

DORA Metrics

The Baseline Metrics For Engineering Excellence

Introduction

Digital Transformation has become a strong reality for organizations large and small. Pre- Covid, pre-2020, Digital Transformation was a buzzword, and a nice-to-have initiative that only the most mature engineering organizations embarked on. But everyone has had to accelerate their digital transformation initiatives to support more remote work, more online access, more on-demand services. And the brunt of the effort of course, falls on the engineering teams.

While the reliance of the business on the engineering teams is growing, so is the demand for accountability and efficiency. To be successful in this new era, Engineering Leaders have to replace gut instinct with a data driven approach to improve efficiency.

What Gets Measured, Gets Done

As engineering leaders try to build and run more efficient teams, they need insights into all aspects of software delivery across various data and tools to make data-informed decisions. In most cases, reporting responsibility falls on overburdened program managers and engineering managers. Some companies can fund an internal tools team to build custom reports for this problem. But most organizations are unable to get data across tools without spending significant time and resources.

Table of Contents

Introduction	1
The 4 DORA Metrics	2
• Software Delivery Lead Time	2
• Deployment Frequency	2
• Change Failure Percentage	3
• Mean Time To Restore (MTTR)	3
The Challenge With Dora Metrics	3
The Propelo Solution	4
Conclusion	6

Due to lack of tools or measurement, many leaders resort to running their teams with a gut feeling or intuition. This must change, and most engineering leaders recognize this.

There need to be some basic software metrics in place that can track the productivity of the engineering team and provide some predictability for the business.

To determine the right software metrics to track the effectiveness and efficiency of software teams, the DevOps Research & Assessment (DORA) research program came up with some findings. In the book *Accelerate – The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations* by Nicole Forsgren PhD, Jez Humble and Gene Kim, four key metrics, known as DORA Metrics, are recommended that can be used to measure Software Delivery Performance.

These four metrics provide a good start to measure the current tempo, rhythm and responsiveness of an engineering organization.

In addition to these DORA metrics, many engineering team leaders have traditionally measured “productivity” metrics such as lines of code, code coverage and time spent on tasks, and related techniques such as story points and time estimates. These metrics have sometimes been used to measure the performance of teams and individuals.

However, these development metrics often give teams a false sense of certainty for the following reasons.

3 pitfalls of the wrong metrics

- › Can be gamed by individuals
- › Do not factor external dependencies
- › May not provide desired predictability and insight

The Four DORA Metrics

- › Lead Time
- › Deployment Frequency
- › Mean Time to Restore
- › Change Fail Percentage

The 4 DORA Metrics

Software Delivery Lead Time

Lead Time is defined as the time from when development starts working on a task to the time the customer gets the feature. It is the time spent in implementation, integration, build, test, and deployment, and does not include the time spent in backlog, prioritization, and in design.

Lead Time helps organizations understand how quickly they can deliver software. It gives you a sense of the efficiency of the development teams. With shorter lead times, you can deploy to production in smaller deployments and more often. This enables faster feedback on what is getting built and allows for quicker course correction. Conversely, longer lead times signify bottlenecks in the development process.

Deployment Frequency

Deployment Frequency is the number of times code or software is deployed to production or “shipped”. Depending on the type of application for certain products it might not be possible or even necessary to ship code frequently to production. Software teams should evaluate the needs of their business and ensure that the velocity of their development process matches their business needs. However, the principles of Lean and Agile can still be applied by delivering software in small batches rather than delivering as large monoliths.

As teams grow, it is critical to find a balance between how much and how often to deploy vs how stable the product is.

Change Failure Percentage

This is the percentage of deployment causing a failure in production. It is the measure of the number of times “a hotfix, a rollback, a fix-forward, or a patch” is required after a software deployment or a service change. This is typically measured using data from the project management, version control and CI/CD systems.

Change Failure percentage can give organizations a sense of how frequently they are shipping out code that causes issues. Ideally, the Change Failure Percentage should be as low as possible indicating good quality code.

