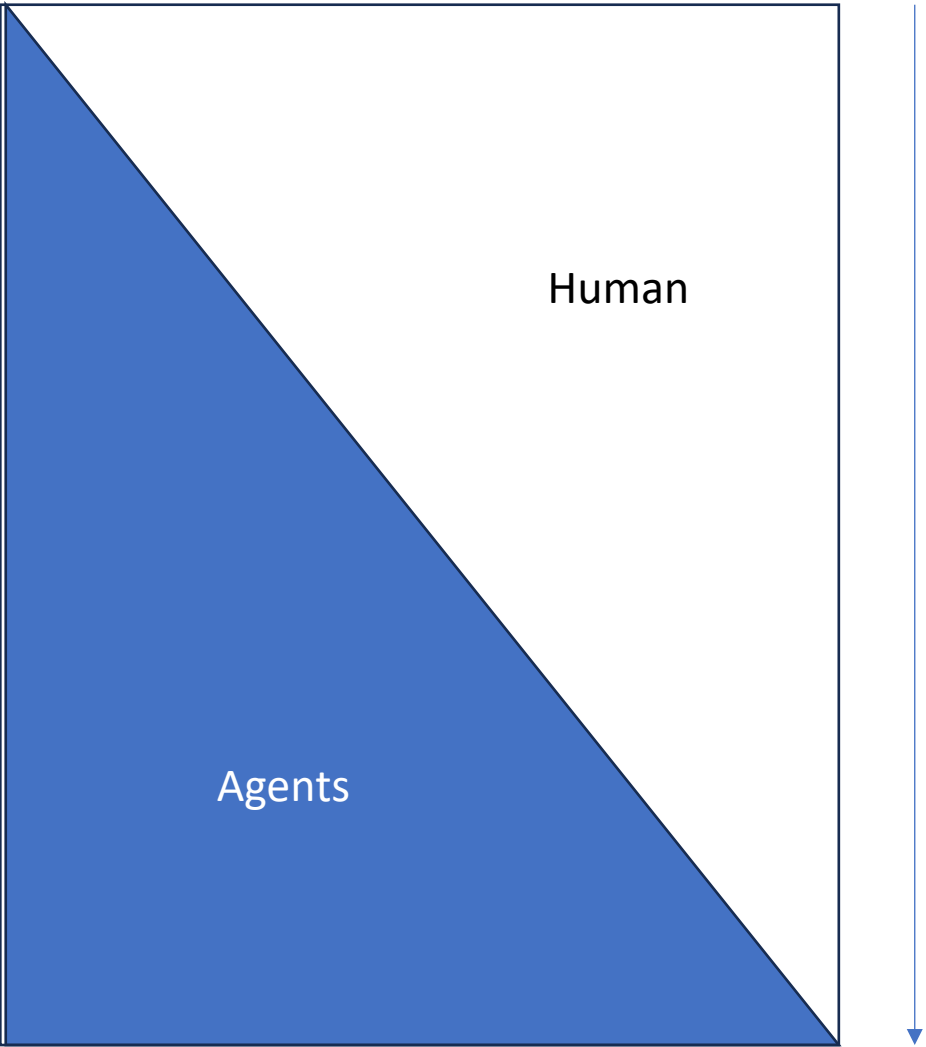
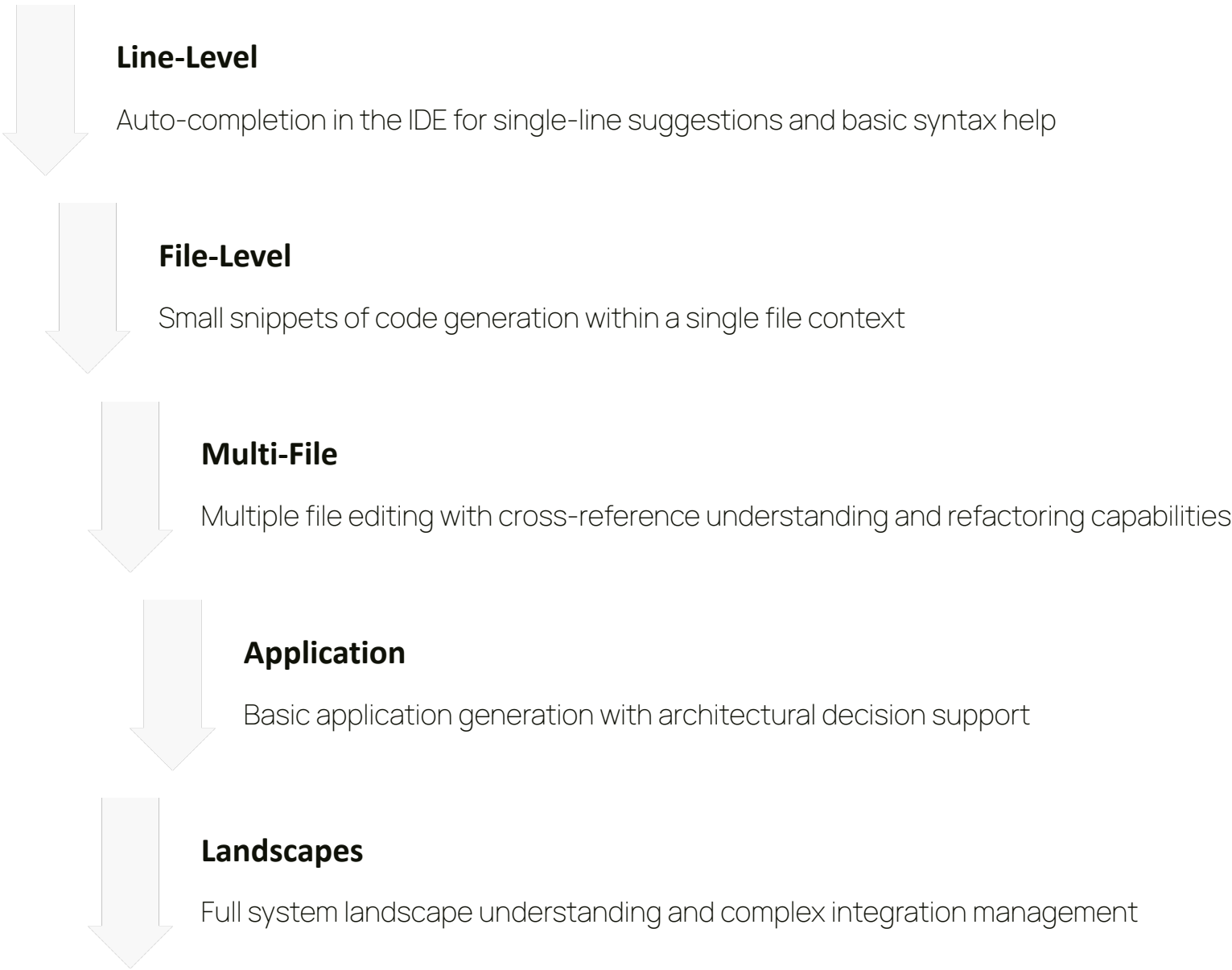
## Development Company

### Project-Based / Scrum

- AI team
- Feature teams  
DevOps
- ...

# AI Transitions in the Dev World

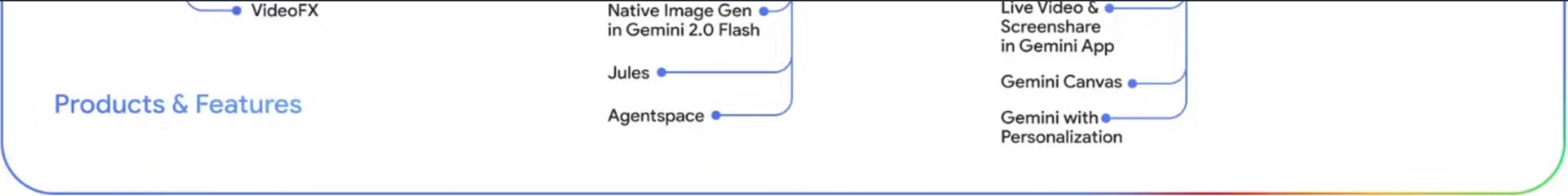
## Scope





# A year of Gemini progress

amazon-bedrock/eu.anthropic.cl	Claude Sonnet 4.5 (EU)
amazon-bedrock/global.anthropi	Claude Sonnet 4.5 (Global)
amazon-bedrock/us.anthropic.cl	Claude Sonnet 4.5 (US)
amazon-bedrock/cohere.command-	Command R
➔ <b>amazon-bedrock/cohere.command-</b>	<b>Command R+</b>
amazon-bedrock/us.deepseek.r1-	DeepSeek-R1
amazon-bedrock/deepseek.v3-v1:	DeepSeek-V3.1
amazon-bedrock/google.gemma-3-	Gemma 3 4B IT
amazon-bedrock/google.gemma-3-	Google Gemma 3 27B Instruct
23/689	

