



The **AI** Productivity

---

# Polarity/ Paradox

---

There's more to it than you think



# Hi, I'm **Karthik.**

---

Karthik Rameshkumar, *Director - Solutions Engineering*  
[linkedin.com/in/karthik-rameshkumar](https://linkedin.com/in/karthik-rameshkumar)

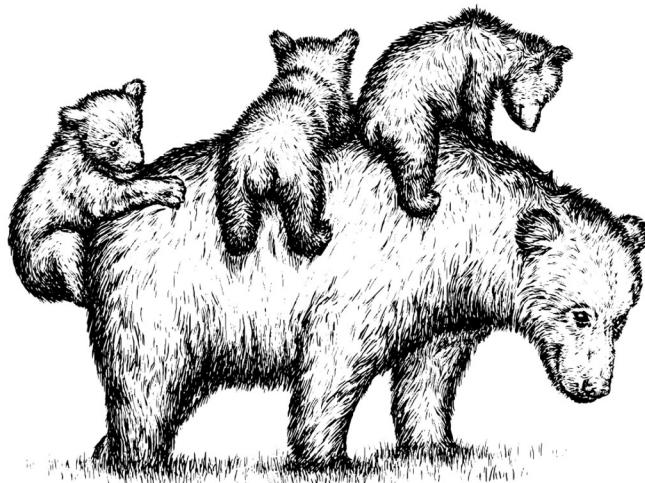
---



Karthik Rameshkumar  
Director @ GitHub | Leading APAC West Solutions Engin...



*Getting the wrong idea from that conference talk you attended*



# Solving Imaginary Scaling Issues

*At Scale*

O RLY?

@ThePracticalDev



# Why developer productivity matters

Learning how to define, measure, and predict developer productivity can help organizations:



Make higher-quality software



Produce software more efficiently



Make smarter planning decisions



Improve the well-being and satisfaction of developers



Facilitate team collaboration and success



# On behalf of all EMs out there

This is how it feels



# Myths and misconceptions about developer productivity

You can't improve what you don't understand. Here are the common myths and misconceptions around developer productivity:

01

Productivity is all about developer activity

02

Productivity is only about individual performance

03

One productivity metric can tell you everything

04

Productivity measures are only useful for managers

05

Productivity comes down to engineering systems and developer tools



# SPACE: A framework for understanding developer productivity

The SPACE framework offers a new way of thinking about developer productivity.



Satisfaction and well-being

Performance

Activity

Communication and collaboration

Efficiency and flow



S

# Satisfaction and well-being

## Satisfaction

How fulfilled developers feel with their work, team, tools, or culture

## Well-being

How healthy and happy developers are

### Metrics:

- Employee satisfaction
- Developer efficacy
- Engagement
- Burnout



# Performance

## Performance

The outcome of a system or process.  
It's hard to quantify performance  
because there are so many variables.

## Metrics:

### Quality of code

- Reliability
- Absence of bugs
- Ongoing service health

### Impact of code

- Customer satisfaction
- Customer adoption and retention
- Feature usage
- Cost reduction



A

# Activity

## Activity

the number of actions or outputs completed in the course of performing work

### Metrics:

- Design and coding
- Continuous integration and deployment
- Operational activity
- Intractable activities



C

# Communication and collaboration

**Communication and collaboration** capture how people and teams communicate and work together.

## Metrics:

- Discoverability of documentation and expertise
- How quickly work is integrated
- Quality of reviews of work contributed by team members
- Network metrics that show who is connected to whom and how



