

Oct 17, 2025 at LeadingEng NY

Making AI work for you:

Examining and Enhancing your developers' workflows

Neha Batra @nerdneha



Oct 17, 2025 at LeadingEng NY

Making AI work for you:

Examining and Enhancing your developers' workflows

Neha Batra @nerdneha



Oct 17, 2025 at LeadingEng NY

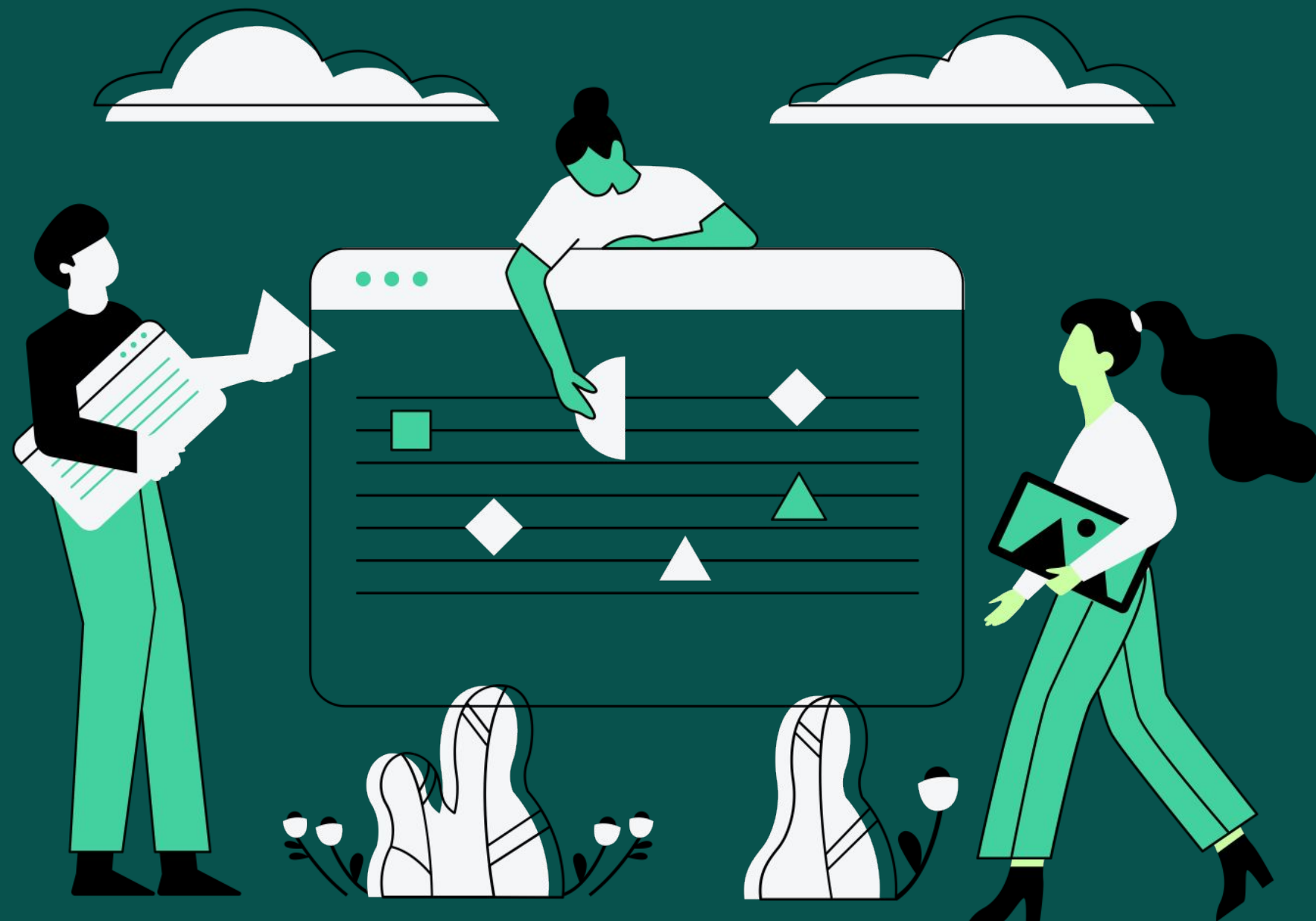
Making AI work for you:

Examining and Enhancing your developers' workflows

Neha Batra @nerdneha



To fix, you must find

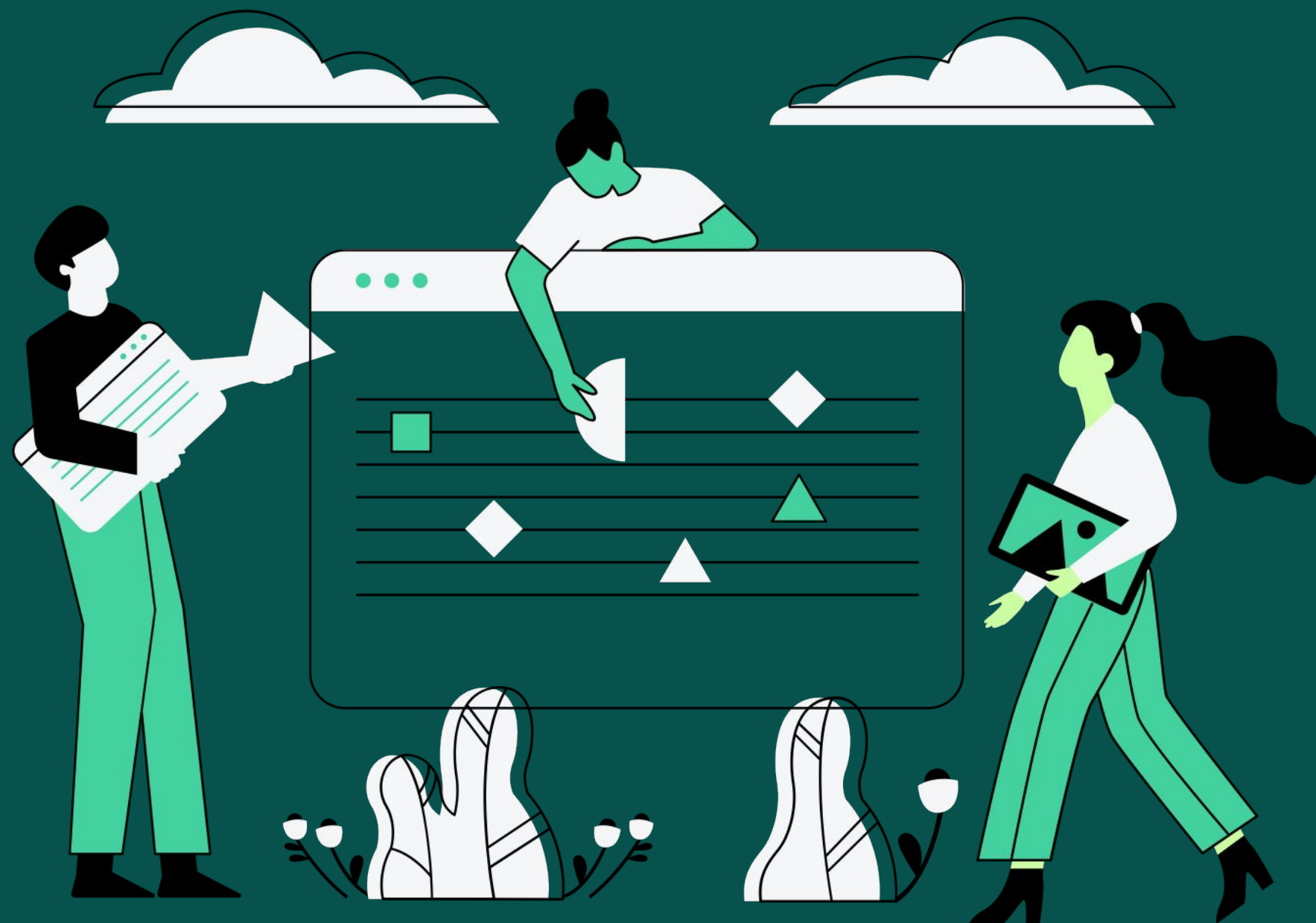


01 Identifying
your coding workflows

02 Modifying
your coding workflows

03 Measuring productivity of
your coding workflows

To fix, you must find



01 *Identifying*
Inner Loop vs. Outer Loop

02 *Modifying*
Automation vs. AI

03 *Measuring*
Flexible Time vs. Inflexible Time



01 IDENTIFYING

Inner Loop of Dev Workflows

- **IDEA to PICK UP**
 - Persona: Spec-er
 - capture, create, break down, retrieve info, add context, assign
- **PICK UP to CODE**
 - Persona: Coder
 - pick up, understand context, plan files to touch, write code, test, run, organize pull request
- **CODE to CLOUD**
 - Persona: Reviewer, Merger/Releaser
 - assign, read, run, test, review, improve, merge, ship



Inner Loop of Dev Workflows

Stage	Tasks
Idea to Pick up Persona: "Spec-er" (ex: Product Manager)	Capture an idea
	Create Epic
	Break Epic into tasks & subtasks
	Retrieve missing info before pick up
	Add helpful context (patterns, optimizations, old code gotchas, etc)
	Assign issue
Pick up to Code Persona: "Writer" (ex: Eng IC)	Pick up the right task
	Understand necessary context
	Create a plan for files to touch
	Write code
	Test against different scenarios (integration/end-to-end testing)
	Run code
	Make PR ready (final touches)
Code to Cloud Persona: "Reviewer" and maybe "Merger" (ex: Staff Eng & Eng IC)	Assign self to PR
	Read context
	Run code
	Test
	Review
	Improve code
	Merge
	Ship



Inner Loop of Dev Workflows

Stage	Tasks
Idea to Pick up Persona: "Spec-er" (ex: Product Manager)	Capture an idea
	Create Epic
	Break Epic into tasks & subtasks
	Retrieve missing info before pick up
	Add helpful context (patterns, optimizations, old code gotchas, etc)
	Assign issue
Pick up to Code Persona: "Writer" (ex: Eng IC)	Pick up the right task
	Understand necessary context
	Create a plan for files to touch
	Write code
	Test against different scenarios (integration/end-to-end testing)
	Run code
	Make PR ready (final touches)
Code to Cloud Persona: "Reviewer" and maybe "Merger" (ex: Staff Eng & Eng IC)	Assign self to PR
	Read context
	Run code
	Test
	Review
	Improve code
	Merge
	Ship



Inner Loop of Dev Workflows

Stage	Tasks
Idea to Pick up Persona: "Spec-er" (ex: Product Manager)	Capture an idea
	Create Epic
	Break Epic into tasks & subtasks
	Retrieve missing info before pick up
	Add helpful context (patterns, optimizations, old code gotchas, etc)
	Assign issue
Pick up to Code Persona: "Writer" (ex: Eng IC)	Pick up the right task
	Understand necessary context
	Create a plan for files to touch
	Write code
	Test against different scenarios (integration/end-to-end testing)
	Run code
	Make PR ready (final touches)
Code to Cloud Persona: "Reviewer" and maybe "Merger" (ex: Staff Eng & Eng IC)	Assign self to PR
	Read context
	Run code
	Test
	Review
	Improve code
	Merge
	Ship



Outer Loop of Dev Workflows

- **ORGANIZE** the work
 - collect the right pre existing issues/features/bugs into the epic
- **VISUALIZE** the organized work
 - order along what's important to us (ex: num ppl impacted, depth of impact, ease of completion)
- **PRIORITIZE** the visualized work
 - reorder and sort out milestones and iterations
- **PLAN** the prioritized work
 - make each task as easy to pick up as possible and easy to determine if it's done right
- **COMMUNICATE** the plan
 - ensure it's easy for a developer is ready to pick up an item of work in an epic or project with the right context



Outer Loop of Dev Workflows

Tasks
Organize/Collect
Label
Prioritize
Catch up
Notify/Inform/Distribute
Delegate
Document



Outer Loop of Dev Workflows

Tasks
Organize/Collect
Label
Prioritize
Catch up
Notify/Inform/Distribute
Delegate
Document

?