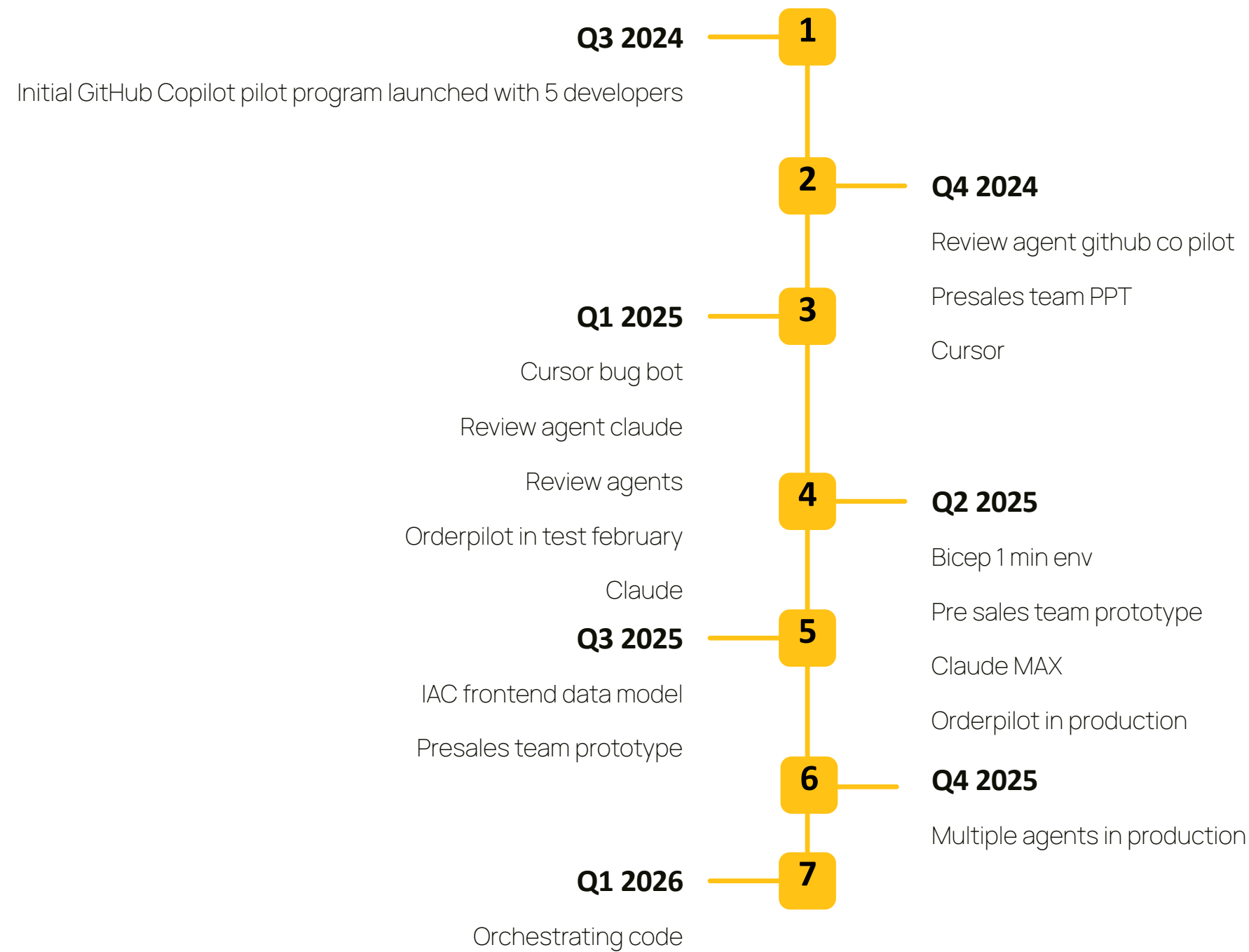






## **Timeline: Development Company**

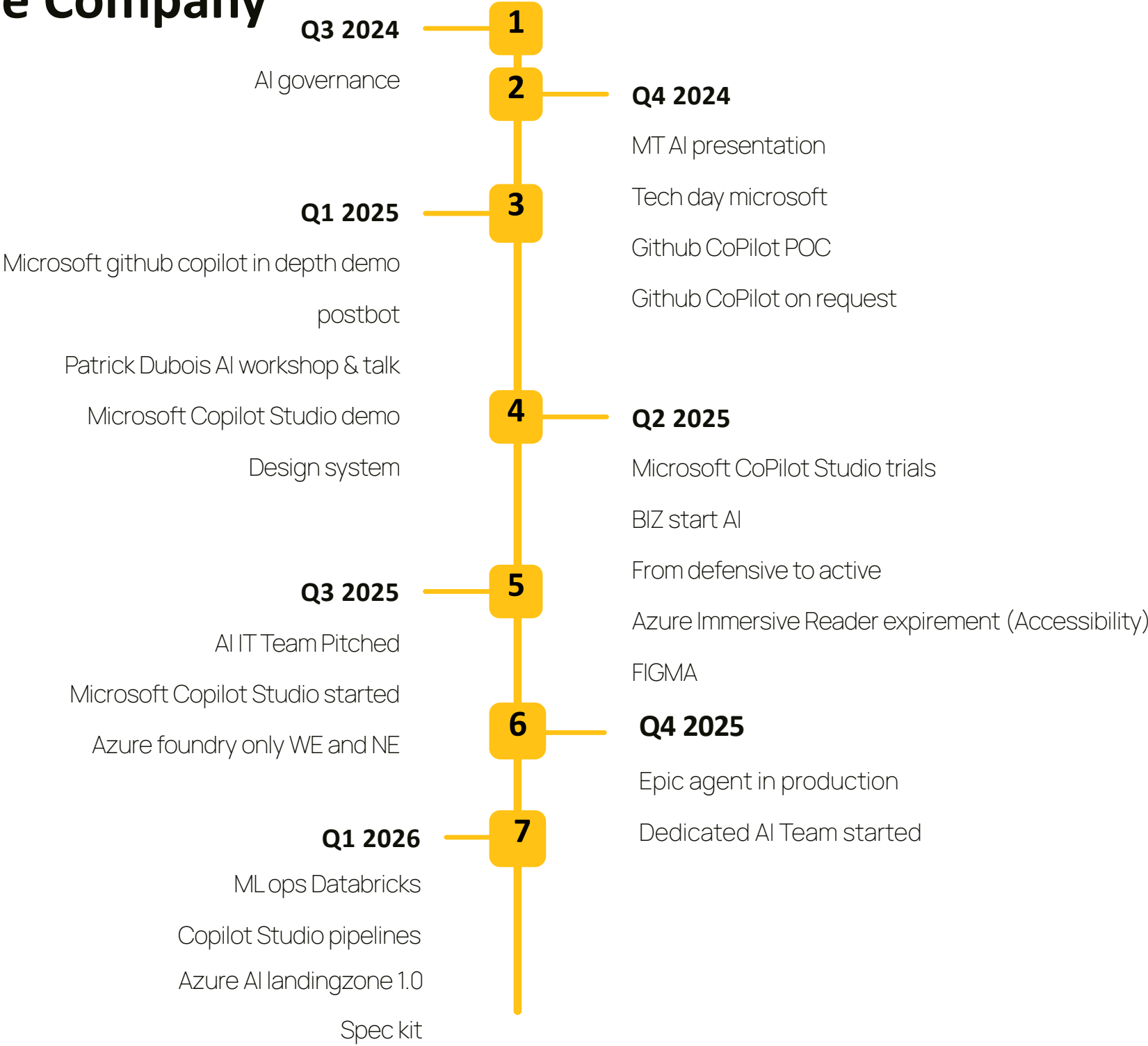
# Timeline: Development Company





## **Timeline: Insurance Company**

# Timeline: Insurance Company





## Key Timeline Differences

**Development company  
started 3 months earlier  
on Co-Pilot**

**Insurance company gets  
more formal training /  
information sessions**

**AI team formed in  
development company in  
2025 Q2 in insurance  
2025 Q4**



# Experiments Worth Mentioning

## Development Company

- Coding assistants
- **Review bot**
- Prototyping with customers v0, etc.
- Claude code generation
- Multiple Review agents changing code
- Dev agents change code and deploy to production
- Creating Agents for customer workflows
- ...

## Insurance Company

- Azure Copilot, Microsoft Office Copilot,
- Copilot Studio, GitHub Copilot
- **Review bot**
- Release notes
- Vibe coding prototypes
- Document generation
- Design system ( MCP Figma)
- VAT check
- Spec Kit
- Epic AI assistant
- ...



# Insurance Company Deep Dive

Let's explore how the insurance company implemented the AI experiment: the automated review bot.



# Open AI Code Review

[Aidan Cole](#) |  939 installs | ★★★★★ (0) | Free

Integrate Open AI for automated code reviews in Azure DevOps Pull Requests. Enhance code quality and improve review efficiency.

Get it free

## Simple Idea: A Review Agent

### Step 1: Initial Exploration

#### OpenAI Plugin

OpenAI key needed

- Risky with data?
- Output OK-ish

```
20 self.settings = settings
21 self.azure_devops = AzureDevOpsService(settings)
22 self.openai = OpenAIService(settings)
23 self.git = GitService(settings)
24
25 def run_review(self) -> bool:
26     """Run AI review of the PR."""
27     print("Starting AI review of the PR...")
28
29     try:
30         # Get PR information
31         # Check for duplicate comments if enabled
32         if self.settings.prevent_duplicate_comments:
33             if self.azure_devops.has_ai_reviewed_pr(self.settings.review_title):
34                 print("AI review already exists in the PR. Skipping review.")
35                 return True
36
37         # Get repository path and diff information
```

# Simple Idea: A Review Agent

## Step 2: Enterprise Solution

### We built an Azure Open AI - Tailored to our insights

- Rights already available
- Cutoff possible (data risk/costs)
- Output OK-ish
- Works in the same repo

Build Validation 2

Validate code by pre-merging and building pull request changes.

Filter



Enabled	Name ↑	Path filter	Trigger	Inheritance
<input checked="" type="checkbox"/> On	<b>Delaonline - Continuous Integration</b> Required		Automatic Expires when code changes	
<input checked="" type="checkbox"/> On	<b>PR Bot pipeline</b> Optional		Automatic Never Expires	

Status Checks 1

Require other services to post successful status to complete pull requests.

Filter



Enabled	Name ↑	Path filter	Inheritance
<input checked="" type="checkbox"/> On	<b>Sonar Cloud Quality Status</b> Required		



# Review Agent Implementation

## Checks

The displayed list of checks may be truncated for optimal performance. For a smoother experience, please maintain a reasonable number of policies (fewer than 100).



### Required

**Delaonline - Continuous Integration**   
Build succeeded

**Work items must be linked**   
Succeeded

**Comments must be resolved**   
Succeeded

### Optional

**PR Bot pipeline**  
Build succeeded

**[delanl\_online\_libraries] Quality Gate passed**  
Succeeded

**Quality Gate passed**  
Succeeded

Online Build Service (delanl)

Friday

Resolved

**Azure OpenAI Bot - Review**

File: `libs/delanl/feature-dup-application/src/components/coverageCalculatorSlideIn/CoverageCalculatorSlideIn.styled.ts`

- `width: 100%` might cause layout issues on smaller screens; verify if this change is intentional.
- Duplicate padding definitions in `theme.breakpoints.up('sm')` can lead to maintenance issues.

Write a reply...

Reactivate

# Review Agent in Action

delanl / PipelineToolbox / Repos / Files / pt-pr-ai-bot



Search

P

+

Contents

History

Name ↑	Last change	Commits
 pr-ai-bot-template.yml	Tuesday	<a href="#">fc7b82cb</a> Updated pr-ai-bot-template.yml Pascal Dufour
PY pr-ai-bot.py	Friday	<a href="#">adac301e</a> Another fix for the repo path Danny Scheelings
MD README.md	Tuesday	<a href="#">0ce20956</a> Readme updated. Repo alias in template changed Danny Scheelings
 requirements.txt	Friday	<a href="#">842bf833</a> 1st version of PR AI bot script Danny Scheelings

### PR AI bot for Azure DevOps

This repository contains a reusable Azure DevOps pipeline that automatically performs an AI-powered code review on each pull request. It uses OpenAI (via Azure OpenAI) to generate a concise review and summary based on the diff.

