semaphore

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

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

March 2020

Contents

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) pipelines[1].

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.

[1] Semaphore: "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 focus[2]. 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.

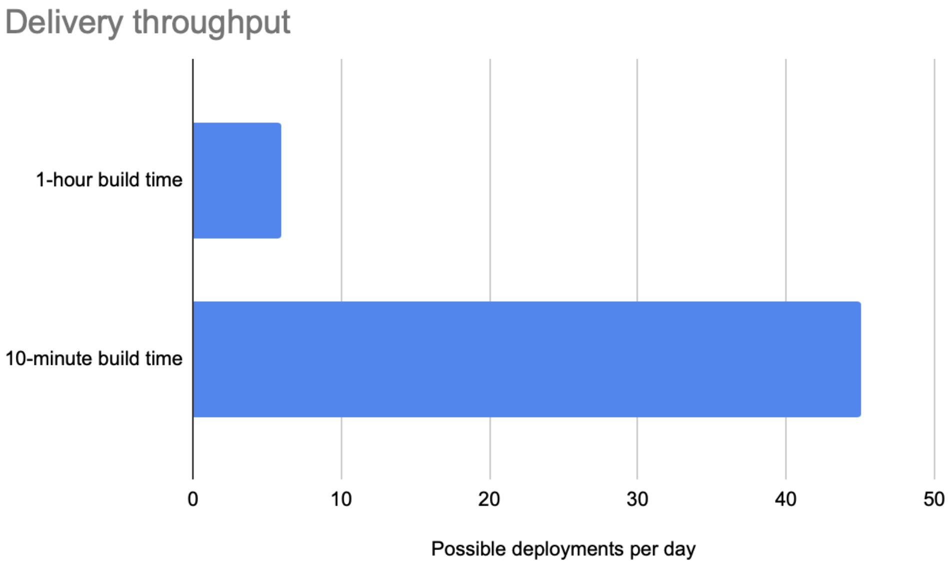
The problem is when developers can cause their own interruptions simply by seeking to verify their code.

Pushing new code to run automated build and test steps as part of the continuous integration (CI) process[3] 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.

[2] The New York Times: "Multitasking Can Make You Lose … Um … Focus" (*https://www.nytimes.com/2008/10/25/business/yourmoney/25shortcuts.html*)

[3] Semaphore: "Continuous Integration Explained" (*https://semaphoreci.com/continuous-integration*)



There are more factors that inhibit the speed of software development. It can be useful to consider these questions:

- **Can your CI/CD pipelines scale to meet development demands in real time?** Traditionally CI/CD pipelines have limited capacity, meaning that only a certain number of pipelines can run at a given time. As a result, resources sit idle most of the time, while during busy periods of the day, developers wait in a queue for pipelines to become available.
- **How quickly can you set up a new pipeline?** Difficulty with scaling CI/CD infrastructure or reusing existing configuration creates friction, and that stifles development. Today's cloud infrastructure is best utilized by writing software as a composition of microservices, which calls for frequent initiation of new CI/CD pipelines.

How Fast Should We Go?

Jez Humble, a leader in software delivery and co-author of the seminal *Continuous Delivery* book, came up with the following informal CI certification process[4]:

> He usually begins the certification process by asking his audience to raise their hands if they do Continuous Integration. Usually, most of the audience raises their hands.
>
> He then asks them to keep their hands up if everyone on their team commits and pushes to a shared mainline (usually shared master in git) at least daily.
>
> Over half the hands go down. He then asks them to keep their hands up if each such commit causes an automated build and test. Half the remaining hands are lowered.
>
> Finally, he asks if when the build fails, it's usually back to green within ten minutes.
>
> With that last question, only a few hands remain. Those are the people who pass his certification test.

So it's simple: **Does it take you less than 10 minutes from pushing new code to getting results?** If so, congratulations. Your team is equipped for high performance. If not, your workflow only has elements of a CI process, for lack of a better term. Moreover, this slowness develops detrimental habits and limits the productivity of all developers in a team. That ultimately impedes the performance of the company as a whole.

You Don't Have to Change How You Work

Fortunately, CI/CD can move faster by simply changing tools. Semaphore[5] brings capabilities that empower development teams to move faster and more efficiently than before:

- **Fully customizable CI/CD pipelines**: From simple builds to complex workflows described in the *Continuous Delivery* book, such as fan-out/fan-in and multi-stage releases, Semaphore makes it all possible. Your organization will be able to accurately automate and visualize every software delivery process and eliminate manual errors.