INNER vs. OUTER Loop



Stage	Tasks
Idea to Pick up Persona: "Spec-er" (ex: Product Manager)	Capture an idea
	Create Epic
	Break Epic into tasks & subtasks
	Retrieve missing info before pick up
	Add helpful context (patterns, optimizations, old code gotchas, etc)
	Assign issue
Pick up to Code Persona: "Writer" (ex: Eng IC)	Pick up the right task
	Understand necessary context
	Create a plan for files to touch
	Write code
	Test against different scenarios (integration/end-to-end testing)
	Run code
	Make PR ready (final touches)
Code to Cloud Persona: "Reviewer" and maybe "Merger" (ex: Staff Eng & Eng IC)	Assign self to PR
	Read context
	Run code
	Test
	Review
	Improve code
	Merge
	Ship

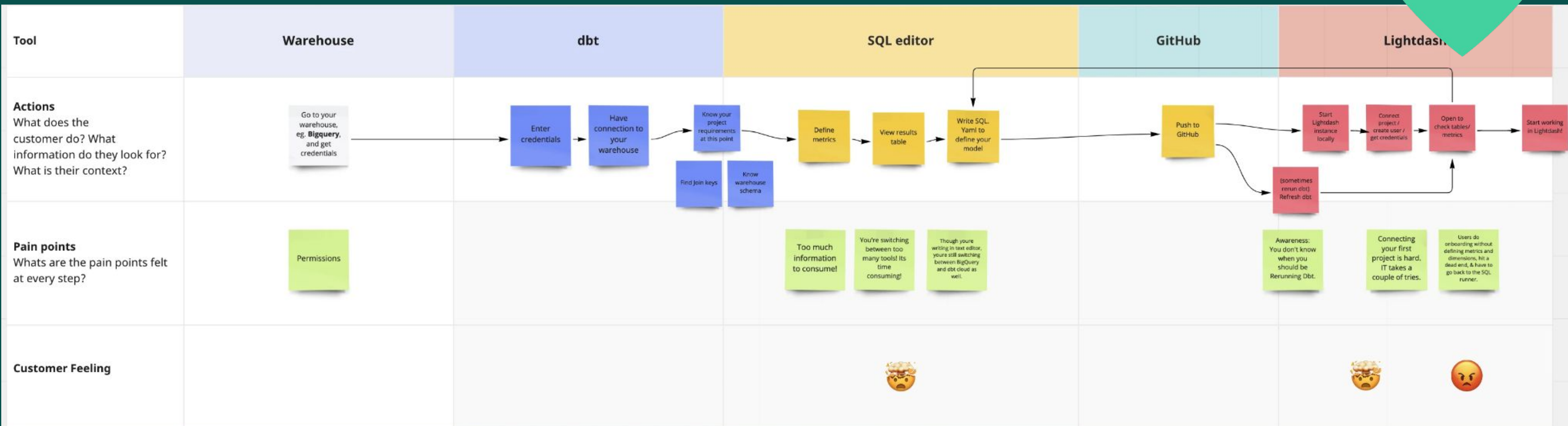
Tasks
Organize/Collect
Label
Prioritize
Catch up
Notify/Inform/Distribute
Delegate
Document

What vs. Where in Loops

Stage	Tasks	GitHub Feature
Idea to Pick up Persona: “Spec-er” (ex: Product Manager)	Capture an idea	Issues (create)
	Create Epic	Issues (create)
	Break Epic into tasks & subtasks	Issues (subtasks)
	Retrieve missing info before pick up	Issues (comments)
	Add helpful context (patterns, optimizations, old code gotchas, etc)	Issues (comments)
	Assign issue	Issues (assign)
Pick up to Code Persona: “Writer” (ex: Eng IC)	Pick up the right task	Issues (read)
	Understand necessary context	Issues (comments)
	Create a plan for files to touch	Issues (comments)
	Write code	IDE
	Test against different scenarios (integration/end-to-end testing)	IDE
	Run code	Browser
	Make PR ready (final touches)	PRs (create)
Code to Cloud Persona: “Reviewer” and maybe “Merger” (ex: Staff Eng & Eng IC)	Assign self to PR	PRs (assign)
	Read context	PRs (conversation)
	Run code	IDE
	Test	Browser
	Review	PRs (files changed)
	Improve code	IDE or PRs
	Merge	PRs (merge)
	Ship	Actions, Releases

Tasks	GitHub Feature
Organize/Collect	Projects, Issues (milestones)
Label	Issues/PRs (labels)
Prioritize	Projects, Issues (milestones)
Catch up	Notifications
Notify/Inform/Distribute	Discussions, Issues (comments), PRs (comments) + Notifications
Delegate	Issues (assignees) PRs (assignees)
Document	Discussions, Issues (comments), PRs (comments), Repos (ex: docs/)