#### Features

- Inspects changed files in the PR
- Performs AI review using OpenAI
- Automatically posts a summary and/or review comment to the PR
- Reusable across multiple repositories via pipeline templates

---

#### Structure

```
ai-pr-bot/  
├── pr-ai-bot-template.yml # Reusable pipeline template  
├── pr-ai-bot.py           # Python script to invoke AI  
├── requirements.txt       # Python libraries  
└── README.md
```

#### How to use in another repository

##### 1. Define your PR AI bot variables

```
variables:  
  AI_SYSTEM_PROMPT_SUMMARY: |  
    You are an AI assistant that summarizes code changes in a pull request.  
    Provide a concise summary, using bullet points, of the overall changes and their impact, but keep it short.  
  AI_SYSTEM_PROMPT_REVIEW: |  
    Your task is to act as a code reviewer of a Pull Request  
  PREVENT_DUPLICATE_COMMENTS: true  
  SCRIPT_OPTION: ''
```

Online

Overview

Boards

Repos

Files

Commits

Pushes

Branches

Tags

Pull requests

Advanced Security

OS Online Build Service (delanl) 4h ago

Azure OpenAI Bot - Summary

- Added Accessibility Test:** Introduced `accessibilityDashboard.spec.ts` for dashboard accessibility validation using Axe, ensuring no serious or critical violations.
- Refactored Code:** Removed redundant accessibility checks in `dashboardDup.spec.ts` and updated `dashboardPage.ts` for improved selector handling.
- UI/UX Enhancements:** Updated styles in `MijnDeLaHeader.module.scss` and replaced icons with `EditIcon` for consistency. Improved button and navigation styles.
- Design System Updates:** Enhanced `OutlinedCard` and `KeyValueList` components with additional props and modernized styles.
- Dependency Updates:** Integrated new design system styles and components for uniformity.
- Bug Fixes:** Addressed minor issues like `autoComplete` and improved semantic HTML usage.

MR Write a reply... Reactivate

OS Online Build Service (delanl) 4h ago

Azure OpenAI Bot - Review

File: `apps/mijn-dela-e2e/src/e2e/pre-deploy/dashboard/accessibilityDashboard.spec.ts`

- Bug:** `loginPage().setAccount` assumes `loginPage` returns a singleton. Ensure it handles state correctly.
- Bug:** `dashboardPage(page).visit` and `dashboardPage(page).verifyVisit` are invoked twice. Optimize reuse.
- Performance:** `new AxeBuilder({ page }).analyze()` could be slow. Consider caching or batching scans.

File: `apps/mijn-dela-e2e/src/e2e/pre-deploy/dashboard/dashboardDup.spec.ts`

- Accessibility test removal may lead to undetected issues; consider retaining or relocating it.

File: `apps/mijn-dela-e2e/src/pages/dashboardPage.ts`

- The change to `aria-labelledby="Polisnummer"` introduces a starts-with selector. Ensure it doesn't unintentionally match multiple elements or cause false positives.

File: `apps/mijn-dela/components/layouts/MijnDeLaHeader.module.scss`

- `$section-elevation` and `var(--dds-background-page)` usage may cause inconsistent styling across themes.
- Removing `font-size: 0.875em` from `.mijnDeLaHeader__icon--edit` might break visual consistency.
- Ensure `var(--dds-typography-modest)` and other variables are defined and compatible with `$dela-blue--1_0`.

File: `apps/mijn-dela/components/layouts/MijnDeLaHeader.tsx`

- `EditIcon` usage lacks `aria-hidden` where appropriate; may impact accessibility.
- Inline `sx` styles for `Link` could cause performance issues; consider moving to CSS.
- `window.origin` usage might fail in older browsers; ensure compatibility or fallback.
- `nav` element wrapping tabs is correct but ensure proper ARIA roles for accessibility.

File: `libs/design-system/src/components/cards/outlinedCard/OutlinedCard.tsx`

- component prop default value:** Ensure the `component` prop default ( `'section'` ) doesn't conflict with accessibility or styling expectations.
- Dynamic `sx` prop for `CardContent` :** Validate `contentBackgroundColor` to prevent invalid CSS values or runtime errors.



## Development Company Deep Dive

# Review Agent Implementation

Automatic Copilot code review

Copilot will automatically review your pull requests. [Learn more.](#)

Select an option ▾

Enabled

Copilot automatically reviews pull requests you authored from your repositories.

No policy

Copilot won't enforce automated reviews.



# Review Agent Implementation

- ☒ **Require a pull request before merging**  
Require all commits be made to a non-target branch and submitted via a pull request before they can be merged.

Hide additional settings ^

## Required approvals

0 ▾

The number of approving reviews that are required before a pull request can be merged.

- ☐ **Dismiss stale pull request approvals when new commits are pushed**  
New, reviewable commits pushed will dismiss previous pull request review approvals.
- ☐ **Require review from specific teams** Private preview  
A collection of reviewers and associated file patterns. Each reviewer has a list of file patterns which determine the files that reviewer is required to review.
- ☐ **Require review from Code Owners**  
Require an approving review in pull requests that modify files that have a designated code owner.
- ☐ **Require approval of the most recent reviewable push**  
Whether the most recent reviewable push must be approved by someone other than the person who pushed it.
- ☐ **Require conversation resolution before merging**  
All conversations on code must be resolved before a pull request can be merged.
- ☒ **Request pull request review from Copilot** Preview  
Automatically request review from Copilot for new pull requests, if the author has access to Copilot code review.

## Allowed merge methods Preview

Merge, Squash, Rebase ▾

When merging pull requests, you can allow any combination of merge commits, squashing, or rebasing. At least one option must be enabled.

QaS

github.com

rb2-bv / MCS-Commerce

Q

Type to search

<> Code

Issues 2

**Pull requests 1**

Actions

Projects

Wiki

Security

Insights

fix: Fix cart quantity race condition with simplified debounce #1093

<> Code

Merged

boydenhartog merged 2 commits into main from fix-cart-quantity on Jul 7

Conversation 7

Commits 2

Checks 5

Files changed 1

+42 -30

boydenhartog commented on Jul 7

Summary

- Fixed race condition in cart quantity updates that caused incorrect values when rapidly clicking
- Simplified state management from complex ref-based approach to cleaner React patterns
- Ensured only the final quantity update is sent to server when making rapid changes

The Problem

When rapidly updating cart quantities (e.g., clicking from 55 to 50), the final value would be incorrect (e.g., 51) due to:

- Multiple debounced calls executing with intermediate values
- Race conditions between local state updates and server responses
- Complex state synchronization logic causing circular dependencies

The Solution

Implemented a cleaner approach that:

- Uses standard React `useState` for immediate UI updates
- Leverages existing `debounce` utility (no manual timeout management)
- Tracks server quantity separately to prevent overwriting user input

Reviewers

cursor[bot]

+1 more reviewer

gemini-code-assist[bot]

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

QaS

<


>

github.com

Merged

fix: Fix cart quantity race condition with simplified debounce #1093

boydenhartog merged 2 commits into `main` from `fix-cart-quant...` on Jul 7



gemini-code-assist (bot) reviewed on Jul 7

View reviewed changes

gemini-code-assist (bot) left a comment

### Summary of Changes

Hello @boydenhartog, I'm Gemini Code Assist! I'm currently reviewing this pull request and will post my feedback shortly. In the meantime, here's a summary to help you and other reviewers quickly get up to speed!

This pull request addresses a critical race condition in the cart quantity update functionality by refactoring the `EditItemQuantity` component. The primary goal is to ensure accurate and reliable quantity updates, even during rapid user interactions, by simplifying state management and improving the debouncing mechanism for server calls. This leads to a more robust and user-friendly experience for managing cart items.

### Highlights

- Race Condition Fix:** This pull request resolves a race condition in the cart quantity update mechanism that previously led to incorrect final values when users rapidly adjusted item quantities. The fix ensures that only the final, intended quantity is sent to the server.
- Simplified State Management:** The component's state management has been significantly simplified. It transitions from a complex ref-based approach (removing `isUpdatingRef`, `timeoutRef`, and `pendingQuantityRef` as mentioned in the PR description) to a cleaner pattern utilizing React's `useState` for immediate UI updates and a single `useRef` (`lastServerQuantity`) to track the server's last confirmed quantity.
- Streamlined Debouncing:** The custom or implicitly managed debounce logic has been replaced with a more explicit and cleaner use of an existing `debounce` utility. This ensures that server updates for quantity changes are properly debounced, preventing excessive network requests and race conditions.
- Improved UI Responsiveness:** By introducing `localQuantity` managed by `useState`, the UI now provides immediate visual feedback to the user as they change quantities, while the actual server update is handled asynchronously via the debounced call.
- Code Reduction and Clarity:** The changes result in a more concise and readable component, reducing the line count from approximately 100 to 80 lines, and establishing a clearer separation between UI state and server-synced state.

> Using Gemini Code Assist



QaS

<

>

github.com

Merged

fix: Fix cart quantity race condition with simplified debounce #1093

boydenhartog merged 2 commits into `main` from `fix-cart-quanti...` on Jul 7

gemini-code-assist bot left a comment

...

## Code Review

The pull request simplifies cart quantity updates and resolves a race condition. However, error handling in `debouncedUpdate` is missing, and the input field's value should consistently reflect the local state. Addressing these points will improve robustness and user experience.

frontend/components/client/EditItemQuantity/EditItemQuantity.tsx

Comment on lines +53 to 58

53

+

`const debouncedUpdate = useCallback(`

54

+

`debounce((quantity: string) => {`

55

+

`startTransition(async () => {`

56

+

`const numericQuantity = quantity ? Number(quantity) : 1`

57

+

`await performUpdate(numericQuantity)`

52

58

`})`

gemini-code-assist bot on Jul 7

...

High Priority

The `debouncedUpdate` function doesn't handle potential errors from `performUpdate`. If `performUpdate` fails (e.g., due to a network error), the UI won't reflect the actual cart state, violating the style guide's rule on graceful handling of network failures.<sup>[1]</sup> Additionally, when the input field is cleared, the quantity defaults to 1 instead of 0, which is not the desired behavior.

```
debounce((quantity: string) => {
  startTransition(async () => {
    const numericQuantity = quantity ? Number(quantity) : 0;
    const res = await performUpdate(numericQuantity);
```


QaS

github.com

merged

fix: Fix cart quantity race condition with simplified debounce #1093

boydenhartog merged 2 commits into `main` from `fix-cart-quant...` on Jul 7

 cursor bot reviewed on Jul 7

View reviewed changes

cursor bot left a comment

⋮


• Bug: API Response Validation and State Tracking Flaws

The `EditItemQuantity` component can display an incorrect quantity due to incomplete API response validation and stale state tracking. The `performUpdate` function only checks for `res?.redirectUrl` and assumes success, failing to validate `res?.success`. If the `updateItemQuantity` API returns an unsuccessful response without a redirect or error, the local UI state ( `localQuantity` ) becomes inconsistent with the server. Furthermore, `lastServerQuantity.current` is not updated after successful server operations, which prevents the `useEffect` from correctly detecting external quantity changes. This can lead to `localQuantity` being overwritten with stale `defaultQuantity` values, causing user input loss, incorrect UI display, and flawed error recovery.

