



Sample

The 41:1 ROI of Moving CI/CD to Semaphore



The 41:1 ROI of Moving CI/CD to Semaphore

March 2020

Contents

The Idea in Brief	4
Executive Summary	5
The Impact of a Slow Continuous Integration Process	6
How Fast Should We Go?	7
You Don't Have to Change How You Work	8
The ROI of Semaphore	9
Conclusion	11

© 2022 Rendered Text. All rights reserved.

Published on the Semaphore website (<https://semaphoreci.com>).

This document is provided for informational purposes only. It represents Rendered Text's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of Rendered Text's products, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from Rendered Text, its affiliates, suppliers or licensors. The responsibilities and liabilities of Rendered Text to its customers are controlled by Rendered Text agreements, and this document is not part of, nor does it modify, any agreement between Rendered Text and its customers.

The Idea in Brief

The challenge: Software developers lose focus and time by waiting for code changes to propagate through slow CI/CD pipelines. That goes unnoticed because everyone assumes it's inevitable.

The solution: Engineering teams can upgrade to a CI/CD solution that's fast and scalable while also meeting all the functional requirements of their current and upcoming projects.

The outcome: Teams that switch to Semaphore report much faster code testing cycles, leading to increased productivity and more frequent deployments. The ROI of the total cost of Semaphore relative to the value of saved development time averages to 1:41.

Executive Summary

Every engineering team wants to ship working code faster. A foundational step is to automate the build, test, and deploy processes across all applications and services via continuous integration and delivery (CI/CD) pipelines¹.

When every change in code flows through it, a CI/CD pipeline can become a bottleneck. Leading continuous delivery experts suggest a 10-minute test: if it takes more than 10 minutes for a developer to get feedback on their code, the process is not providing proper continuous integration. Additionally, waiting for test results for more than a few minutes causes developers to lose focus.

To reclaim a game-changing amount of productive time, an engineering team should maximize the performance of their CI/CD pipelines. There are two ways to make your CI/CD cycles faster: optimize your code, and use a CI/CD solution with features that can speed up the process. The latter is easier to accomplish.

Semaphore is a cloud-based CI/CD solution with features that increase development velocity, such as automatic scaling to massively parallel workloads, fully customizable pipelines defined as code, and first-class support for containers. Although many hosted CI/CD products claim to be fast, Semaphore is the only one that runs on bare metal hardware, providing the best runtime performance on the market.

Teams that adopt Semaphore report significantly faster time from code commit to deploy, increased number of deployments to end users, and reductions in errors — all without making any changes in their code. Based on results reported by customers, the average ROI of CI/CD transformations with Semaphore is 41x.

¹Semaphore: “CI/CD Pipeline, A Gentle Introduction” (<https://semaphoreci.com/blog/cicd-pipeline>)

The Impact of a Slow Continuous Integration Process

Developers are known for being very protective of their time. It takes an average of 25 minutes to regain lost focus². With fixed team routines such as stand-up meetings and having to deal with outstanding issues, a developer is usually about two major interruptions away from not being able to complete the main challenge of the day.

The paradox is that it may seem as if developers cause their own interruptions.

The problem is when developers can cause their own interruptions simply by seeking to verify their code.

Pushing new code to run automated build and test steps as part of the continuous integration (CI) process³ shouldn't take longer than the time it takes to get a cup of coffee. If it does, integrating any code change creates a gap that is the equivalent of asking a developer to join a meeting in the middle of solving a problem. If they can't quickly verify their work in progress, developers will be less effective due to inevitable context switching.

It's not uncommon to have a CI build that takes an hour to run. In that scenario, the entire engineering team — no matter its size — has a hard limit of up to seven deploys for the whole workday. This causes developers to opt for less frequent and more risky deployments, instead of the rapid change that businesses today need.

²The New York Times: “Multitasking Can Make You Lose ... Um ... Focus” (<https://www.nytimes.com/2008/10/25/business/yourmoney/25shortcuts.html>)

³Semaphore: “Continuous Integration Explained” (<https://semaphoreci.com/continuous-integration>)

Download the full whitepaper for free

We hope you have enjoyed this small sample of the whitepaper.

Download the the full whitepaper for free here:

<https://semaphoreci.com/resources/roi>