[4] Martin Fowler: ContinuousIntegrationCertification (*https://martinfowler.com/bliki/ContinuousIntegrationCertification.html*)

[5] Semaphore, hosted CI/CD service (*https://semaphoreci.com*)

- **Unbeatable speed**: Semaphore is by far the fastest CI/CD service on the market, providing bare metal performance with the convenience of the cloud.
- **Developer focus**: With a visual Workflow Builder that lets anyone configure pipelines without writing YAML, configuration that is stored along with rest of the source code, and a first-class command-line interface (CLI), Semaphore makes it possible for every developer to contribute to CI/CD, and it easily fits into the existing workflows.
- **Serverless auto-scaling model**: Semaphore is a fully hosted CI/CD solution that scales on-demand to massively parallel workloads, which supports developer productivity. Pricing starts at $0 and is based on per-minute usage, without additional per-user costs.

The ROI of Semaphore

A survey of typical Semaphore customers finds that for every dollar spent on Semaphore, engineering teams gain $41 in saved time that they can use productively.

How we measured it: Semaphore customers shared the difference in CI build performance before and after using Semaphore. That was our starting point. Next, we took a look at how often these teams push new code per day, taking into account the team size. Finally, we considered the monthly cost of Semaphore for these teams, benchmarking it against the monetary value of developer time. After analyzing all of these factors, we calculated the ROI.

In summary, the ROI is based on comparing *the difference in CI build performance* versus *the total price of Semaphore's CI/CD service.*

Table 1: Calculating the ROI of Semaphore.

Outcomes per customer	C1	C2	C3	C4	C5	**Average**
Team size	10	48	12	12	16	**20**
Saved time/build, hours	0.33	0.13	0.33	0.87	0.88	**0.51**

Outcomes per customer	C1	C2	C3	C4	C5	**Average**
New code pushed, times/day	33	120	16	22	44	**47**
Value/month	$16,170	$23,520	$7,840	$28,028	$57,134	**$26,538**
Semaphore cost/month	$499	$2,036	$118	$738	$1,008	**$880**
ROI (Value:Semaphore cost)	32	12	66	38	57	**41**

Using the constants $salary = \$70$, representing median hourly total compensation for a Software Engineer in New York City in 2020[6], and $month = 21$ as the number of days in a month, we calculate monthly value created by Semaphore as:

$$Value = (Saved\ time/build) \times (New\ code\ pushed/day) \times salary \times month$$

The drastic improvement in CI feedback loop encourages developers to make changes more frequently without losing focus. Instead of avoiding pushing new code, developers feel comfortable integrating smaller changes.

The CTO of Superhuman was looking for a CI/CD solution to support their web, desktop, and mobile applications. After trying Semaphore, he said[7]:

> It's now feasible to push to the CI server just to see whether the build passes, and get results without losing context.

As we know, this in turn leads to more frequent code deployment and product iterations. Here's another example:

A team at Indeed, a popular careers website, has also experienced increased ROI after moving to Semaphore. Indeed's engineering team cut their average

[6] Built In NYC: Software Engineer Salary in New York, March 2020 (*https://www.builtinnyc.com/salaries/dev-engineer/software-engineer/new-york*)

[7] Case study: "Superhuman finds the perfect solution to ship software faster" (*https://semaphoreci.com/customers/superhuman*)

CI build time by more than half and hit their goal of doing builds in under 10 minutes. Before the switch, the team averaged two to three deployments a day. Using Semaphore, they now deploy anywhere from 4 to 10 times a day[8]. They achieved all of these results despite a complex application with a large test suite hosted on Kubernetes. From Indeed's Senior Software Engineer:

> Semaphore allows us to build, tag, push, and run Docker images easily. This makes building a powerful pipeline where we can deploy to our Kubernetes cluster fairly easily and quickly.

The drastic improvement in the CI feedback loop that Semaphore customers have gained encourages developers to make changes more frequently by removing the fear of losing focus. Instead of avoiding pushing push new code, developers feel comfortable integrating smaller changes. That, in turn, leads to more frequent code deployment and product iterations.

Conclusion

If your build times are longer than 10 minutes, you're leaving time and money on the table. A well-oiled CI/CD pipeline helps, not hinders, your team's productivity, giving you more time to focus on creating better products. Your software will improve as a result, and your team's job satisfaction will inevitably increase.

Semaphore will give you greater control over your pipeline and allow you to release more frequently, but you don't have to take our word for it. Try Semaphore free with unlimited capacity for 14 days with your own projects. Learn more at https://semaphoreci.com.

[8] Case study: "Indeed Deploys 3x More Often with Semaphore 2.0" (*https://semaphoreci.com/customers/indeed*)