



# Boost Your Productivity

---

with VS Code, GitHub and Azure



Developer Summit 2022



@m3rtyeter



/in/mertyeter



/mertyeter



/azureishlive  
/yenimshowto  
/TraefikLabsTurkey



/azureishlive  
/mshowto  
/traefikistanbul

# Mert Yeter

Software Architect  
**360Dotnet**



Certified  
Traefik  
Ambassador



Cloud  
Community  
Champion (2020)





# What Is Developer Velocity?

---

Developer velocity reflects how much work a developer can complete in a specific timeframe.



## Developer Velocity Assessment

**Power up your developer teams and take your business to the next level.**



**Gain deeper insights into your team's performance**

Discover your organization's score on the Developer Velocity Index (DVI) by taking your first Developer Velocity Assessment.

By answering a few simple questions, you can benchmark your maturity score relative to peers in your industry. Then, get actionable guidance to boost Developer Velocity and drive better business results.



## Get started

Are you a registered user?  
[Sign in here to check your results](#)

[Privacy Policy >](#)

See improved business outcomes like other companies with high Developer Velocity<sup>1</sup>, such as:



# Results

## Developer Velocity Index (DVI) Score

Industry: Software ▾

Industry Average: 2.9

**3.40** ^

### Category scores

### Industry

Technology	<b>4.00</b> ^	3.1
Working Practices	<b>3.10</b> ^	2.5
Organizational enablement	<b>3.20</b> ^	3.0

[Explore Now](#)

Learn how to get resources to increase  
Developer Velocity for your organization.

[Share your results](#)

Consider registering to save your progress and access your assessment anytime with your Microsoft account.

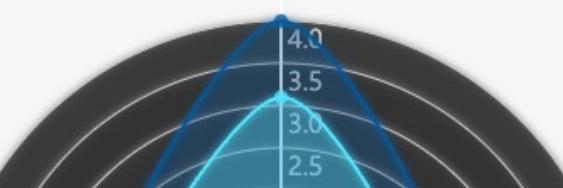


[◀ Return to the assessment](#)

[Categories](#) [Sub-categories](#)

■ Developer Velocity Score ■ Industry Average (Software)

### Technology





# GitHub Codespaces

Reducing time to set up  
development environments from  
hours per week  
to seconds.

The screenshot shows a GitHub Codespace interface. On the left, a file explorer displays files like index.js, App.ts, and package.json. The main area is a code editor with tabs for index.js (selected) and App.ts. index.js contains server-side logic using Express and PostgreSQL. The App.ts tab is also visible. Below the code editor are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL tab is active, showing the command "califa visitor-app(feature-branch-1) \$ npm run dev" and the output "Starting server on port 3000...". The status bar at the bottom shows the branch "feature-branch-1" and the name "califa".

```
JS index.js x TS App.ts
1 < const express = require('express')
2 const path = require('path');
3 const app = express()
4 const bodyParser = require('body-parser')
5 const Pool = require('pg').Pool
6
7 const http = require('http').createServer(app);
8
9 app.use(express.static(path.join(__dirname, 'build')));
10 app.use(bodyParser.urlencoded({ extended: false }))
11 app.use(bodyParser.json())
12
13 app.get('/', function (req, res) {
14   res.sendFile(path.join(__dirname, 'build', 'index.html'));
15 });
16
17 app.get('/visits/day', (request, response) => {
18   pool.query(`SELECT * FROM visits WHERE created_at > current_date - interval '24 hours' ORDER BY`)
19     if (error) {
20       throw error
21     }
22     response.status(200).json(results.rows)
23   })
24 }
25
26 app.get('/visits/week', (request, response) => {
27   pool.query(`SELECT * FROM visits WHERE created_at > current_date - interval '7 days' ORDER BY`)
```