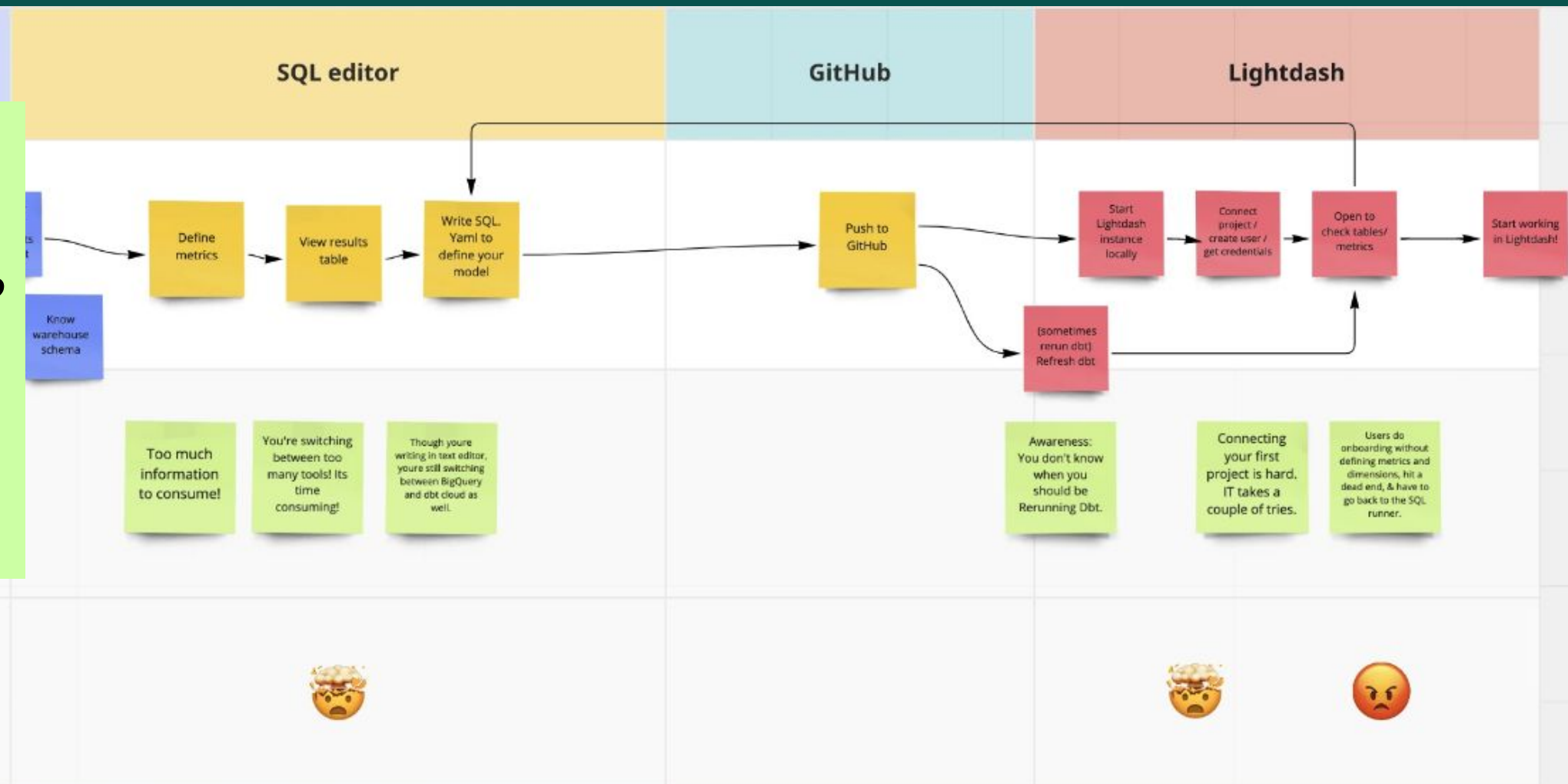
Identifying: User Journeys



Identifying: Switching Cost

Look for SWITCHING COSTS

- *When are we switching between two apps?*
- *When do you need two windows open?*
- *What requires human intervention (esp across timezones)*



Identifying: Follow the Pain

Tool	Warehouse	dbt	SQL editor	GitHub	Lightdash
Actions What does the customer do? What information do they need? What is their current workflow?	Look for SWITCHING COSTS ● When are we switching between two apps? ● When do you need two windows open? ● What requires human intervention (esp across timezones)		Follow the PAIN ● What are people most frustrated about? why? ● What requires more than 5 clicks? ● What requires you to leave and come back to the computer? ● What causes ppl to “lose face”?		
Pain points Whats are the pain points at every step?					
Customer Feeling					 

Identifying: Follow the Pain

Tool	Warehouse	dbt	SQL editor	GitHub	Lightdash
Actions What does the customer do? What information do they need? What is their goal?	Look for SWITCHING COSTS ● When are we switching between two apps? ● When do you need two windows open? ● What requires human intervention (esp across timezones)		Follow the PAIN ● What are people most frustrated about? why? ● What requires more than 5 clicks? ● What requires you to leave and come back to the computer? ● What causes ppl to “lose face”?		
Pain points Whats are the pain points at every step?					
Customer Feeling					 



02

MODIFYING

Automation vs. AI



Automation

- Deterministic
- “known transform”
- Fast without thinking
- Low context
- **FREE**



AI

- Non-Deterministic
- “unknown transform”
- Can afford time
- Requires context
- **LESS FREE**



Automation vs. AI



Automation

**HARD CODE
WHEN YOU
CAN**



AI

- Non-Deterministic
- “unknown transform”
- Can afford time
- Requires context
- **LESS FREE**



Automation vs. AI



Automation

**HARD CODE
WHEN YOU
CAN**



AI

**SKIP
MULTIPLE
STEPS**



Thinking **Small** / Locally

Tasks	GitHub Feature	AI Feature Ideas
Organize/Collect	Projects, Issues (milestones)	<i>Suggesting epic groupings</i>
Label	Issues/PRs (labels)	<i>Suggest default labels or go ahead and “apply the right labels” as a first round</i>
Prioritize	Projects, Issues (milestones)	<i>Suggest dependencies and prioritize backlog based on dependencies</i>
Catch up	Notifications	<i>“While you were gone” on your issue or project board</i>
Notify/Inform/Distribute	Discussions, Issues (comments), PRs (comments) + Notifications	<i>If an individual is consuming from a project board and priorities change, propagating that priority shift to the place they consume</i>
Delegate	Issues (assignees) PRs (assignees)	<i>Suggest assignees based on who has typically completed these tasks</i>
Document	Discussions, Issues (comments), PRs (comments), Repos (ex: docs/)	<i>Suggest cross-posting to an epic for a big decision within a sub issue</i>