Mean Time To Restore (MTTR)

MTTR is the time it takes to restore a failure in production. A failure can be an unplanned outage or a service failure. Measuring MTTR can be trickier than it appears because service failures and outages can be of different types or severity making the reporting of a single MTTR score incorrect and not-valuable. For instance, in many organizations minor and low severity issues are never prioritized to be fixed. Therefore these low priority issues should not be counted towards the final MTTR score.

The first two metrics are a measure of software delivery performance “tempo”, also known as Development Velocity. The second set of metrics measure the reliability and stability of the software that is delivered. A higher development velocity is important but should not come at the expense of quality or the stability of the delivered software.

Accurately measuring DORA metrics can be a challenge for most organizations. Much of the data that is needed to calculate these DevOps metrics lies in various systems across the DevOps toolchain - project management, SCM, CI/CD,

The challenge with DORA metrics

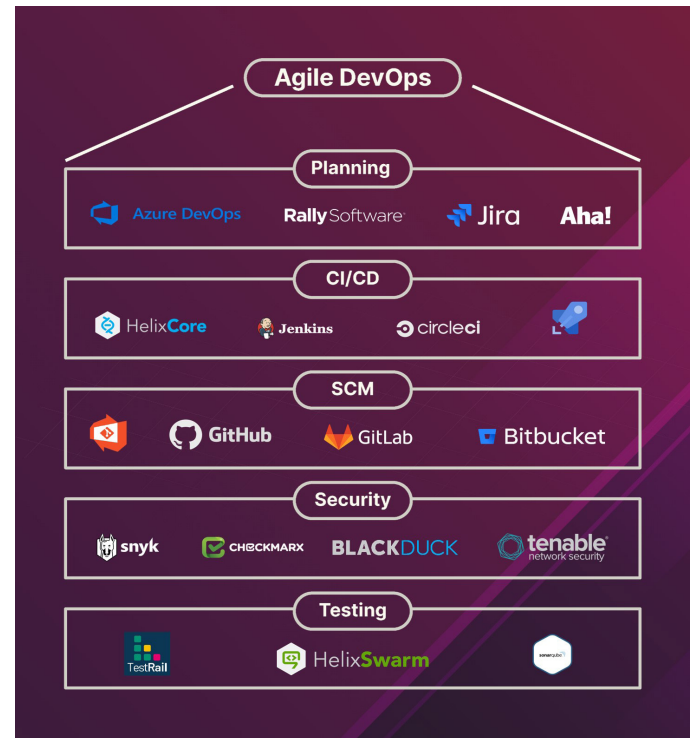


Figure: Tools used across the DevOps landscape

service desk, issue tracking, and other systems. The data must be parsed, broken into spreadsheets, and then correlated to get the right DORA metrics.

As an example, consider an organization that uses Jira for planning, GitHub for SCM, Jenkins for CI/CD, ServiceNow for service desk, PagerDuty for production monitoring, and various other tools for testing, security, etc. This is a very common scenario in many organizations where they select different tools that meet their needs for different purposes.



FOR EVERY METRIC, THERE IS A COUNTER-METRIC

JOE CHEN

VP OF ENGINEERING, BROADCOM

To get a metric such as Lead Time, you need to correlate the data from Jira with data from GitHub and Jenkins, so you can accurately understand how much time was required from the start of the task till it made it to Production.

Similarly, for accurate MTTR, you need to correlate data from PagerDuty back to ServiceNow and then to Jira. This can be quite a challenge.

For many organizations, this can take several days or even a few weeks. Even with such effort, the outcome is brittle due to the following reasons:

- **Multiple data sources:** Data for the metrics comes from multiple systems across the DevOps toolkit - project management, SCM, CI/CD, service desk, issue tracking, etc.
- **Data Hygiene:** Even if you are able to get data from different sources, correlating it can be difficult due to missing data. Certain fields might not be populated correctly. For example, tickets might not be “groomed” properly. A change in format, data.

The Propelo Solution

Propelo can help organizations solve this challenge. Propelo integrates with over 40+ tools across the DevOps toolchain, and provides 150+ out-of-the-box software metrics and insights into the performance of engineering organizations.