## Code editor

**VS Code Desktop**  
Local installation

**VS Code Browser**  
Chrome, Safari, Edge

Your editor is how you view and edit your codespace

Changes to your codespace are reflected back in your editor

## Virtual Machine

Linux hardware

### Container

Docker development environment

**A clone of your repository**  
Source code

**Languages**  
Python, Ruby, etc

**Tooling**  
Extensions, linting, etc



# GitHub Copilot

Your AI pair programmer

A screenshot of a code editor interface. At the top, there are tabs for "sentiment.ts", "write\_sql.go", "parse\_expenses.py", and "addresses.rb". The main area shows a snippet of TypeScript code for determining text sentiment. The code uses the "fetch-h2" library and makes an API call to "text-processing.com/api/sentiment". The "Content-Type" header is set to "application/x-www-form-urlencoded". The code then parses the JSON response to check if the label is "pos".

```
1#!/usr/bin/env ts-node
2
3import { fetch } from "fetch-h2";
4
5// Determine whether the sentiment of text is positive
6// Use a web service
7async function isPositive(text: string): Promise<boolean> {
8  const response = await fetch(`http://text-processing.com/api/sentiment/`, {
9    method: "POST",
10   body: `text=${text}`,
11   headers: {
12     "Content-Type": "application/x-www-form-urlencoded",
13   },
14 });
15  const json = await response.json();
16  return json.label === "pos";
17}
```

**Copilot**

**Uses the context you've provided  
and synthesizes code to match**

- Convert comments to code
- Autofill for repetitive code
- Autosuggest tests
- Show alternatives

OpenAI Codex Model



Public code and text  
on the internet

GitHub



**GitHub Copilot  
Service**

Provide Editor context

Provide Suggestions

Improve Suggestions

Private Code

JS fetch\_pic.js

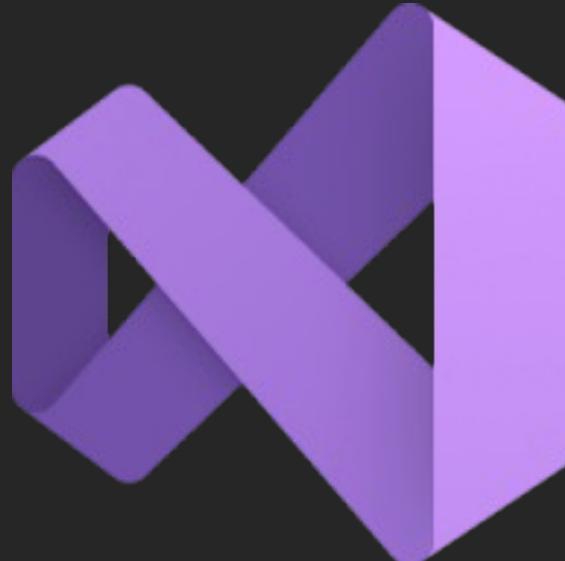
```
1 const fetchNASAPictureC
 2   return fetch('https://api.nasa.gov/mars/...
 3   method: 'GET',
 4   headers: {
 5     'Content-Type': 'application/json'
 6   },
 7 }
 8 .then(response => r...
 9 .then(json => {
10   return json;
11 });
12 }
```

Copilot



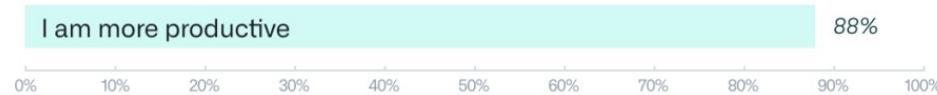
Extension for  
popular code  
editors

---

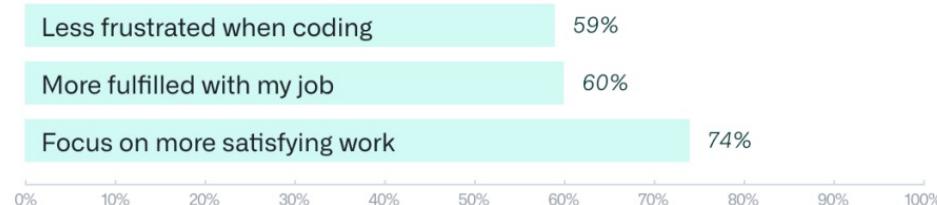


## When using GitHub Copilot...

### Perceived Productivity



### Satisfaction and Well-being\*



### Efficiency and Flow\*



We recruited

**95**

developers, and split them randomly into two groups.

We gave them the task of writing a web server in JavaScript

**45 Used**  
GitHub Copilot

**78%**  
finished

**1 hour, 11 minutes**  
average to complete the task

71 minutes | that's 55% less time!