# Efficiency and flow

## Efficiency and flow

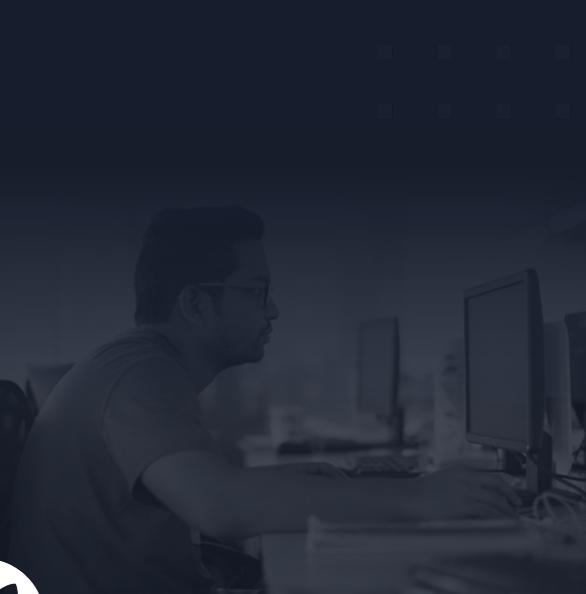
refer to how well developers and teams can make progress on their work or complete it without interruptions or delays.

### Metrics:

- Number of handoffs in a process; number of handoffs across different teams in a process
- Perceived ability to stay in flow and complete work
- Interruptions
- Time measures through a system



# How to use the framework



Measuring metrics across at least three different dimensions of the framework.

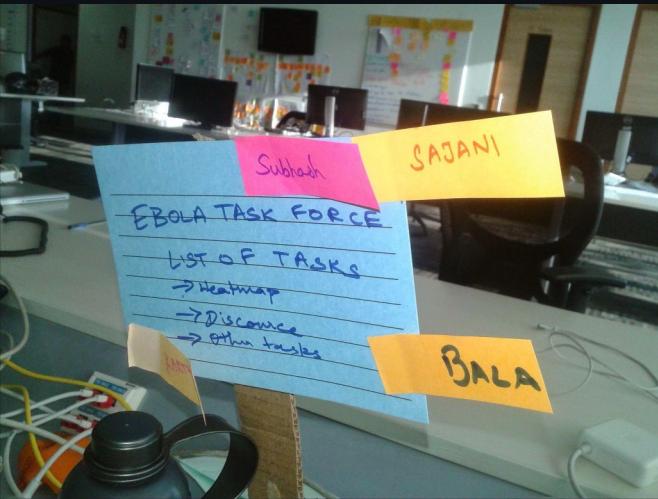
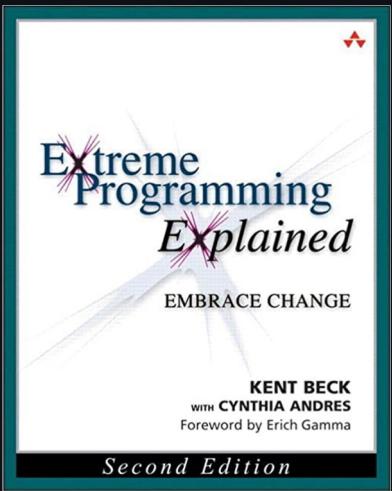
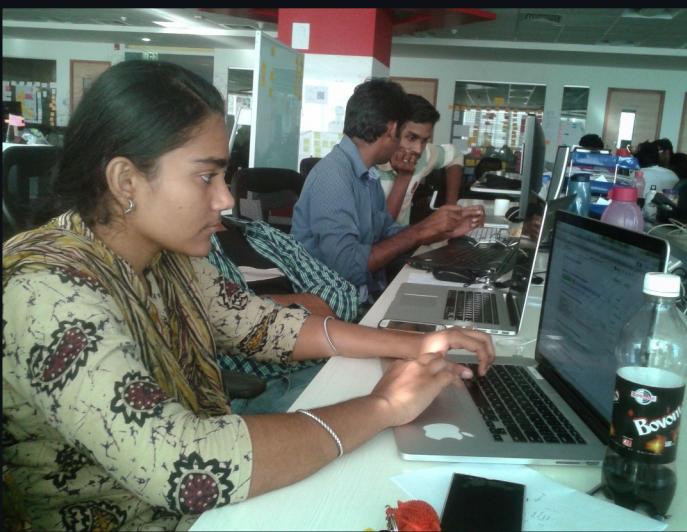
## Balance your metrics:

- Consider your organization
- Review multiple metrics

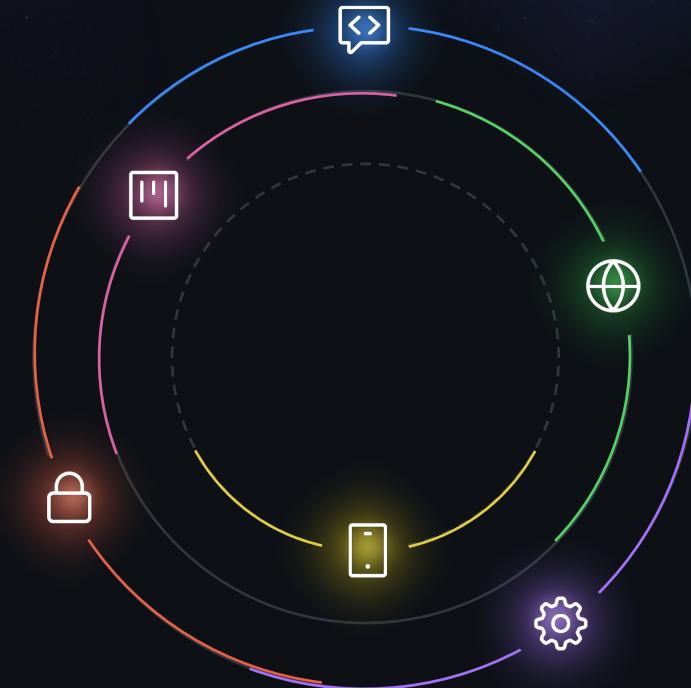
## Remember

Metrics signal what a company values. The more varied the metrics, the more balanced your picture of overall developer productivity.

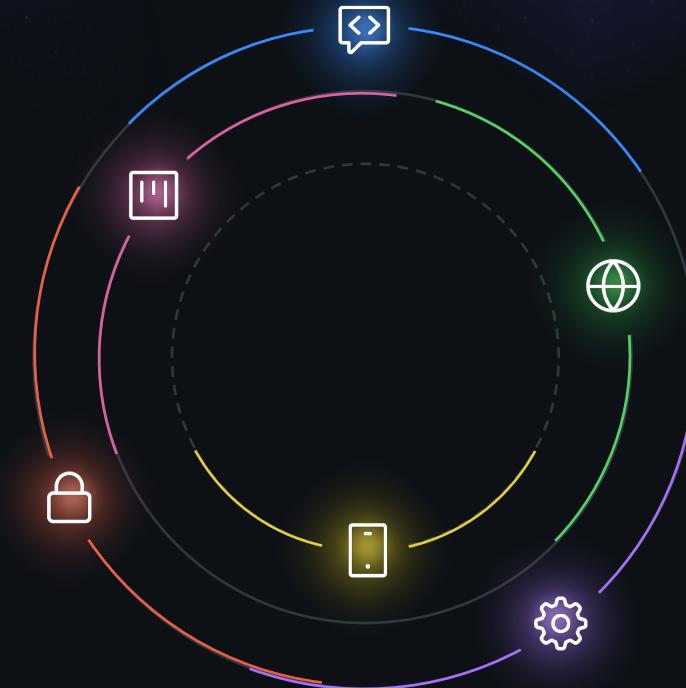




# But it is no longer 2015!



# The age of generative AI is here



# Generative AI |

## through the lens

### of software development



# Developer Journey

## Notepad

Hardcore development with no syntax support

## CLI

Command based development and publishing

## IDE with IntelliSense

Built-in language support in the IDE

## IDE

Integrated Development Environment with built-in support for build and deployment

## AI Companion

Ideas to Code in faster and efficient manner

# GitHub Copilot

## GitHub Copilot

Your AI Pair Programmer

2021

## GH Copilot Chat

Empowering Developers with Natural Language

2023

## Copilot CLI

## Copilot Enterprise

Collaborate Smarter, Code Faster

2024

## Copilot Mobile

## Copilot APIs

## Copilot Extensions

Unlocking unlimited possibilities with our ecosystem of partners

## Copilot Workspace

From Ideas to Code to Software, Naturally

\*Future

\*Future



Using **SPACE**

---

# Copilot meets SPACE

---

How GitHub Copilot supports productivity



# Developers distrust productivity tools



Anxiety-inducing



Often used against  
devs



Inaccurate/unfair  
representation of  
devs' work



Feel like spying





Talk is Cheap.  
Show me the  
Productivity!