▸ `frontend/components/client/EditItemQuantity/EditItemQuantity.tsx#L37-L56`

[Fix in Cursor](#) • [Fix in Web](#)

Was this report helpful? Give feedback by reacting with 👍 or 👎



merged

boydenhartog merged commit `bca6e25` into `main` on Jul 7

3 checks passed

View details

deleted branch

boydenhartog deleted the `fix-cart-quantity` branch 2 months ago



QaS

github.com

🔗 Merged

fix: Fix cart quantity race condition with simplified debounce #1093

boydenhartog merged 2 commits into `main` from `fix-cart-quantity` on Jul 7

• Bug: API Response Validation and State Tracking Flaws

The `EditItemQuantity` component can display an incorrect quantity due to incomplete API response validation and stale state tracking. The `performUpdate` function only checks for `res?.redirectUrl` and assumes success, failing to validate `res?.success`. If the `updateItemQuantity` API returns an unsuccessful response without a redirect or error, the local UI state ( `localQuantity` ) becomes inconsistent with the server. Furthermore, `lastServerQuantity.current` is not updated after successful server operations, which prevents the `useEffect` from correctly detecting external quantity changes. This can lead to `localQuantity` being overwritten with stale `defaultQuantity` values, causing user input loss, incorrect UI display, and flawed error recovery.

▸ `frontend/components/client/EditItemQuantity/EditItemQuantity.tsx#L37-L56`

[Fix in Cursor](#) • [Fix in Web](#)

Was this report helpful? Give feedback by reacting with 👍 or 👎

😊

🔗

boydenhartog

merged commit `bca6e25` into `main` on Jul 7

3 checks passed

View details


🔗

boydenhartog

deleted the `fix-cart-quantity` branch 2 months ago

Pull request successfully merged and closed

You're all set — the branch has been merged.



Add a comment

Agents

Editor

CHANGES

Message (⌘↵ to com...

✧

✓ Commit

▼

▼ Changes

10

↓

ARCHITECTURE.md

U

{ }

settings.json

.vscode

U

■

test.zip

st-tst-testapp-front

U

<>

new.component.html

st-t...

M

TS

CRMLogin.spec.ts

st-tst-t...

D

TS

SystemTeamTestSite.spec.ts

st-tst-t...

M

AGENT REVIEW

⚙

○ Reviewing

Found 1 Potential Issue (Diff wit...

<> ...es/new/new.component.html

Form field name changed, breaking data binding and tests

×

Fix

TS SystemTeamTestSite.spec.ts M

st-tst-testapp-front > test > pages

1

import type { Page } from

2

import { expect } from

3

import { HomePage } from

4

5

export const newPage =

6

const BASE\_URL = ho

7

//PageLocators

8

const listPageButto

9

const listPageTitle

10

const usernameTextb

11

const addressTextbo

12

const countryTextbo

13

const informationTe

14

const submitButton

15

16

//Pagefunctions

17

const goto = async

18

await page.goto(`

19

await expect( lis

Problems

Output

Debug Conso

pascaldufour@Pascals-MacBook-P

Agents

Editor

CHANGES

Message (⌘↵ to com...

✧

✓ Commit

▼

▼ Changes

10

↓

ARCHITECTURE.md

U

{ }

settings.json

.vscode

U

■

test.zip

st-tst-testapp-front

U

<>

new.component.html

st-t...

M

TS

CRMLogin.spec.ts

st-tst-t...

D

TS

SystemTeamTestSite.spec.ts

st-tst-t...

M

AGENT REVIEW

⚙

○ Reviewing

Found 1 Potential Issue (Diff wit...

<> ...es/new/new.component.html

Form field name changed, breaking data binding and tests

×

Fix

TS SystemTeamTestSite.spec.ts M

st-tst-testapp-front > test > pages

1

import type { Page } from

2

import { expect } from

3

import { HomePage } from

4

5

export const newPage =

6

const BASE\_URL = ho

7

//PageLocators

8

const listPageButto

9

const listPageTitle

10

const usernameTextb

11

const addressTextbo

12

const countryTextbo

13

const informationTe

14

const submitButton

15

16

//Pagefunctions

17

const goto = async

18

await page.goto(`

19

await expect( lis

Problems

Output

Debug Conso

pascaldufour@Pascals-MacBook-P



**Creating code** – > creating behavior.  
Proven by verifications





# Comparing Experiment Portfolios 2025

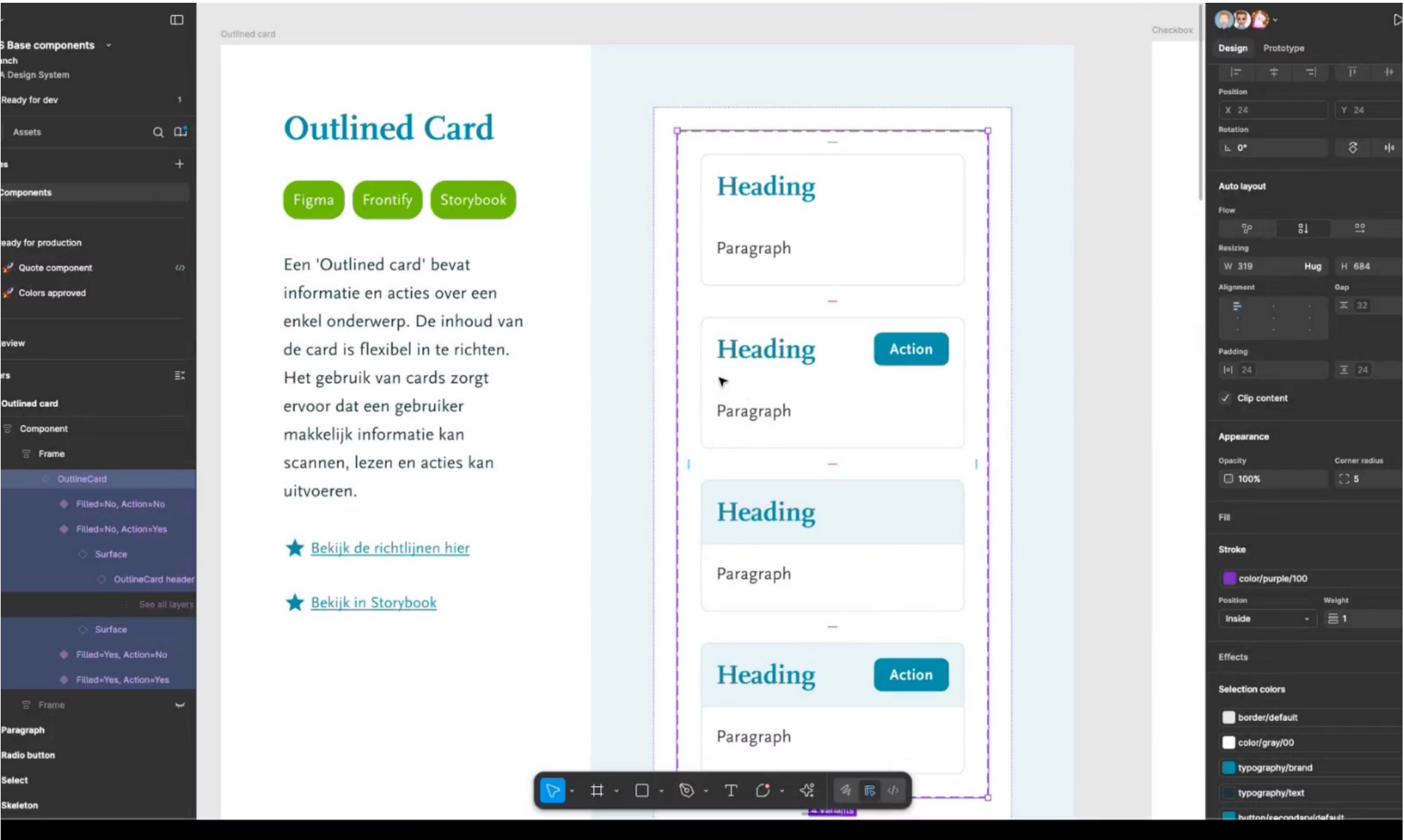
## Development Company

1. Coding assistants
2. Review bot
3. **Customer prototyping** – v0 and similar rapid prototype tools
4. **Claude code generation**
5. **Collaborative agents** – Multiple agents modifying code
6. **Dev agents change code and deploy to production**
7. **Design system integration** – MCP Figma
8. **Customer workflow agents**