Thinking **Big** / Globally

Tasks	GitHub Feature	AI Feature Ideas
Organize/Collect	Projects, Issues (milestones)	<i>Suggesting epic groupings</i>
Label 	Issues/PRs (labels)	<i>Suggest default labels or go ahead and “apply the right labels” as a first round</i>
Prioritize 	Projects, Issues (milestones)	<i>Suggest dependencies and prioritize backlog based on dependencies</i>
Catch up	Notifications	<i>“While you were gone” on your issue or project board</i>
Notify/Inform/Distribute	Discussions, Issues (comments), PRs (comments) + Notifications	<i>If an individual is consuming from a project board and priorities change, propagating that priority shift to the place they consume</i>
Delegate	Issues (assignees) PRs (assignees)	<i>Suggest assignees based on who has typically completed these tasks</i>
Document	Discussions, Issues (comments), PRs (comments), Repos (ex: docs/)	<i>Suggest cross-posting to an epic for a big decision within a sub issue</i>

Thinking **Big** / Globally

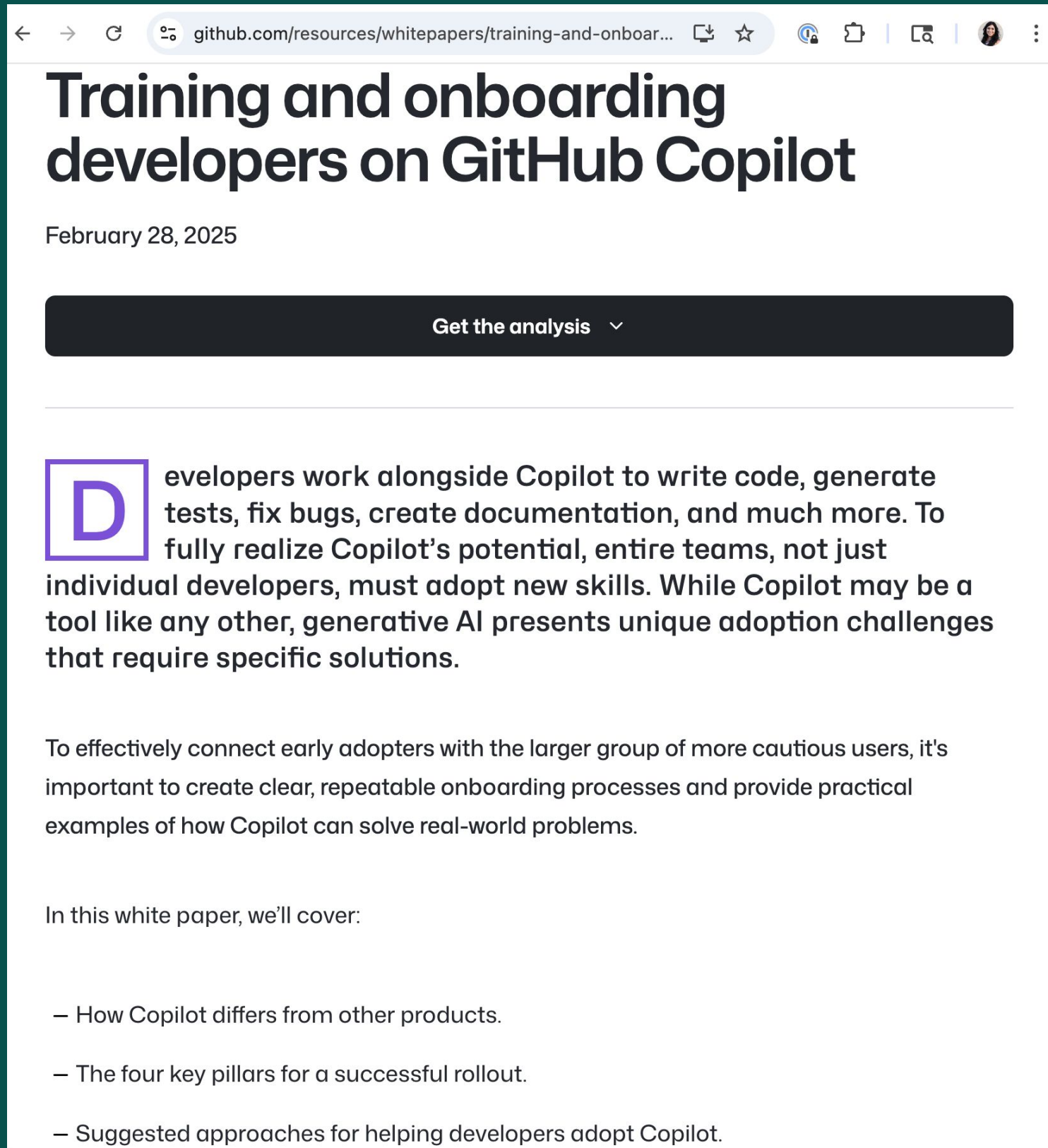
Tasks	GitHub Feature	AI Feature Ideas
Organize/Collect	Projects, Issues (milestones)	<i>Suggesting epic groupings</i>
Label ★	Issues/PRs (labels)	<i>Suggest default labels or go ahead and “apply the right labels” as a first round</i>
Prioritize ★	Projects, Issues (milestones)	<i>Suggest dependencies and prioritize backlog based on dependencies</i>
Catch up	Notifications	<i>“While you were gone” on your issue or project board</i>
Notify/Inform/Distribute	Discussions, Issues (comments), PRs (comments) + Notifications	<i>If an individual is consuming from a project board and priorities change, propagating that priority shift to the place they consume</i>
Delegate	Issues (assignees) PRs (assignees)	<i>Suggest assignees based on who has typically completed these tasks</i>
Document	Discussions, Issues (comments), PRs (comments), Repos (ex: docs/)	<i>Suggest cross-posting to an epic for a big decision within a sub issue</i>