Propelo is purpose-built to address the data challenge in engineering and has highly sophisticated mechanisms to automatically manage the twin challenges of (a) a plethora and ever-growing list of data sources and (b) the data hygiene challenge.

Software Delivery Lead Time

Lead Time is a great metric to track, especially if looking at the trend over time. It shows whether there are any issues, and if things are getting better or worse. At the end of the day, how quickly you are able to respond to the business. However, what is more important is to get further breakdown of the different stages.

- Are there any bottlenecks? What stage are they in?
- Are PRs taking long to Review? If so, why?
- What is the Merge Time?

Propelo automatically correlates data across the Project Management, SCM, CI-CD and Deployment systems, and provides accurate Lead Time information. It provides a detailed breakdown of time spent in each stage so users can drill-down and check which activity or step within the stage takes the most time to complete.

Mean Time To Restore (MTTR)

MTTR shows how long it takes an organization to recover from a failure in production - how quickly those issues can be resolved. Production issues can impact multiple customers, and directly affect revenue and the business. Hence, organizations should strive to reduce their MTTR as much as possible.

To accurately track MTTR, you need data from your project management, service desk as well as production monitoring systems.

Propelo can integrate with many systems such as Jira, Salesforce service desk, or PagerDuty to measure the MTTR. Propelo lets you take a scalpel approach to precisely measure certain types of failures across all your teams and gives you a much standardized view of MTTR across all your teams.



**PROPELO HELPS US OPERATIONALIZE
ENGINEERING METRICS AND KPIS AND AIDS
IN MAKING DATA DRIVEN DECISIONS**

VISHAL SAXENA

VP OF CLOUD ENGINEERING, AKTANA



Deployment Frequency

Deployment Frequency helps organizations set their release schedules. It sounds like a simple metric and measures the number of “deployments” for a given time period. But the meaning of “deployment” varies from team to team.

Not every organization will want to or need to deploy very quickly or frequently. SaaS organizations may need to deploy very frequently. On the other hand, for certain business applications, deployment frequency of once or twice a year might be sufficient - their customers may not be happy with frequent changes.

Propelo is flexible to accommodate various types of deployment models and technologies. Whether you are pushing a package or container or turning on feature flags Propelo can measure your Deployment Frequency.

Change Failure Percentage

Change Failure Percentage is defined as the number of times a “failure” happens after a change is made. A “Change” can be either a modification in code or infrastructure. The definition of “failure” varies between customers, service, or software.

With Propelo, the customer can define the meaning of a “failure” whether that is captured in PagerDuty or Jira, and also control the meaning of a “hotfix” or “rollback” or “fix-forward” whether it is in the SCM(GitHub/Gitlab) or CI/CD (Jenkins).

Propelo aggregates data from over 40 tools in the DevOps Toolchain pipeline such as Jira, Github, GitLab, Jenkins, PagerDuty etc. Propelo centralizes the right data from these tools for easy consumption and reporting on DORA Metrics and many other metrics. Propelo also provides visibility into how the development velocity impacts your software quality.



PROPELO MAKES IT EASY FOR DEVELOPERS TO SHIFT-LEFT SECURITY BY PROVIDING THE RIGHT INFORMATION A TIMELY MANNER, WITHIN THE TOOLS DEVELOPERS USE

KHUSHBOO KASHYAP
RISK MANAGEMENT LEADER, RUBRIK

Conclusion

DORA Metrics are the foundation to understanding the efficiency and effectiveness of your engineering organization. Once you have these basic North Star metrics established, you need deeper insight into the root cause of the slowdowns and quality issues.

With the right software metrics, you can make data-driven decisions and demonstrate alignment with the business towards customer-centric outcomes.



Propelo is a venture-backed silicon valley company that serves customers from Fortune 100 companies to Series B startups. Propelo provides out-of-the-box software metrics and insights into the performance of engineering organizations.

Learn more about Propelo at propelo.ai or contact us at sales@propelo.ai



Propelo Inc.
700 S Bernardo Ave, Ste 103
Sunnyvale, CA 94087
sales@propelo.ai