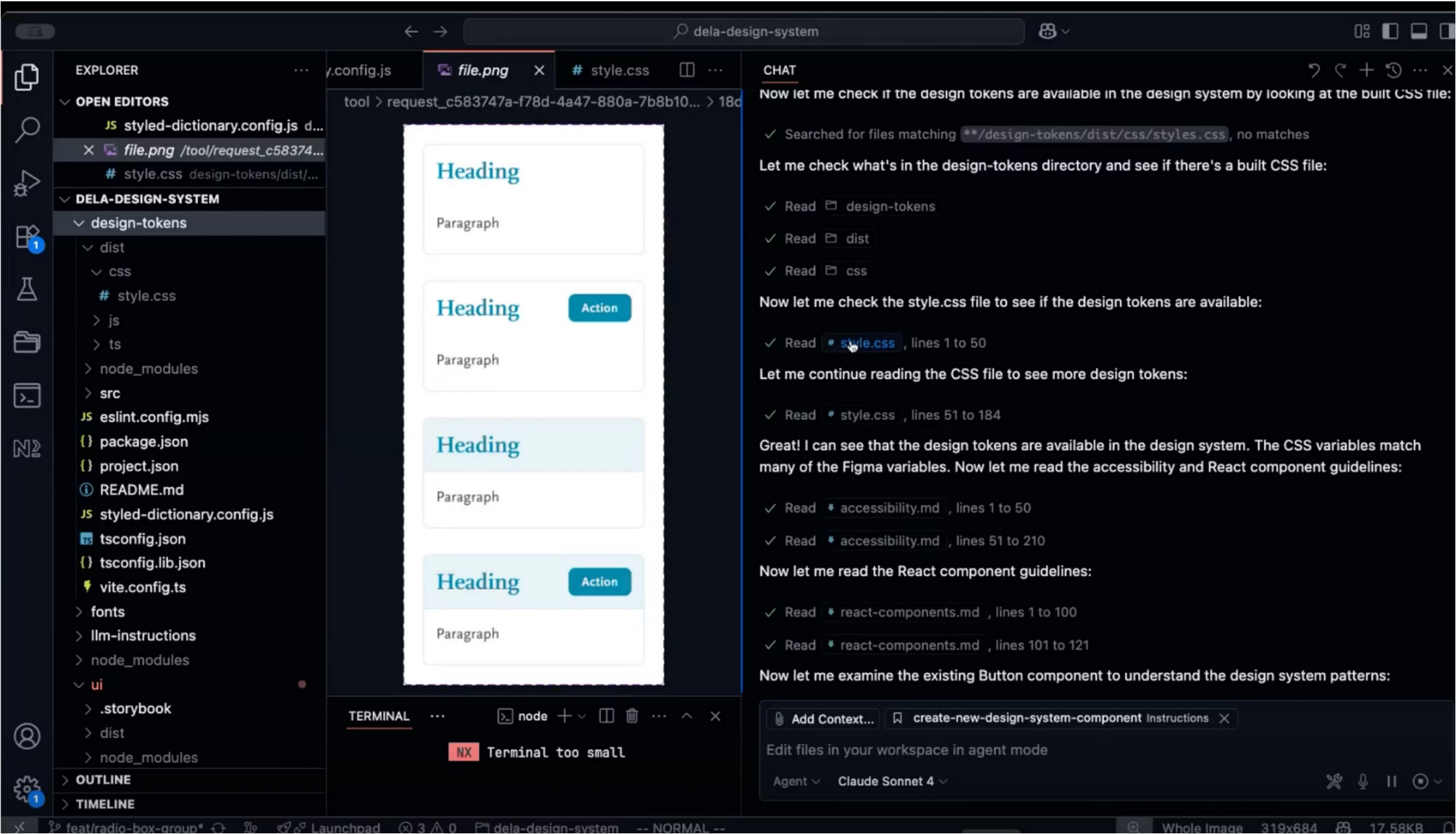
## Insurance Company

1. **Azure Copilot, Microsoft Office Copilot, Copilot Studio, GitHub Copilot**
2. **Review bot**
3. **Release notes**
4. **Vibe coding** – Rapid prototype development
5. **Document generation**
6. **Design system** – MCP Figma integration
7. **VAT checking**
8. **Epic AI assistant**

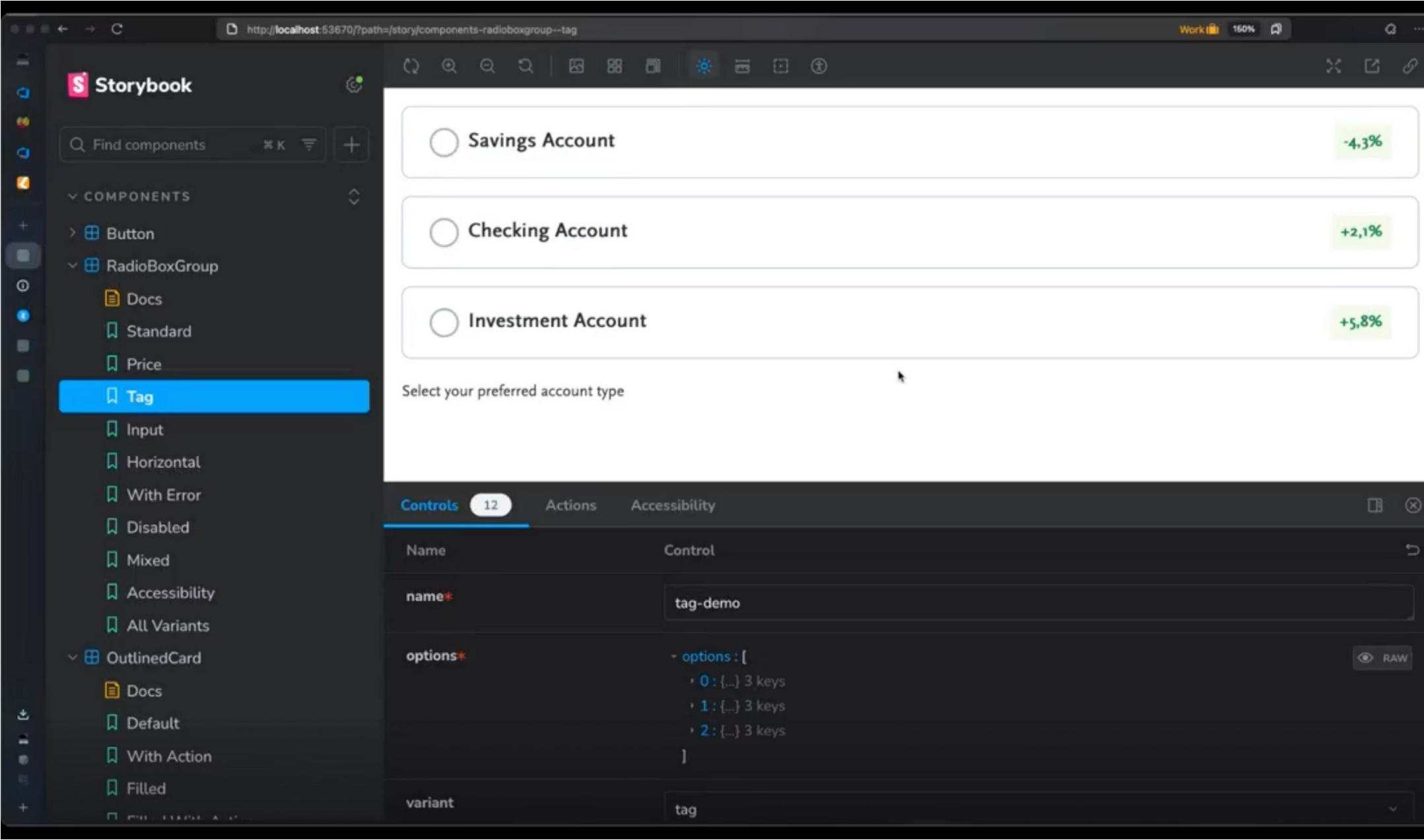
# Figma Insurance



# Figma Insurance



# Figma Insurance







## **Figma at Development Company**

# Experiments to mention

## Development Company Experiments

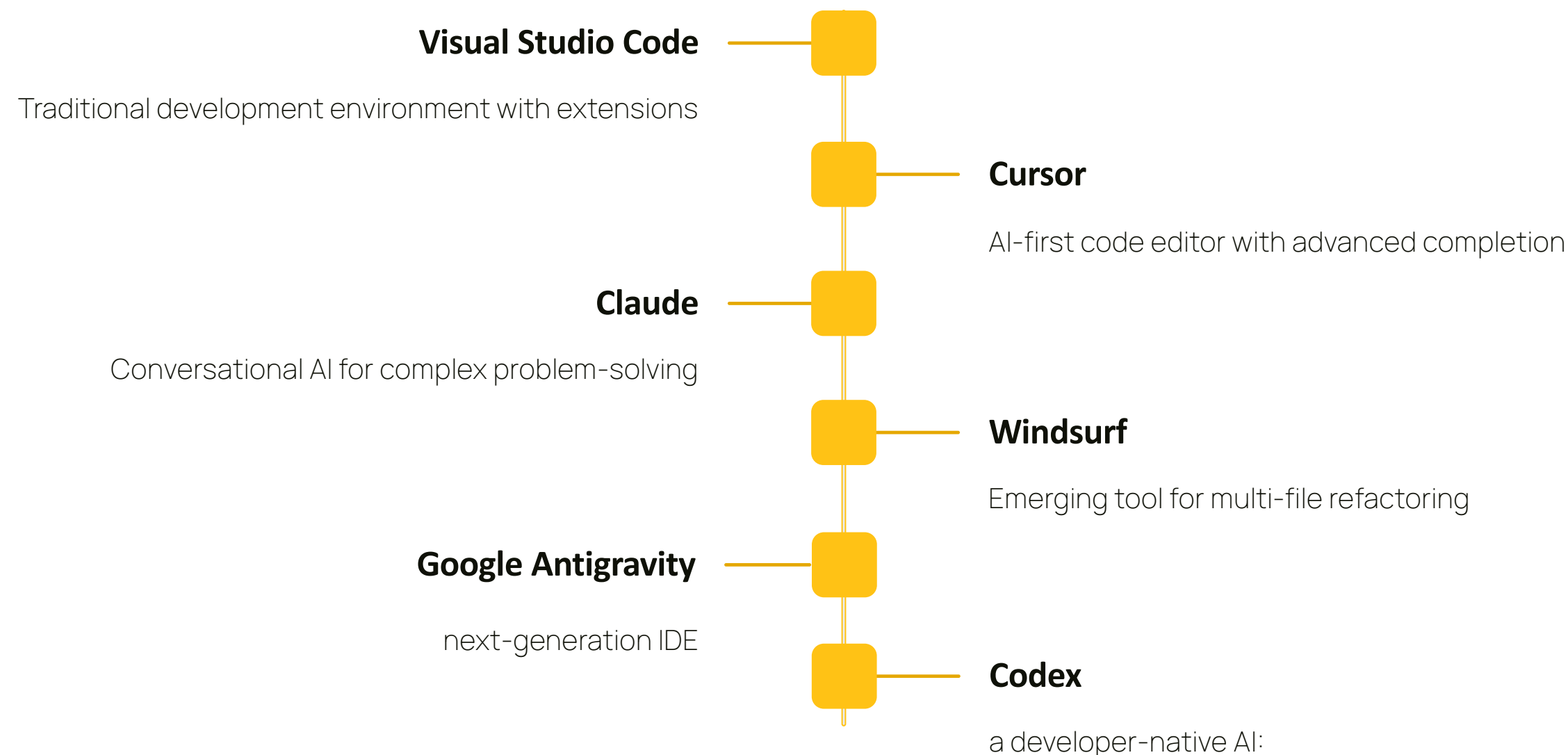
- **Coding assistants**
- Review bot
- Prototyping with customers v0, etc.
- Claude code generation
- Multiple Review agents changing code
- Dev agents change code and deploy to production
- Design system (MCP Figma)
- Creating Agents for customer workflows
- ...

## Insurance Company Experiments

- Azure Copilot, Microsoft Office Copilot, Copilot Studio,
- **GitHub Copilot**
- Review bot
- Release notes
- Vibe coding prototypes
- Document generation
- Design system ( MCP Figma)
- VAT check
- Software testing with AI
- ...

# Tool Evolution at Development Company

## Every "Week" a New Tool



# Insurance Company



## Deliberate Simplicity

- **Visual Studio Code** – Primary development environment
- **With GitHub**



# GitHub Copilot review Hype

[Meer details](#)

When using GitHub CoPilot, I...

