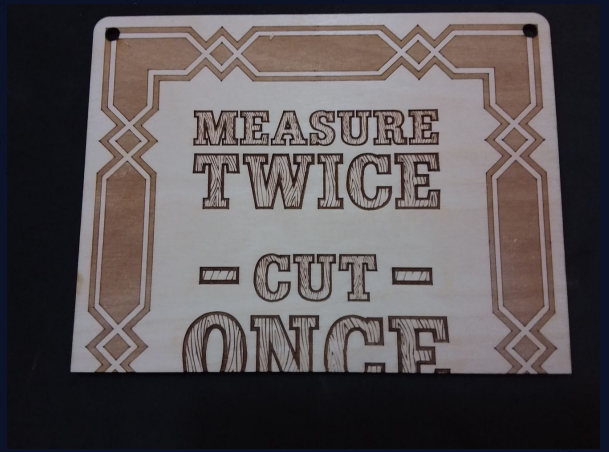
Key performance indicators

Breno Gomes
Senior Solution Engineer

January 2022



Why it is important

Modern software has many interconnected components. In this world the number and types of failures grow proportionally.

Key performance indicators are more than numbers you report during the software development lifecycle.

Metrics provide targets enabling teams to set goals, milestones to gauge progress and insights that help people across the organization make informed decisions.

A good KPI, by definition, should be measurable and trackable.

KPI in the context of feature flags and experimentation

Existing and prospective customers normally go through a Proof of Value (PoV) before adopting or subscribing new products.

A PoV targets the financial, technical and organisational benefits of such implementation.

KPIs are fundamental to define the success criteria. They are used to compare metrics before, during and after the PoV.

Not all firms track KPIs and metrics widely, This guide intends to share indicators frequently adopted during product evaluation.

Use cases

Synchronously toggle features across platforms.

Moving from an old service to a new service or cloud using LaunchDarkly to incrementally accept traffic.

Control the rollout of expensive operations during application updates.

Switching between different UI themes for given sets of client users, allowing controlled testing before release to the store.

Safely test features in production releases to gather feedback.

Flag application elements to test click-through rate.

DevOps

Builds

Commits

Deployment frequency

Approving a feature release

Lead time for changes

Change volume

Change failure rate

Defect escape rate

Mean time to detect

Mean time to identify

Mean time to restore service

Percentage of code covered by automated tests

Application usage and traffic

Application availability

Support tickets

Business

Marketing conversion funnel

Cart abandonment

Cart optimisation

Revenue per customer

Customer retention rate

Profit margin

A/B tests

Unique users

Number of sessions

Page views

Session duration

Geography

Technical

Perceived page load time or mobile interaction

Error rate: frontend, backend and infrastructure

Mobile crash rate

DOM processing

Page rendering

Web browser

Mobile device

API timeout

API error

API latency

Web and Infrastructure

Availability

Throughput

Application response time

Database execution time

Error rate

Memory footprint

CPU utilisation

Network throughput

DevOps Research and Assessment (DORA)

Aspect of Software Delivery Performance*	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per day and once per week	Between once per week and once per month	Between once per month and once every six months
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one day	Between one day and one week	Between one week and one month	Between one month and six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day[a]	Less than one day[a]	Between one week and one month
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0-15%[b,c]	0-15%[b,d]	0-15%[c,d]	46-60%

Source: https://cloud.google.com/blog/products/devops-sre/using-the-four-keys-to-measure-your-devops-performance