Charting a Path



Tasks		GitHub Feature	AI Feature Ideas
Organize/Collect	1	Projects, Issues (milestones)	<i>Suggesting epic groupings</i>
Label	★	Issues/PRs (labels)	<i>Suggest default labels or go ahead and “apply the right labels” as a first round</i>
Prioritize	★	Projects, Issues (milestones)	<i>Suggest dependencies and prioritize backlog based on dependencies</i>
Catch up	2	Notifications	<i>“While you were gone” on your issue or project board</i>
Notify/Inform/Distribute	★	Discussions, Issues (comments), PRs (comments) + Notifications	<i>If an individual is consuming from a project board and priorities change, propagating that priority shift to the place they consume</i>
Delegate	3	Issues (assignees) PRs (assignees)	<i>Suggest assignees based on who has typically completed these tasks</i>
Document	4	Discussions, Issues (comments), PRs (comments), Repos (ex: docs/)	<i>Suggest cross-posting to an epic for a big decision within a sub issue</i>

Having a successful Rollout



The screenshot shows a web browser window with the URL `github.com/resources/whitepapers/training-and-onboar...`. The page title is "Training and onboarding developers on GitHub Copilot" with a date of "February 28, 2025". Below the title is a dark button labeled "Get the analysis". The main text starts with a large "D" icon and discusses the challenges of adopting Copilot. It mentions that developers work alongside Copilot to write code, generate tests, fix bugs, and create documentation. It states that to fully realize Copilot's potential, entire teams, not just individual developers, must adopt new skills. While Copilot may be a tool like any other, generative AI presents unique adoption challenges that require specific solutions. The text continues by stating that to effectively connect early adopters with the larger group of more cautious users, it's important to create clear, repeatable onboarding processes and provide practical examples of how Copilot can solve real-world problems. Finally, it says "In this white paper, we'll cover:" followed by a list of topics: "How Copilot differs from other products.", "The four key pillars for a successful rollout.", and "Suggested approaches for helping developers adopt Copilot."

github.com/resources/whitepapers/training-and-onboar...

Training and onboarding developers on GitHub Copilot

February 28, 2025

Get the analysis ▾

Developers work alongside Copilot to write code, generate tests, fix bugs, create documentation, and much more. To fully realize Copilot's potential, entire teams, not just individual developers, must adopt new skills. While Copilot may be a tool like any other, generative AI presents unique adoption challenges that require specific solutions.

To effectively connect early adopters with the larger group of more cautious users, it's important to create clear, repeatable onboarding processes and provide practical examples of how Copilot can solve real-world problems.

In this white paper, we'll cover:

- How Copilot differs from other products.
- The four key pillars for a successful rollout.
- Suggested approaches for helping developers adopt Copilot.

<https://github.com/resources/whitepapers/training-and-onboarding-developers-on-github-copilot>

- GOAL: Pick a good goal
- CHAMPIONS: Find the right champions
- REMINDERS: Set the right reminders to increase adoption
- KNOWLEDGE TRANSFER: workshops, hackathons, training, wikis



03

MEASURING

Where does the **time** go?

Time saved = banked time

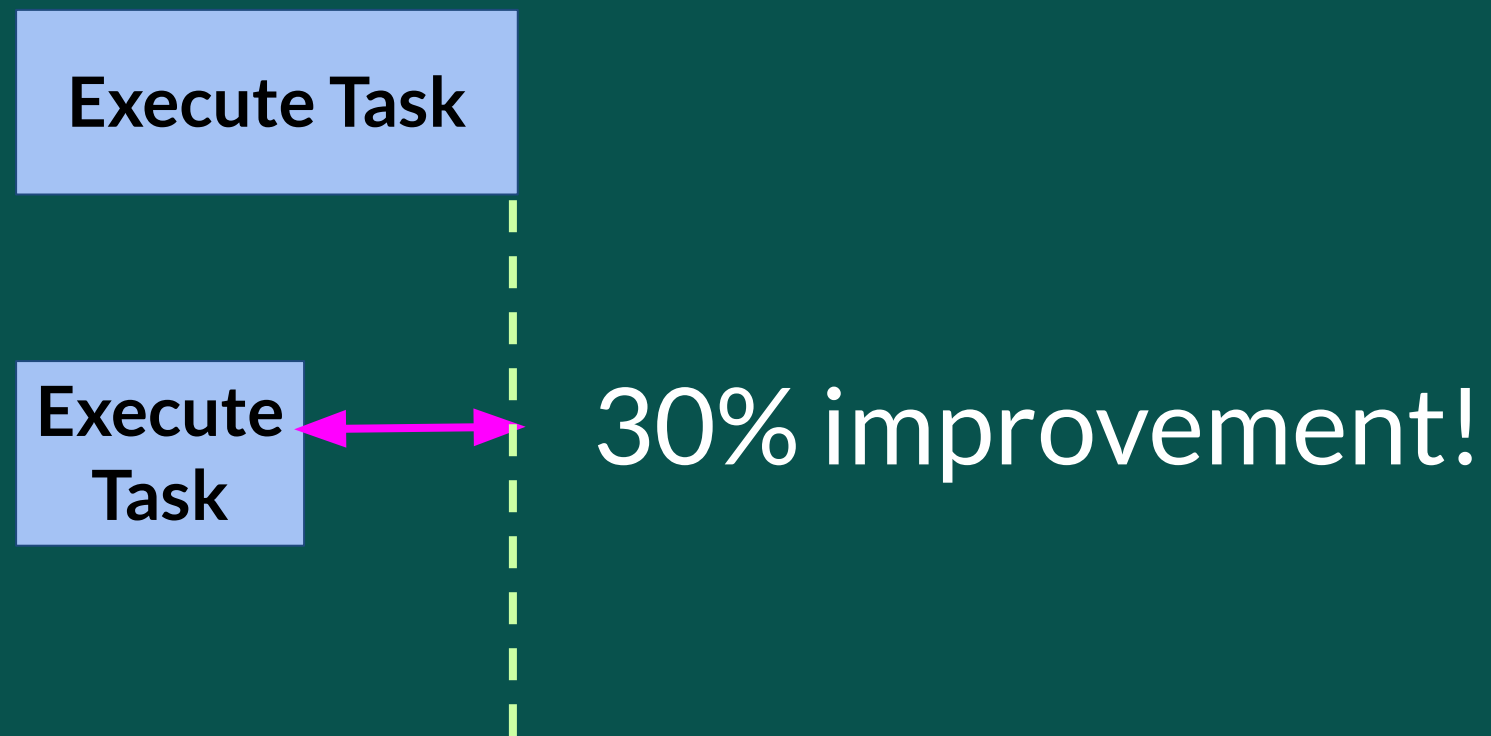
Where is the banked time spent? **Follow the time!**