*Your application is a special snowflake*

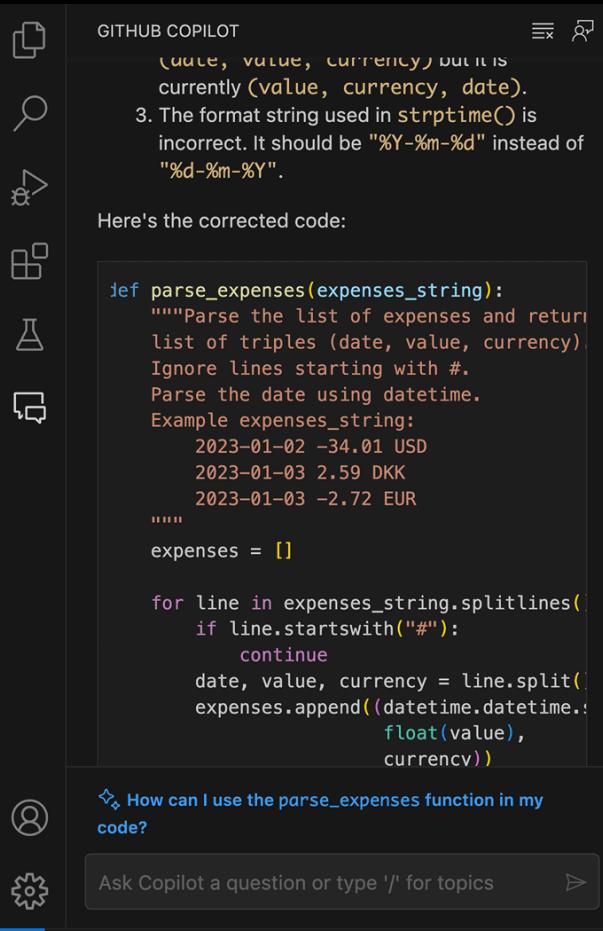


*Expert*

## Excuses for Not Writing Unit Tests



# Unit Tests



GITHUB COPILOT

3. The format string used in `strftime()` is incorrect. It should be "%Y-%m-%d" instead of "%d-%m-%Y".

Here's the corrected code:

```
def parse_expenses(expenses_string):
    """Parse the list of expenses and return
    list of triples (date, value, currency)
    Ignore lines starting with #.
    Parse the date using datetime.
    Example expenses_string:
        2023-01-02 -34.01 USD
        2023-01-03 2.59 DKK
        2023-01-03 -2.72 EUR
    ....
    expenses = []

    for line in expenses_string.splitlines():
        if line.startswith("#"):
            continue
        date, value, currency = line.split()
        expenses.append((datetime.datetime.strptime(date, "%Y-%m-%d"),
                         float(value),
                         currency))
```

❖ How can I use the parse\_expenses function in my code?

The screenshot shows a terminal window with three tabs at the top: PrimeTestCase.py M, utils.py M, and parse\_expenses.py M. The parse\_expenses.py tab is active, displaying the following code:

```
1 import datetime
2
3 def parse_expenses(expenses_string):
4     """Parse the list of expenses and return the
5     list of triples (date, value, currency).
6     Ignore lines starting with #.
7     Parse the date using datetime.
8     Example expenses_string:
9         2023-01-02 -34.01 USD
10        2023-01-02 2.59 DKK
```

Below the code, there are three tabs: OUTPUT, TERMINAL, and DEBUG CONSOLE. The TERMINAL tab is selected, showing the following command-line session:

- `~/Dev/github.c/affrae/github-copilot-examples` ↵ `main !3 > cd py_`
- `0 08:21:07`
- `parse_expenses.py`
- Traceback (most recent call last):  
File "/Users/dfigucio/Developer/github.com/affrae/github-copilot-ex\_thon/parse\_expenses.py", line 28, in <module>  
 expenses = parse\_expenses(expenses\_data)  
File "/Users/dfigucio/Developer/github.com/affrae/github-copilot-ex\_thon/parse\_expenses.py", line 18, in parse\_expenses  
 date, value, currency = line.split(",")  
ValueError: not enough values to unpack (expected 3, got 1)
- `parse_expenses.py`
- `2023-01-02 00:00:00 -34.01 USD`
- `2023-01-03 00:00:00 2.59 DKK`
- `2023-01-03 00:00:00 -2.72 EUR`
- `parse_expenses.py` ↵ `main !4`

github-copilot-examples > python > ✨ parse\_expenses.py > 📁 parse\_expense

```
1 import datetime
2
3 def parse_expenses(expenses_string):
4     """Parse the list of expenses and return the
5     list of triples (date, value, currency).
6     Ignore lines starting with #.
7     Parse the date using datetime.
8     Example expenses_string:
9         2023-01-02 -34.01 USD
10        2023-01-03 2.50 NOK
```

OUTPUT TERMINAL DEBUG CONSOLE

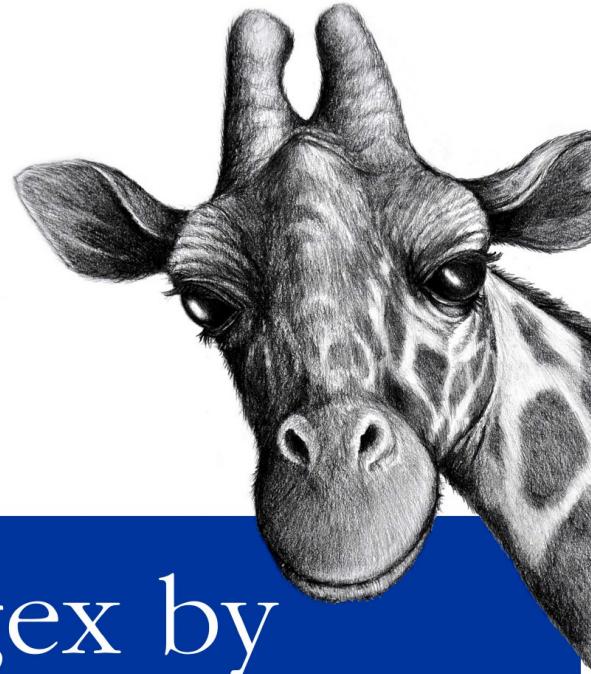
```
● 🍏 ✘ ~/Dev/github.c/affrae/github-copilot-examples 🐍 main !3 > cd python  
    0 08:21:07  
● 🍏 ✘ ~/Dev/github.c/affrae/github-copilot-examples/python 🐍 main !3 > python3  
    parse_expenses.py  
Traceback (most recent call last):  
  File "/Users/dfigucio/Developer/github.com/affrae/github-copilot-examples/py  
thon/parse_expenses.py", line 28, in <module>  
    expenses = parse_expenses(expenses_data)  
  File "/Users/dfigucio/Developer/github.com/affrae/github-copilot-examples/py  
thon/parse_expenses.py", line 18, in parse_expenses  
    date, value, currency = line.split(",")  
ValueError: not enough values to unpack (expected 3, got 1)  
● 🍏 ✘ ~/Dev/github.c/affrae/github-copilot-examples/python 🐍 main !3 > python3  
    parse_expenses.py  
2023-01-02 00:00:00 -34.01 USD  
2023-01-03 00:00:00 2.59 DKK  
2023-01-03 00:00:00 -2.72 EUR
```

○    ~/Dev/github.c/affrae/github-copilot-examples/python  main !4 >

Ln 15, Col 34 Spaces: 4 UT

Ln 15, Col 34 Spaces: 4 UTF-8 LF {} Python 3.9.6 64-bit CodeQuote - No Matches ⚙️