● Strongly Disagree   ● Disagree   ● Neutral   ● Agree   ● Strongly Agree

Enjoy coding more

Write better quality code

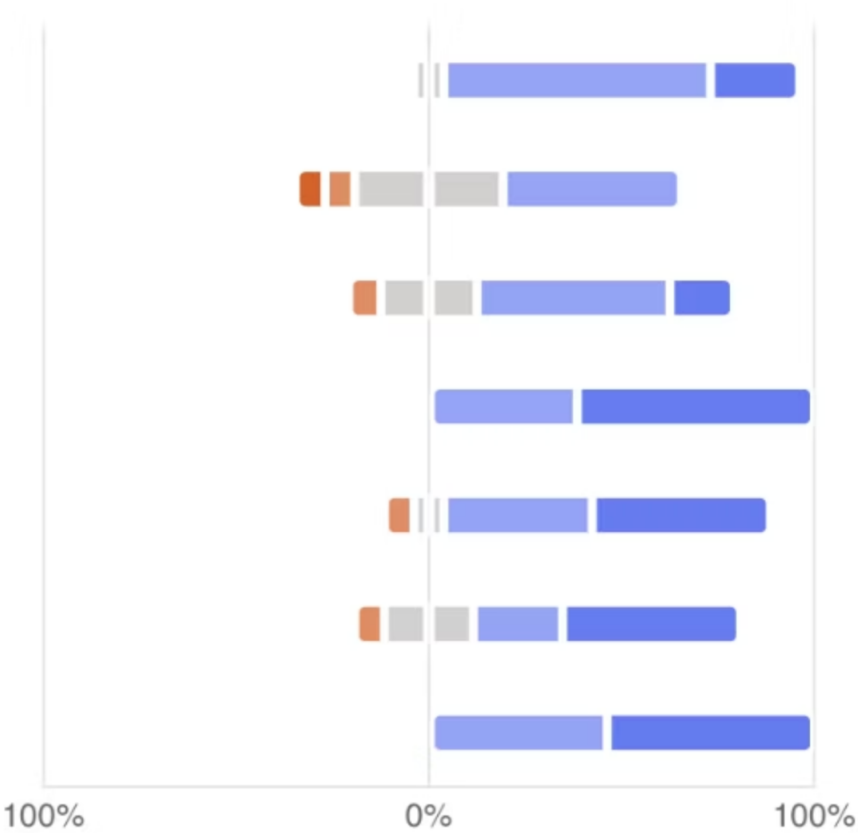
Commit code suggested by Copilot

Complete tasks faster

Learn from the suggestions Copilot shows me

Spend less time searching for information or examples

Spend less mental effort on repetitive tasks



# GitHub Copilot

## Metered usage

Get usage report

Search or filter usage

Group by: SKUs

Timeframe: Last year (2025)

### Metered usage grouped by skus

2025 (All times in UTC)

Copilot Business Copilot Premium Request




### Usage breakdown

Usage for 2025 (All times in UTC).

SKUs	Units	Price/unit	Gross amount	Billed amount
Copilot Business	392.82 license-month	\$19.00	\$7,463.59	\$7,463.59
Copilot Premium Request	8,057.7 requests	\$0.04	\$322.31	\$6.56

# GitHub Copilot: Usage Patterns

# GitHub Copilot

 **GitHub Copilot Business is active for your account**

You are assigned a seat as part of a [GitHub Copilot Business](#) subscription managed by [CooperatieDELA](#).  
Get started by installing the extension in your preferred environment.

Copilot in your IDE

Copilot in the CLI

Chat in GitHub Mobile

More features

## Usage

Premium requests

9.7%

Please note that there may be a delay in the displayed usage percentage. The premium request entitlement for your plan will reset at the start of next month. You can enable additional premium requests in the [Billing settings](#).

Managed by [CooperatieDELA](#)

## Features

<div><b>Editor preview features</b></div> <div>You can use preview features in your editor. <a href="#">Learn more.</a></div>	<div>Enabled</div>
<div><b>Copilot in GitHub.com</b></div> <div>You can use Copilot Chat in GitHub.com and Copilot for pull requests. Copilot code review and preview features are only available for paid licenses. <a href="#">Learn more about Copilot in GitHub.com.</a></div>	<div>Enabled</div>
<div><b>Copilot in the CLI</b></div> <div>You can use <a href="#">GitHub Copilot for assistance in terminal.</a></div>	<div>Enabled</div>
<div><b>Copilot Extensions</b></div> <div>GitHub Copilot Extensions are not available for use in your account. You can contact an administrator in <a href="#">CooperatieDELA</a> to request access.</div>	<div>Disabled</div>
<div><b>Copilot in GitHub Desktop</b></div> <div>You can use <a href="#">GitHub Copilot for assistance in GitHub Desktop.</a></div>	<div>Disabled</div>
<div><b>Copilot Chat in the IDE</b></div> <div>You can use <a href="#">GitHub Copilot Chat in the IDE</a> to explain a piece of code, propose bug fixes, or generate unit tests in a chat interface from your editor.</div>	<div>Enabled</div>
<div><b>Copilot Chat in GitHub Mobile</b></div> <div>You can use <a href="#">Copilot Chat in GitHub Mobile</a> personalized to a codebase.</div>	<div>Enabled</div>

Copilot Usage

Code completions

Included

Chat messages

Included

Premium requests

100%

Additional paid premium requests disabled.

Allowance resets August 1, 2025.

Workspace Index

Locally indexed

Code Completions

☒ All files

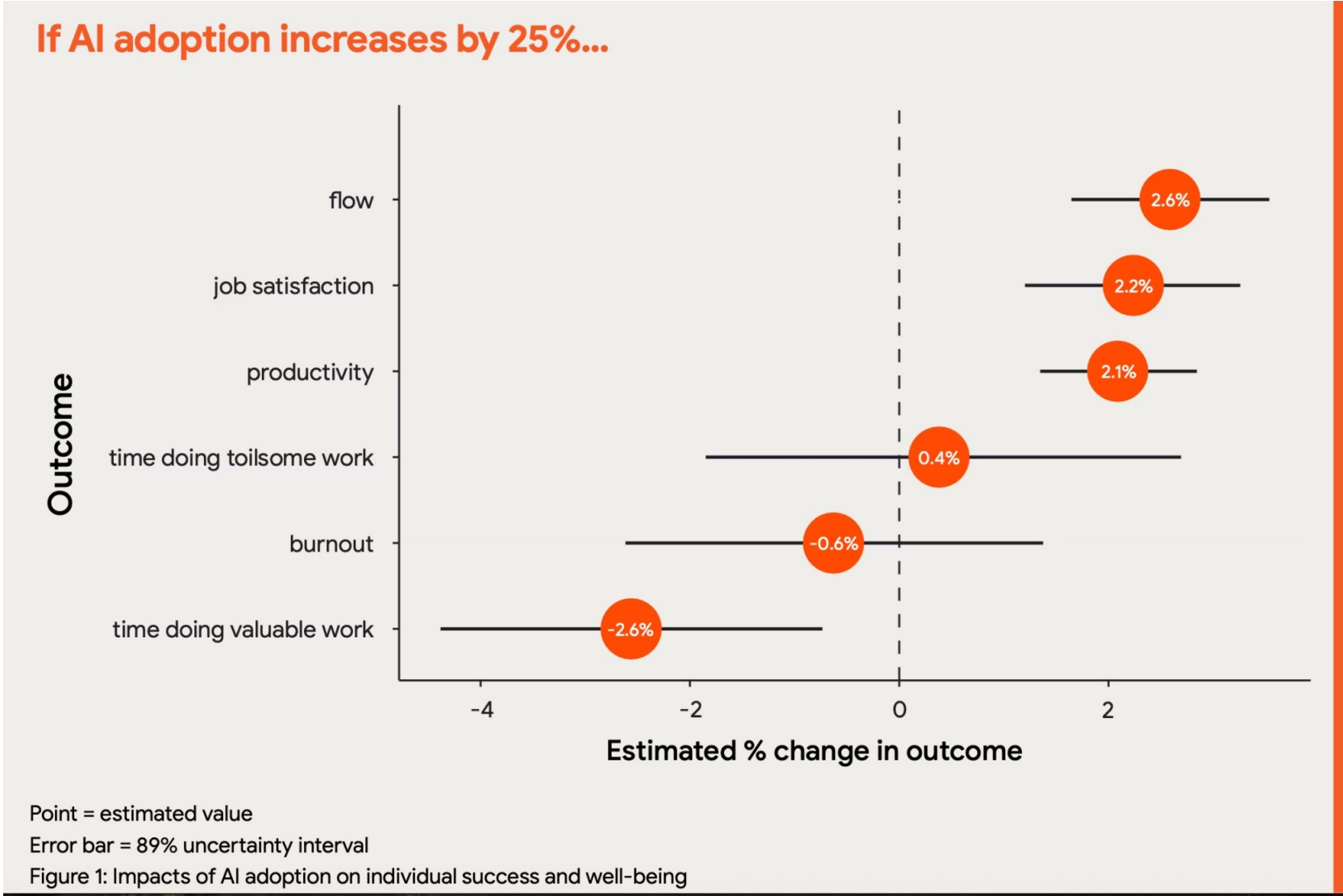
☐ Next edit suggestions

Snooze

Hide completions for 5 min

# Are We Normal?

## DORA Metrics Comparison (2025)





# Are We Normal?

## McKinsey Research: Superagency in the Workplace

*Published January 2026*

A BIG NUMBER

# 40–50%

That’s how much faster organizations can [accelerate their timelines for modernizing IT systems, by using gen AI](#) to eliminate much of the company’s manual work. McKinsey’s Aaron Bawcom and Matt Fitzpatrick and coauthors also estimate that harnessing gen AI can help organizations cut 40 percent of costs derived from their [tech debt](#)—that is, the “tax” a company pays in time and resources needed to solve tech issues.