1. Overworked engineers can now work less
2. Engineers switch to next task faster only to hit the fixed cost of learning what's next

Flexible vs Inflexible Time



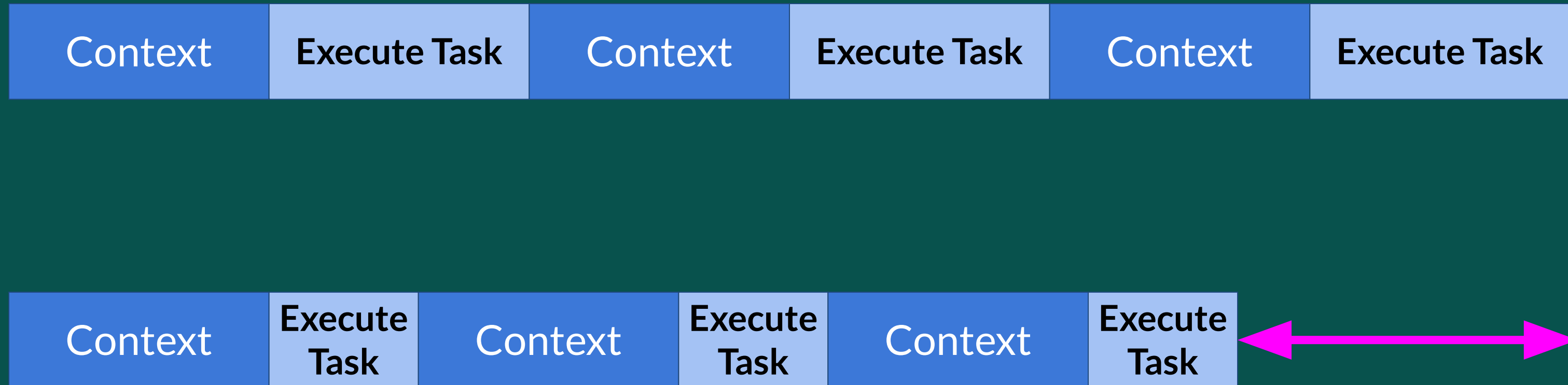
Flexible vs Inflexible Time



Flexible vs Inflexible Time



Flexible vs Inflexible Time



not a 30% improvement!

There's no magic answer

Tasks		GitHub Feature	AI Feature Ideas
Organize/Collect	1	Projects, Issues (milestones)	<i>Suggesting epic groupings</i>
Label	★	Issues/PRs (labels)	<i>Suggest default labels or go ahead and “apply the right labels” as a first round</i>
Prioritize	★	Projects, Issues (milestones)	<i>Suggest dependencies and prioritize backlog based on dependencies</i>
Catch up	2	Notifications	<i>“While you were gone” on your issue or project board</i>
Notify/Inform/Distribute	★	Discussions, Issues (comments), PRs (comments) + Notifications	<i>If an individual is consuming from a project board and priorities change, propagating that priority shift to the place they consume</i>
Delegate	3	Issues (assignees) PRs (assignees)	<i>Suggest assignees based on who has typically completed these tasks</i>
Document	4	Discussions, Issues (comments), PRs (comments), Repos (ex: docs/)	<i>Suggest cross-posting to an epic for a big decision within a sub issue</i>

Pick your **keystone** problems

I'll adopt it when....

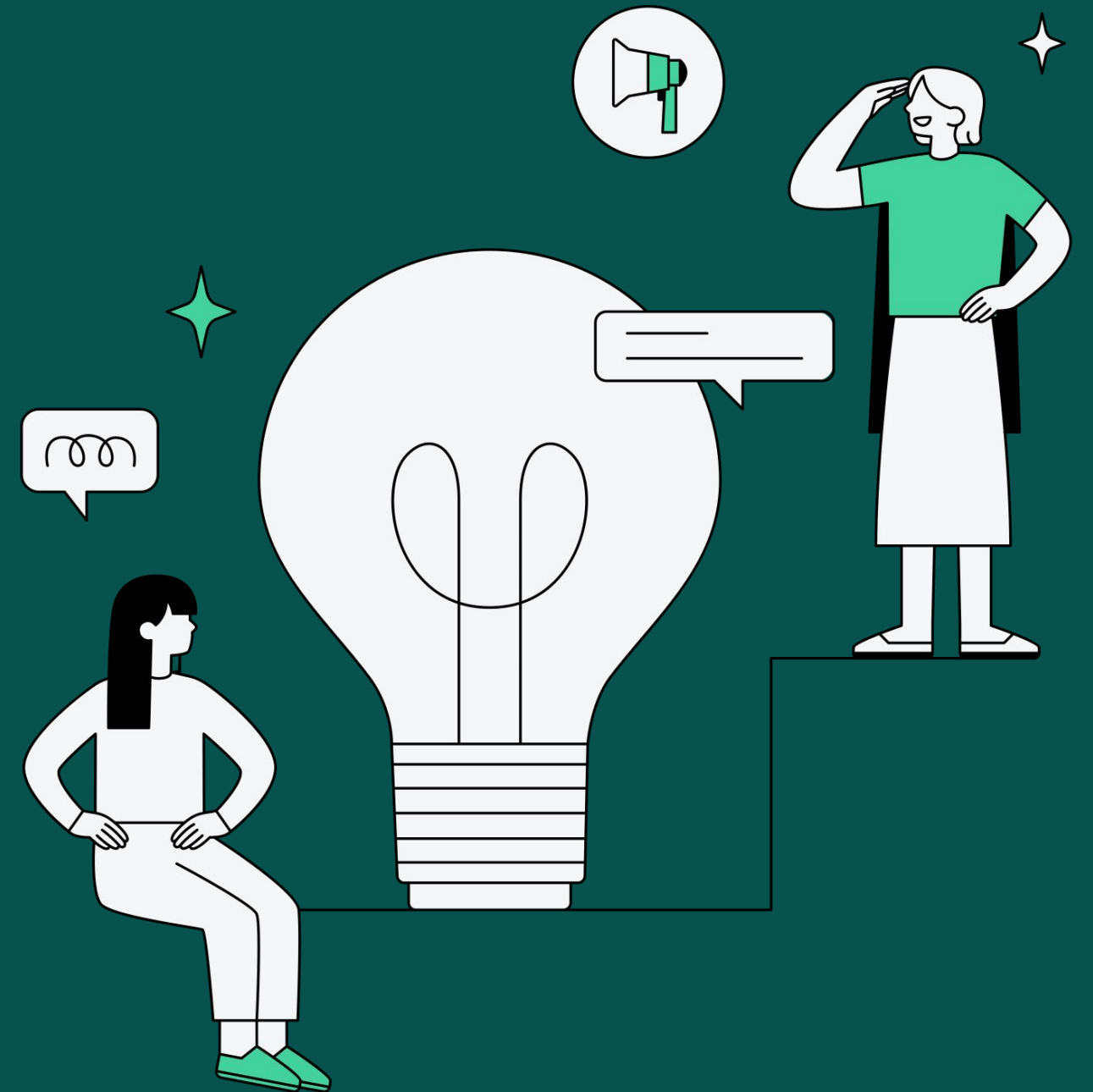
I'd be willing to try it out when...

