HAYSTACK

Make decision with data instead of of gut feeling

Goals

Organizations have 2 goals, increase revenue, decrease churn. Each department should aim for these goals, however, these numbers are too abstracted from each departments work. Each department should create their own metrics which they have full control over those metrics.

For engineering organizations we have talked to > 200 engineering leaders about metrics which has also been independently validated by DORA project and Accelerate Book. What makes a successful engineering team is ***# of Successful Iterations***.

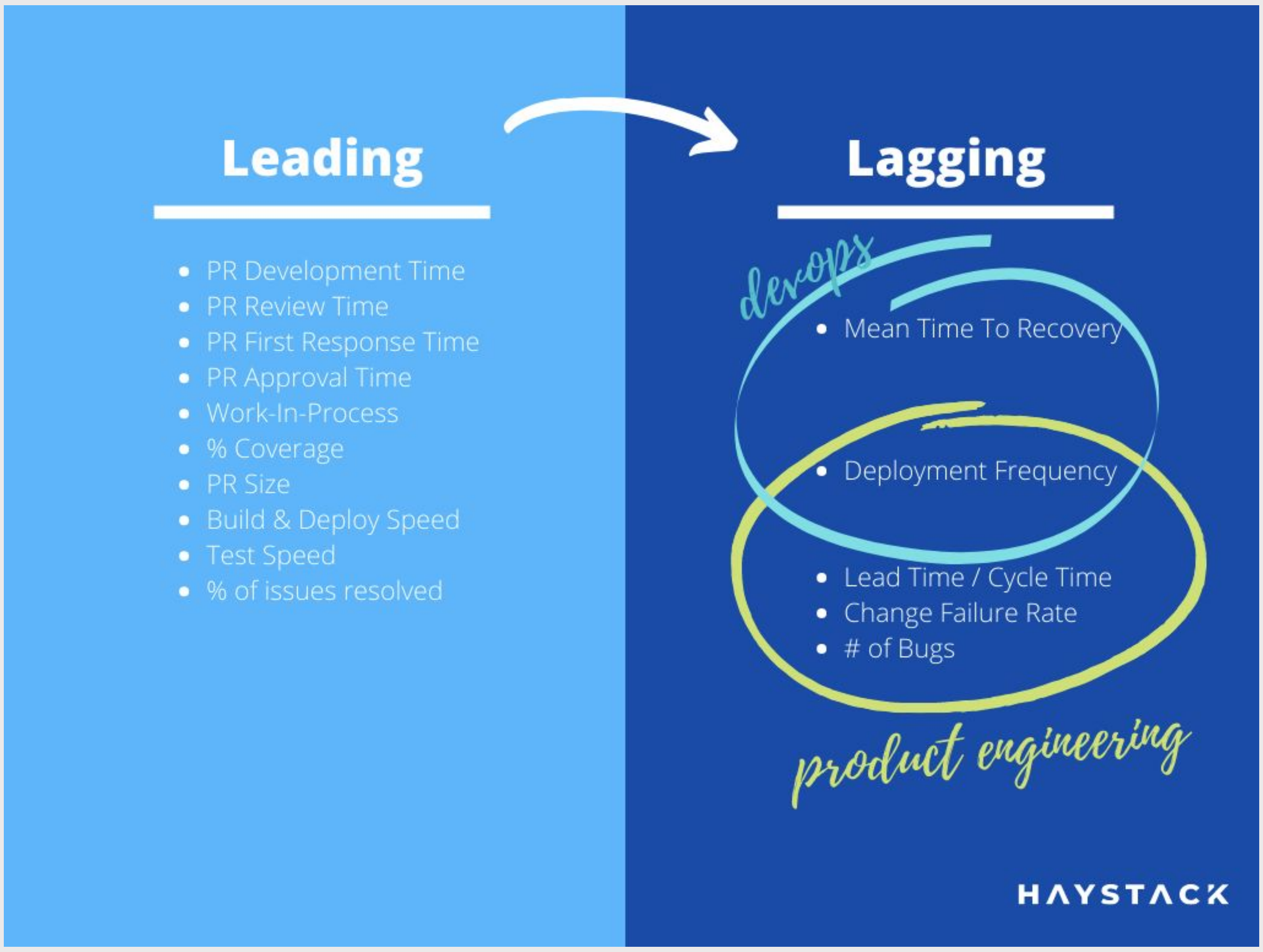
The faster the team can iterate, the more likely the company to build the correct product and be competitive in the market.

There are 2 proxies for iteration speed, Speed and Quality. If a team only focuses on speed, the engineers would stop writing tests to delivery faster and break things. A broken product is not a "Successful Iteration", thus the team should focus Speed and Quality together.

Engineering Metrics To Track

The team must be in full control of the metric they track. It is vital for the team to have an "end-to-end" control of each metric they track.

There are distinct metrics for DevOps and for Product Engineering since each team has different goals.



For DevOps teams the key metric is ***deployment frequency***. The other metric to monitor is ***mean time to recovery (MttR)***.

Product engineering writes the code, owns bugs in functionality and can be affected by bottlenecks in the development cycle. To deliver faster, ***deployment frequency*** and **cycle time** are the key metrics. To improve quality, ***number of bugs*** is the key measure. The number of hotfixes needed can be found by the ***change failure rate***.

All other metrics that are tracked are considered Leading Metrics. These metrics are not goals. Rather, they are used for finding bottlenecks, leading indicators before the lagging indicators are effected.