**50 Did not use**  
GitHub Copilot

**70%**  
finished

**2 hours, 41 minutes**  
average to complete the task

161 minutes

Results are statistically significant ( $P=.0017$ ) and the 95% confidence interval is [21%, 89%]

# Microsoft Dev Box

PREVIEW



## back-end-dev-box

Project 2X

Windows 11

16 vCPU

64GB RAM

1TB Storage

## Add a dev box



### Name

front-end-dev-box

### Project

Project 2x



### Dev box pool

Standard dev box (West US)

4 vCPU 16GB RAM 256GB Storage



[See all pools](#)

# GitHub Codespaces



Linux



Repos on  
GitHub

Cloud native apps  
Including: web apps,  
APIs, backends



GitHub.com

Operating  
system



Windows



Any version  
control system

SCM  
Support

Target  
workloads

Any workload  
Including: Desktop, IoT,  
mobile, games, & more  
(Windows or cross-plat)

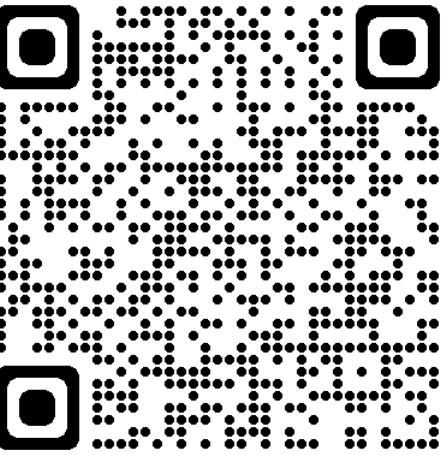
IT  
Mgmt.

Endpoint Manager Microsoft Azure

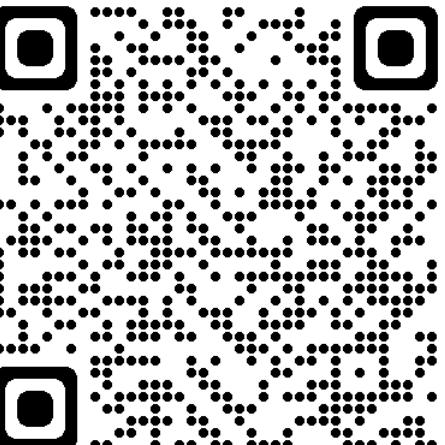


# Microsoft Dev Box

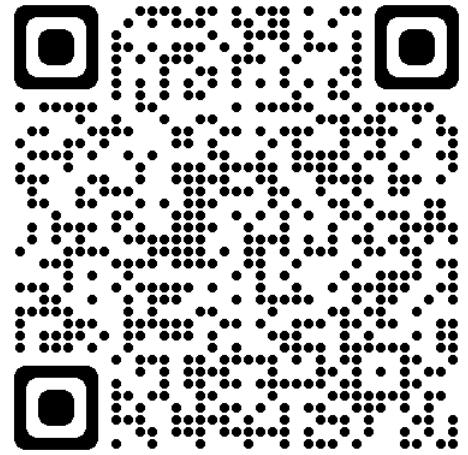
# Resources



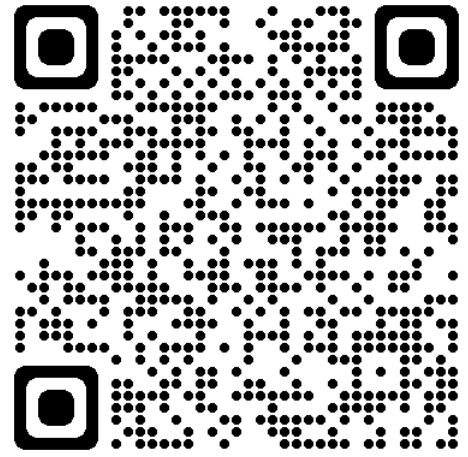
[Developer Velocity Assessment](#)



[Github Copilot](#)



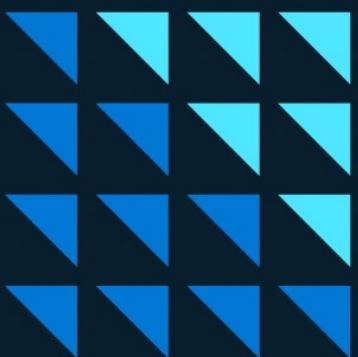
[GitHub Codespaces](#)



[Microsoft Dev Box](#)

# Join Azure Developer Community!

[Subscribe here](#)



You will receive information, tips, and offers about the Developer Community, the Developer Community Newsletter, Microsoft Azure and other Microsoft products and service.



<https://aka.ms/AAi88tb>