*Combining slashes and dots until a thing happens*



*Expert*

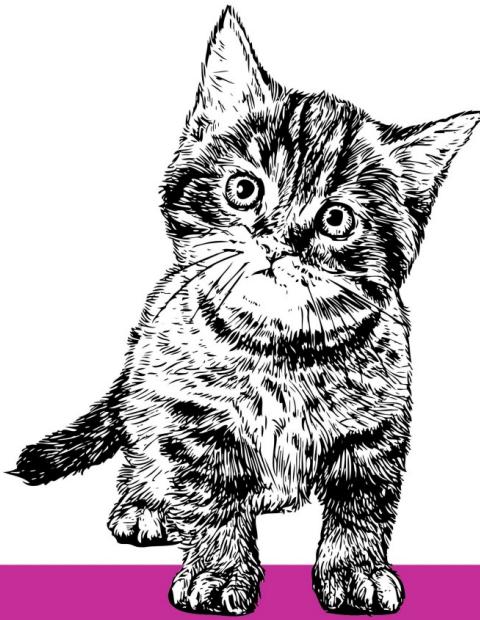
# Regex by Trial and Error

# Copilot in CLI

```
chrisreddington@MBP ~ %
```

I

*How to actually learn any new programming concept*



*Essential*

Changing Stuff and  
Seeing What Happens

# Copilot Workspaces

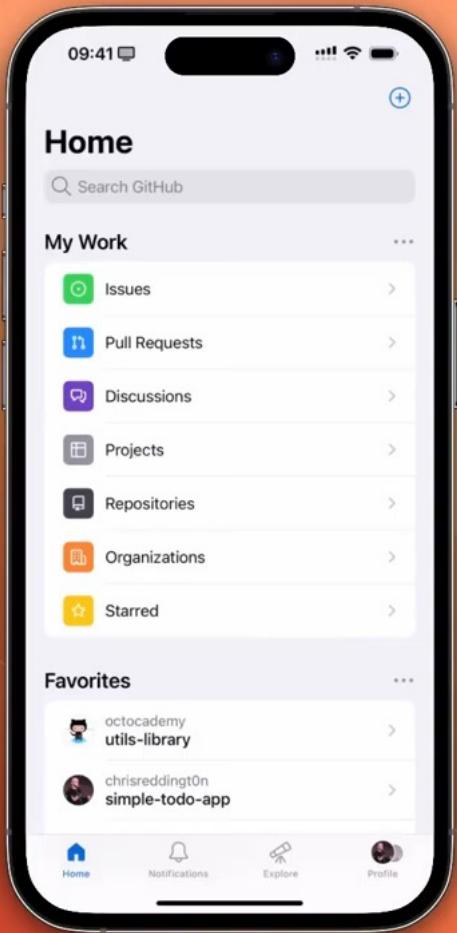
The screenshot shows a Jira workspace titled "Octocademy Utilities Library". The interface includes a header with a search bar and various navigation icons. Below the header, there are tabs for "Status board", "Roadmap", "My items", and "New view". A filter bar allows searching by keyword or field.

The workspace displays four backlog items:

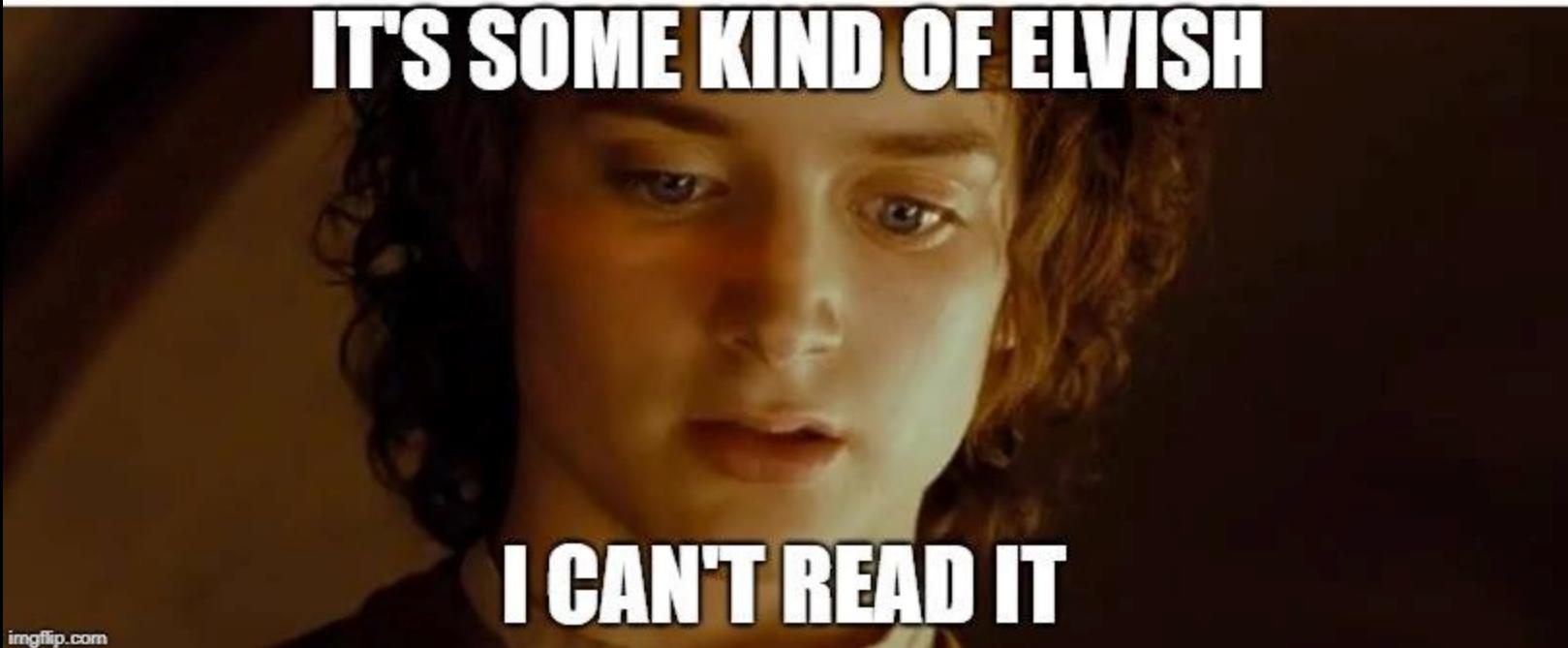
- Backlog (2/3 Estimate: 7)**: This item hasn't been started. It's associated with util-library #2 and has a sub-task "Update the README". Labels include enhancement, documentation, good first issue, M, and PO. Iteration 1.
- Ready (0 Estimate: 0)**: This is ready to be picked up. It's associated with util-library #3 and has a sub-task "Create a frontend to manually test the validation methods". Labels include enhancement, help wanted, M, and P2. Iteration 1.
- In progress (1/3 Estimate: 3)**: This is actively being worked on. It's associated with util-library #5 and has a sub-task "Add additional validations". Labels include enhancement, help wanted, S, and PO. Iteration 1.
- In review (0/5 Estimate: 0)**: This item is in review. It's associated with util-library #6 and has no visible sub-tasks or labels.

Each item has a "..." button for more options and a "+ Add item" button at the bottom of each column.

# Workspaces - Mobile



When you trying to look at  
the code you wrote a month ago

A close-up portrait of Aragorn from The Lord of the Rings. He has long, dark, wavy hair and is looking directly at the camera with a serious, slightly weary expression. The lighting is warm and focused on his face.

**IT'S SOME KIND OF ELVISH**

**I CAN'T READ IT**

How do I?

octodemo / geektrainer-demo-conference

Type ⌘ to search

Issues 2

Pull requests 1

Discussions

Actions

Projects

Wiki

Security

Insights

...

# How does a user know if they've successfully submitted a talk?

## #11

**Open** GeekTrainer opened this issue 2 weeks ago · 0 comments

GeekTrainer commented 2 weeks ago

Right now when a user submits or edits a talk they're just taken back to the list of talks. How do they know one was actually submitted? Can we update the app to display a message?