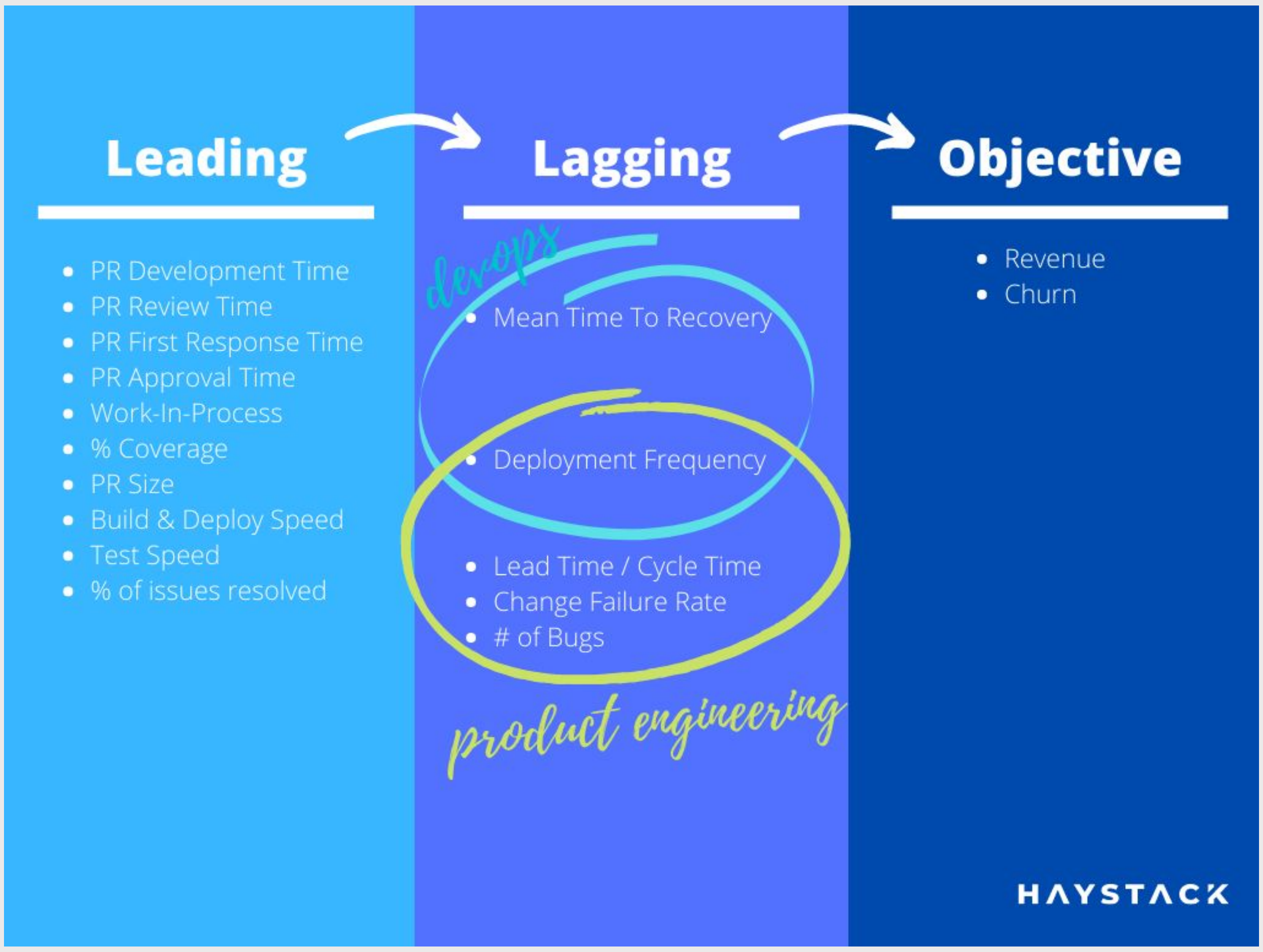
Case Study

Andrew an Engineering Manager, who wants to move faster, and ship more features. He knows the team is moving slowly and that the codebase has lots of technical debt. Rather than going with gut feeling and trying to fix technical debt of potentially low-impact but annoying code for developers, to move faster, he creates a goal for decreasing Cycle Time from ~6 days to ~3 days while keeping quality the same. This concrete goal allows Andrew's team to focus on the bottlenecks of their process which includes First Response Time, Review Time and lack of CI automation.

“We can experiment and make more changes - while knowing we’re headed in the right direction”

In 3 months, after multiple iterations Andrew has decrease their Cycle Time from 6.4 days to 2.2 days, increasing their Deployment Frequency from bi-weekly to 2 times per week. The same size team is able to complete 42% more JIRA tasks and 58% more Pull Requests saving $1.3M worth of engineering resources for a team of 30 engineers.

Metrics in Haystack



Leading

- PR Development Time
- PR Review Time
- PR First Response Time
- PR Approval Time
- Work-In-Process
- % Coverage
- PR Size
- Build & Deploy Speed
- Test Speed
- % of issues resolved

Lagging

- Mean Time To Recovery
- Deployment Frequency
- Lead Time / Cycle Time
- Change Failure Rate
- # of Bugs

Objective

- Revenue
- Churn

Haystack is primarily for product engineering. Haystack provides:

- Cycle Time
- Deployment Frequency
- Change Failure Rate
- and 20+ leading metrics

For DevOps teams, we recommend using tools like PagerDuty to track Mean Time To Recovery (MttR) alongside Deployment Frequency provided by Haystack.

Resources

DORA DevOps Research & Assessment State of DevOps 2019

Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations

Note: Our research is based on Product Engineering Teams. It isn't validated for R&D Teams.