

# Table of Contents

## Contents

- Table of Contents ..... 1
- Lab: Measuring GitHub Copilot Impact for Managers ..... 2
  - Exercise 1: Enable and Access Copilot Usage Reports..... 2
    - Task 1: Activate Reporting Features ..... 2
    - Task 2: Download Usage Data ..... 2
  - Exercise 2: Analyze Developer Adoption Metrics ..... 2
    - Task 1: Import Data into Analytics Tool..... 2
    - Task 2: Visualize Adoption Trends ..... 2
  - Exercise 3: Calculate Productivity Improvements..... 3
    - Task 1: Establish Baseline Metrics ..... 3
    - Task 2: Compute Improvement Ratios ..... 3
  - Exercise 4: Report and Drive Adoption..... 3
    - Task 1: Build an Executive Dashboard ..... 3
    - Task 2: Present Findings & Next Steps ..... 3
  - References: ..... 4

# Lab: Measuring GitHub Copilot Impact for Managers

**Lab Duration:** 1 Hour

## Overview

In this lab, managers will learn how to monitor and quantify GitHub Copilot adoption and its impact on developer productivity. You'll configure usage reporting, interpret key metrics, and calculate productivity improvements using real-world data exports and simple analytics.

## Exercise 1: Enable and Access Copilot Usage Reports

**Duration:** 15 Minutes

### Task 1: Activate Reporting Features

- In GitHub Enterprise settings, navigate to the Copilot section and enable “Usage reports” for your organization.
- Ensure appropriate permissions are granted for report access.

### Task 2: Download Usage Data

- Export the “Copilot Usage per Developer” CSV from the GitHub Admin Console.
- Review fields such as suggestions accepted, completions generated, and active days.

## Exercise 2: Analyze Developer Adoption Metrics

**Duration:** 15 Minutes

### Task 1: Import Data into Analytics Tool

- Load the downloaded CSV into Excel, Power BI, or Google Sheets.
- Cleanse and normalize columns (e.g., developer email, usage counts).

### Task 2: Visualize Adoption Trends

- Create a bar or line chart showing weekly “suggestions accepted” per developer.
- Identify high-usage and low-usage cohorts for targeted coaching.

## Exercise 3: Calculate Productivity Improvements

**Duration:** 15 Minutes

### Task 1: Establish Baseline Metrics

- Define baseline coding metrics (e.g., PR throughput, average PR size) for a pre-Copilot period.
- Extract matching metrics via GitHub Insights or your CI/CD dashboards.

### Task 2: Compute Improvement Ratios

- Compare post-Copilot metrics to baseline to calculate percentage improvements in PR throughput and code delivery velocity.
- Use simple formulas (e.g.,  $(\text{Post} - \text{Pre}) / \text{Pre} * 100\%$ ) in your analytics tool.

## Exercise 4: Report and Drive Adoption

**Duration:** 15 Minutes

### Task 1: Build an Executive Dashboard

- Combine Copilot usage charts and productivity improvement metrics into a single Power BI or Excel dashboard.
- Include callouts for ROI estimates based on time saved per suggestion.

### Task 2: Present Findings & Next Steps

- Draft a summary report highlighting top performers, overall productivity gains, and recommended actions (e.g., training for low adopters).
- Share the dashboard and report with leadership to inform Copilot roll-out strategy.

## References:

- **Reviewing User Activity Data for Copilot**  
[Reviewing user activity data for Copilot in your organization - GitHub Docs](#)
- **Get Copilot Usage Metrics via REST API**  
<https://docs.github.com/en/rest/copilot/copilot-usage#get-copilot-metrics-for-an-organization>
- **Quantifying Copilot's Impact on Developer Productivity**  
<https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/>