---

“

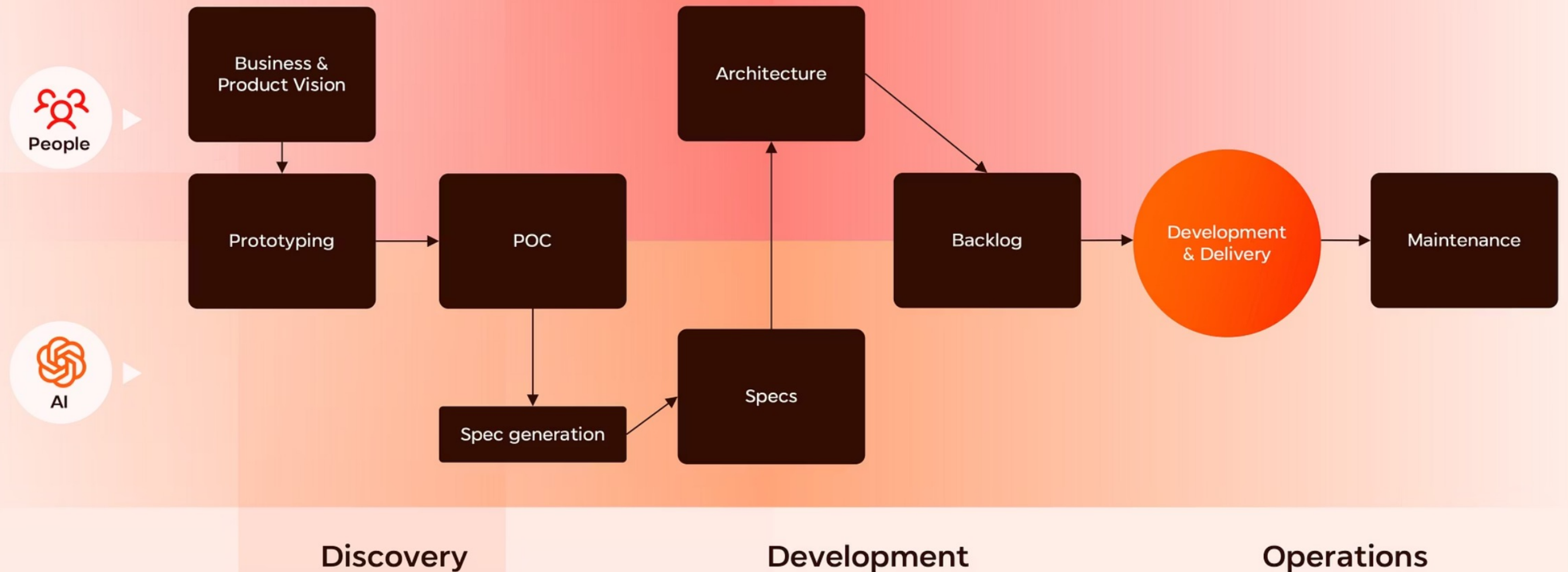
A QUOTE

“Without access to good and relevant data, this new world of possibilities and value will remain out of reach.”

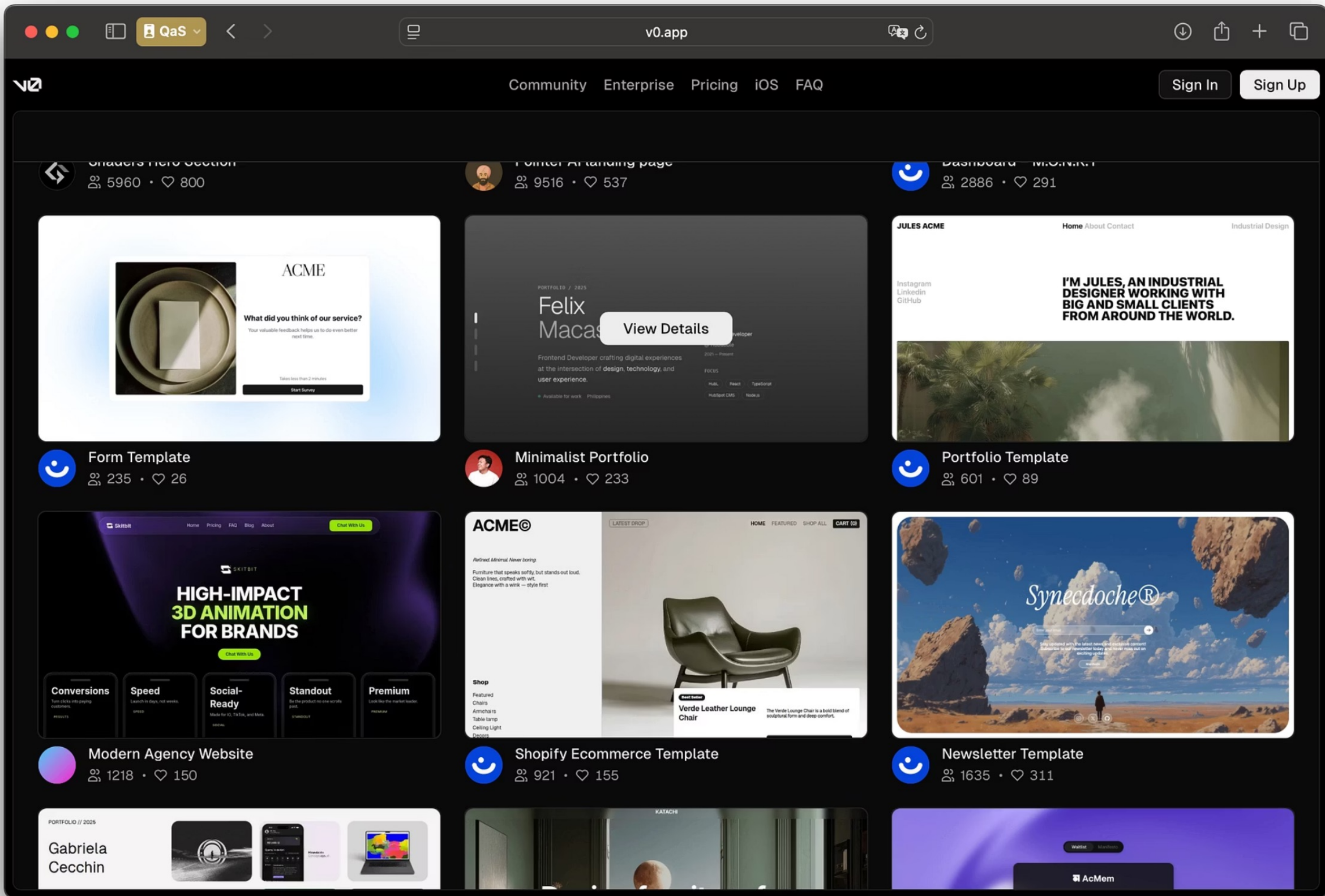
# Results in Organizational Design: Development Company