I'll believe it when...

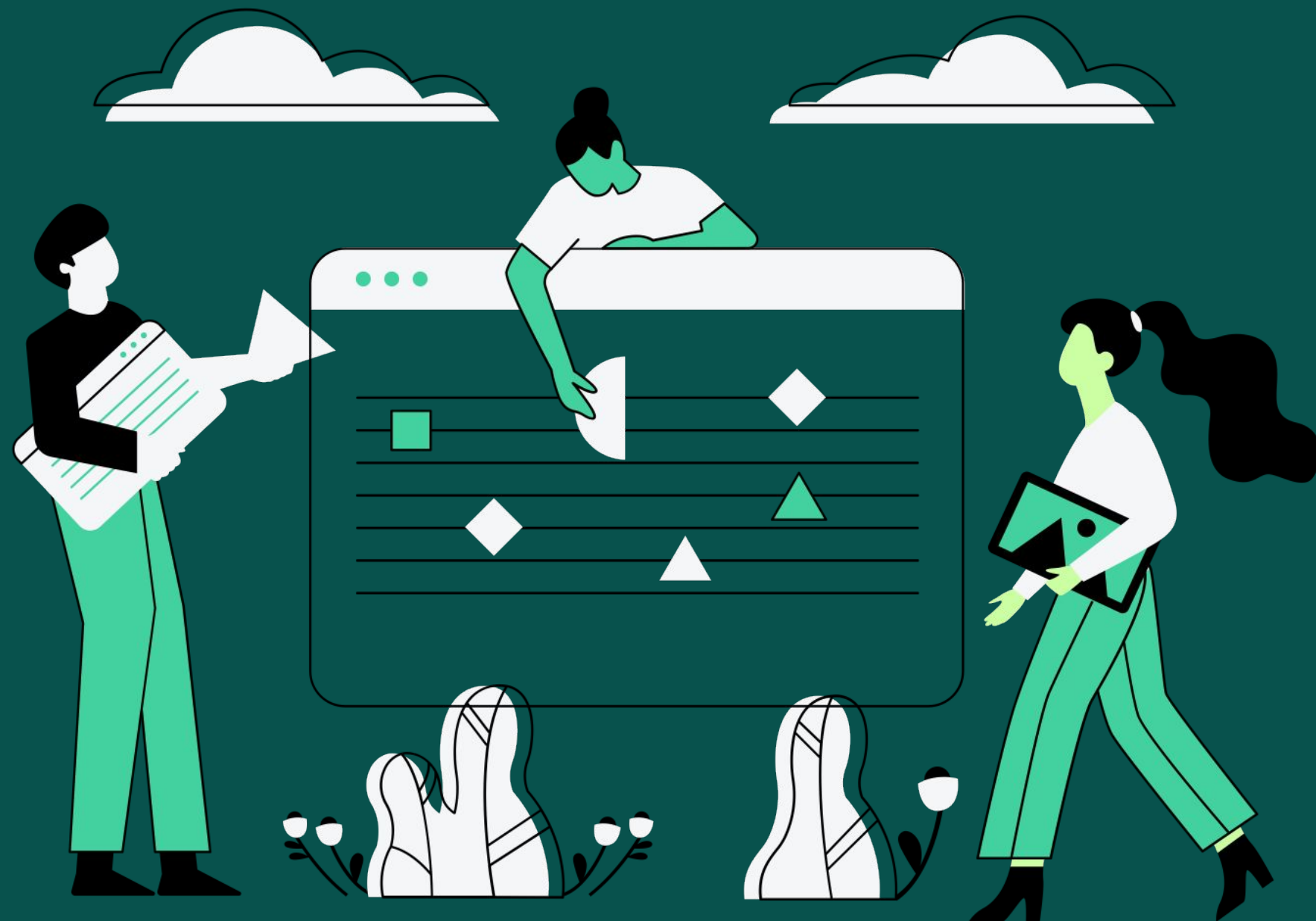
Share and Learn

- We are all trying to **fix our own developer workflows** and improve things for our team.
- We all are **getting the same pressures** to answer, fix, improve.
- We are all **looking the latest** releases of different products.
- We all **don't know what the answer is.**

But we are AT A CONFERENCE WITH
ACCESS TO EACH OTHER 🙄🙄🙄🙄



To fix, you must find



01 *Identifying*
Inner Loop vs. Outer Loop

02 *Modifying*
Automation vs. AI

03 *Measuring*
Flexible Time vs. Inflexible Time

Credits

This presentation template is free for
everyone to use thanks to the following:



for the presentation template

Pexels & Pixabay

for the photos