+ Add tasklist

Add a comment

Write Preview H B I ↵ ↶ ↷ ↸ ↹ ↻

Assignees  
No one—assign yourself

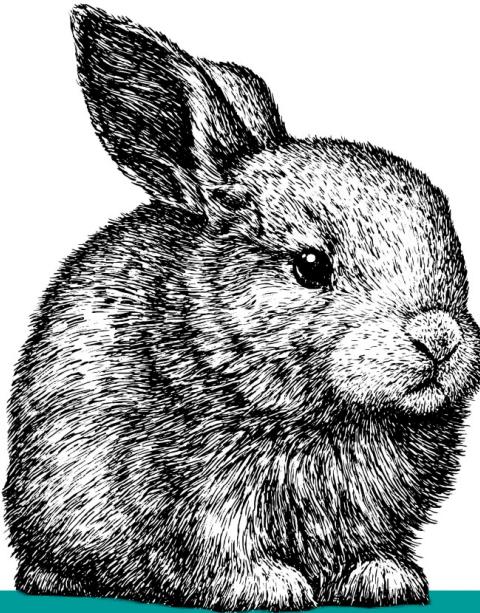
Labels  
None yet

Projects  
None yet

Milestone

Edit New issue

*Feigning knowledge of a word you've heard a few times*



*Expert*

## Pretending to Know About Stuff



*git commit -m "changes"*



*Writing*

# Useless Git Commit Messages

# Copilot PR Summaries

The screenshot shows a GitHub pull request page for the repository `octodemo / geektrainer-demo-conference`. The pull request is titled "Add Django messages to submit and edit flows #12".

**Summary:**

GeekTrainer wants to merge 1 commit into `main` from `add-success-messages`.

**Activity:**

- Conversation: 0
- Commits: 1
- Checks: 0
- Files changed: 2

**Commit Details:**

GeekTrainer commented 7 hours ago • edited

Resolves issue

Add Django messages to submit and edit flows

0fb4de0

**Reviewers:**  
No reviews  
Still in progress? [Convert to draft](#)

**Assignees:**  
No one—[assign yourself](#)

**Labels:**  
None yet

Add more commits by pushing to the `add-success-messages` branch on [octodemo/geektrainer-demo-conference](#).

*It would be a pure function if not for the side effects on your sanity*



# Turning Coffee Into Code

*The Definitive Guide*

O RLY?

@ThePracticalDev

The screenshot shows a GitHub issue page for a repository named 'octodemo / geektrainer-demo-conference'. The page title is 'Investigate latest Django version #13'. The issue was opened by 'chrisreddington' 5 hours ago and has 0 comments. The description of the issue states: 'Investigate latest version of Django (I believe the latest major version is 5.0), and needed application changes to use the latest version.' Below the description are buttons for '+ Add tasklist' and a smiley face emoji. To the right of the issue details are sections for 'Assignees' (No one—assign yourself), 'Labels' (None yet), 'Projects' (None yet), and 'Milestone' (No milestone). At the bottom left, there is a 'Write' button and a rich text editor toolbar. A search bar at the top right contains the placeholder 'Type / to search'.

# Dev-first approach to productivity

[github.co/34i4OIA](https://github.com/34i4OIA)



May 25, 2021 — Community, Company

## Octoverse Spotlight 2021: The Good Day Project—Personal analytics to make your work days better



Eirini Kalliamvakou



# “How is Copilot helping you?”

- 2.6K survey responses
- Users in the technical preview
- Vast majority professional programmers
- Assessed SPACE dimensions ++





# How does Copilot help productivity holistically?

# Satisfaction & well-being

*"Less frustrated when coding"*

*"More fulfilled with my job"*

*"Focus on more satisfying work"*



**S**

# Performance

*"Faster completion"*

*"Faster with repetitive tasks"*



**P**



# Communication & Collaboration

C

*"Learn from suggestions"*



# Efficiency & Flow

E

*"More in the flow"*

*"Less time searching"*

*"Less mental effort on repetitive tasks"*



# Benefits at all levels of the organization



*"The engineers' **satisfaction** with doing edgy things and us **giving them edgy tools** is a factor for me. Copilot makes things more exciting."*

– CTO

*"[With Copilot] I have to **think less**, and when I have to think it's the **fun stuff**. It sets off a little spark that makes coding more fun \*and\* more efficient. But we're not metric-driven, we're **emotion-driven**."*

– Eng. manager

*"It is almost like magic, it has more than once shocked me. Sometimes with a few characters it would give me the entire function. It's **built-in Stack Overflow** customized to you."*

– Senior Engineer





 We make it **flexible.**  
You  push what's  
possible.

Let's build from here.  
**[github.com](https://github.com)**