# Our approach









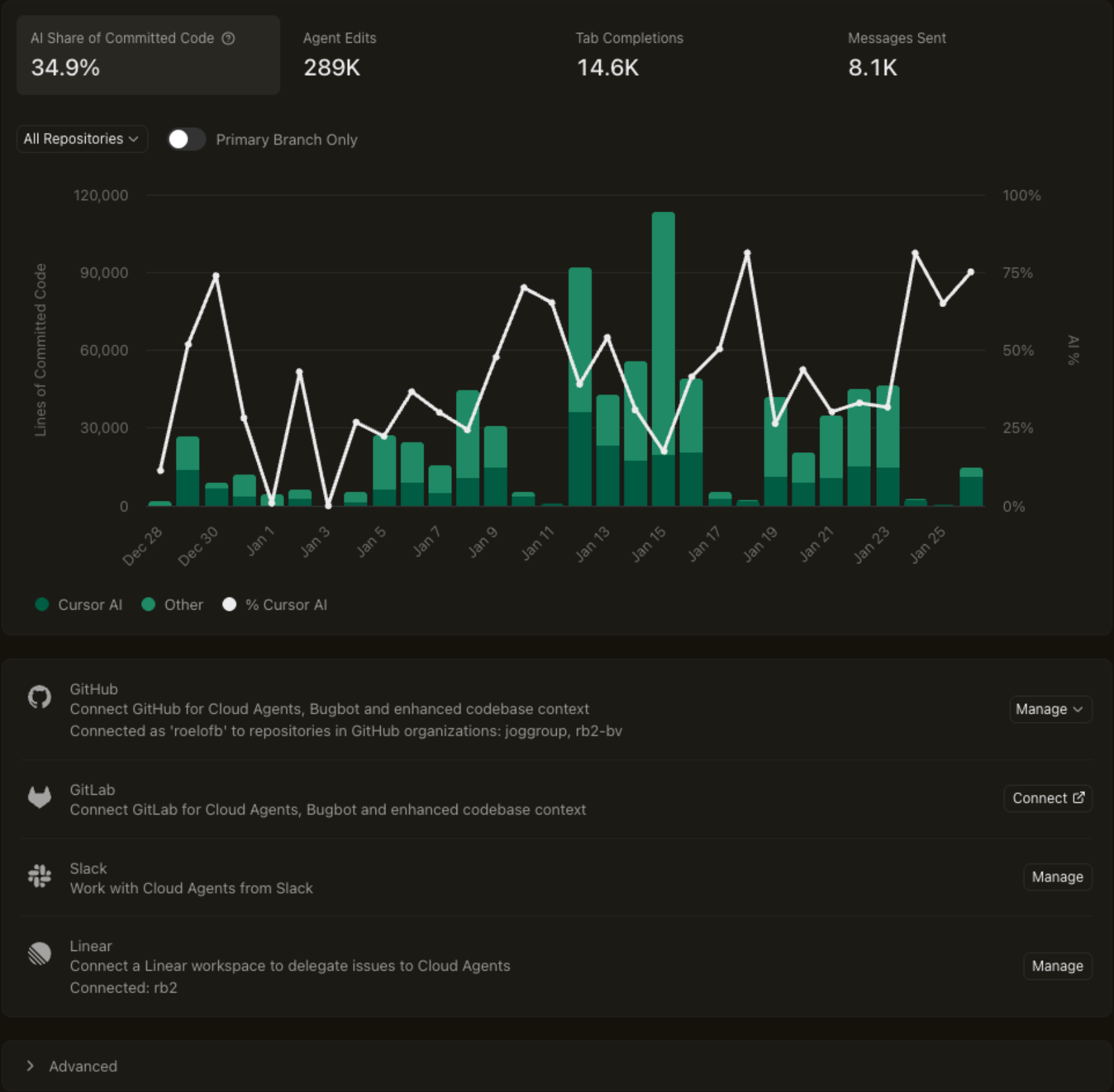
**Move to Orchestrator of agents**



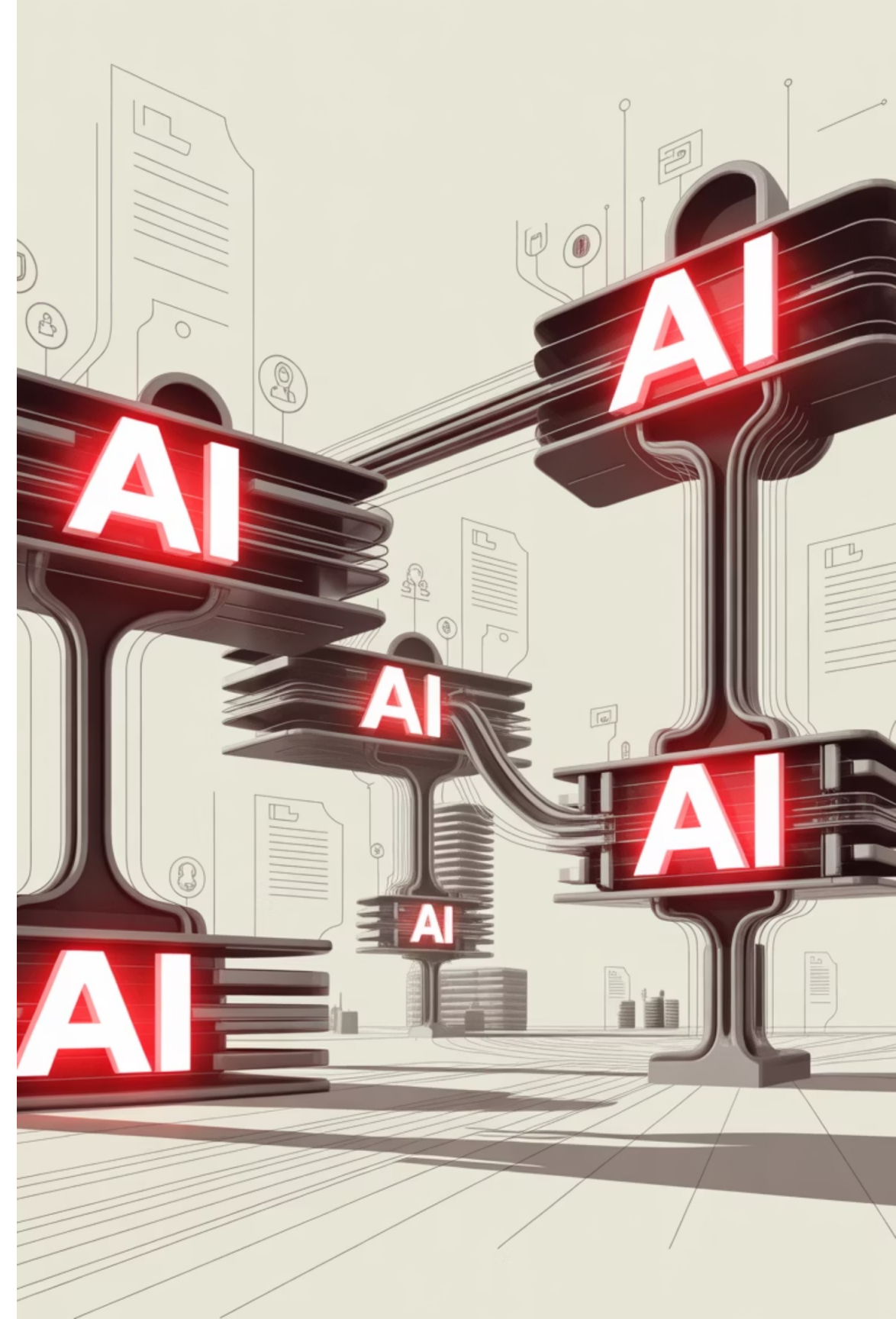
# Development Company

16% of code is generated by AI in September

34% of code is generated by AI in January

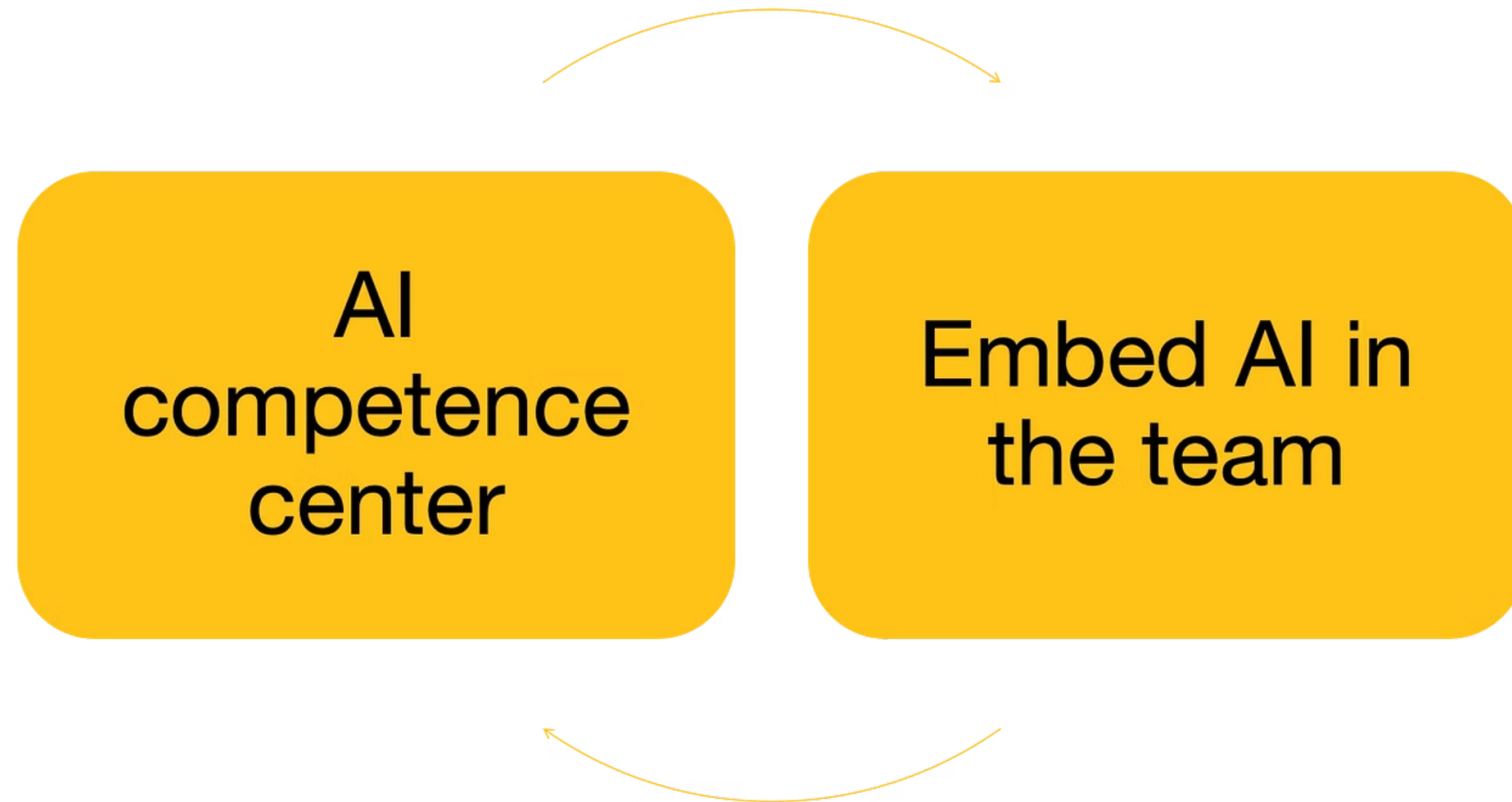


# Results in Organizational Design: Insurance Company





# Organizational Design





# Do We Need More AI Work?



I have to continue to communicate in Dutch, but we will continue with the process.

**Captured.**

---

Now we move on to the next field:

## 2. Epic Description

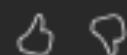
You haven't provided a description yet. This field usually contains:

- **Reason / context:** Why is this important?
- **Purpose:** What do we want to achieve?
- **Scope:** What is and is not covered?

**Proposal (example text):** "The goal of this Epic is to gain insight into the use of AI within the organization. This includes identifying where and how AI solutions are applied, measuring frequency of use, and collecting data to enable future optimizations. The scope focuses on internal AI tools and processes, not external customer applications."

*Please note that this is an illustrative example, not a definitive text. Always test this yourself.*

Do you want to capture this for this field?



9:18 pm

yes please, can you give insight how this epic will help DELA's objectives and goals



**Captured.**

---

Now the next field:



9:18 pm

yes please, can you give insight how this epic will help DELA's objectives and goals

DELA Epic AI Assistant 9:18 pm



### Captured.

Now the next field:

#### 3. Intended business results & contribution to DELA objectives

You specifically ask how this Epic contributes to DELA's goals. This field describes:

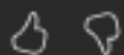
- **Which business goals are supported** (e.g.: efficiency, compliance, innovation).
- **How this Epic contributes to that.**

**Proposal (sample text):**"This Epic supports DELA's strategic objective to use technology responsibly and efficiently. By understanding AI usage, we can:

- Creating transparency about where AI is applied.
- Ensure compliance and ethical use.
- Making data-driven decisions about future investments in AI. This contributes to innovation and the improvement of internal processes, in line with DELA's ambition to use technology responsibly."

*Please note: this is an illustrative example. Always test this yourself.*

Do you want to capture this for this field?



can you explain a bit more regarding making data-driven decisions about future investments in AI. This contributes to innovation and the improvement of internal processes, in line with DELA's ambition to use technology responsibly



DELA Epic AI Assistant 9:19 pm ㊦



Good question! I will explain this further:

**Why data-driven decisions are important:**

- **Insight into usage patterns:** By measuring where and how AI is used, we get hard data on adoption, effectiveness and impact.
- **Substantiated investments:** Instead of investing based on assumptions, we can prioritize AI solutions that demonstrably add value.
- **Resource Optimization:** Data shows which AI applications are little used or do not produce the desired result, so that we use resources more efficiently.
- **Driving innovation:** By understanding which AI solutions contribute the most to process improvement, we can innovate in a more targeted way.
- **Responsible use:** Measuring helps to check whether AI applications meet ethical and compliance requirements, which is in line with DELA's ambition to use technology responsibly.

**How this contributes to DELA's goals:**

- **Innovation:** By understanding success factors, we can develop new AI initiatives that really add value.
- **Process improvement:** Data shows where AI accelerates or improves processes, so that we can scale up these successes.
- **Responsibility:** Transparency in AI use supports governance and compliance.

Would you like me to incorporate this extensive explanation in the field "**Intended business results & contribution to DELA objectives**" and present it again as a proposal?



# Experiments

## Do they help?

### Development Company

- AI Bot (Agent Style): YES
- Figma Components: MODERATE
- Customer Prototyping: YES
- GitHub Copilot: YES/MODERATE
- Workflow Agents: YES
- Vibe coding : YES

### Insurance Company

- AI Bot (Custom Version): MODERATE
- Figma Components: YES
- Copilot Studio: YES – Document checking at insurance
- GitHub Copilot: YES
- Postbot AI: YES
- Epic AI: YES
- CoPilot Studio: maintenance? we don't now yet..

# Key Takeaways

- 1** Early Adopters
- 2** AI Is Genuinely Helpful
- 3** "Vibe Coding" Accelerates POCs ....
- 4** "Standard" Code Is King
- 5** Testing is key for moving fast
- 6** AI Is a Pleaser and it is Addictive (and Costly)
- 7** Agents with SKILLS and Agentic workflows are the current experiments



**Thank you Martijn and  
the team for this journey**

**Questions?**

**Let's Continue the  
Conversation**

**Pascal Dufour**

AI development Consultancy

Agile Test Consultancy

**Email:** [pascal@qualityatspeed.nl](mailto:pascal@qualityatspeed.nl)