As you get to know CD pipelines in general, you'll eventually develop some intuition about how healthy your CD pipeline is or is not. You'll also be able to easily spot problem areas and be able to explain how they impact the whole process.

Whether you're still waiting on that intuition to form or if you've had it for a long time, it's helpful to have metrics to guide decisions and demonstrate impact empirically.

**Healthy Continuous Delivery**

Let's take a look at a few metrics I use when I want to demonstrate the level of health or impact of a CD pipeline:

| **Metric** | **Formula** | **Impact** |
| --- | --- | --- |
| Lead Time to Production | (*Time at Successful Prod Deployment*) - (*Time at Completion of Feature Grooming*) | Shows how CI/CD is impacting overall delivery time, from the point the team hears about the feature to the point at which it is done (deployed to production). Easy metric to collect if using task management system to track feature grooming and deployments. |
| Rollback Rate | (*Total Rollbacks*) / (*Total Deployments*) | Shows the quality of our deployments. Of course, rate should be low because previous stages should filter out defected builds. This metric is a leading indicator for the confidence of the business in the dev team's ability to delivery. |
| Time to Failure | (*Time at Failure Discovery*) - (*Time at Build Start*) | Shows how quickly we find failures. The lower the better. |
| Production Uptime | (*Total Production Working Time*) / (*Total Time*) | Shows the amount of time we are taking production down because of botched deployments or due to our chosen deployment strategy. |
| Failed Pipeline Cost | Various calculations including job run time and resources created | Shows the estimated amount of money spent on a failed build. Encourages us to put cheaper jobs earlier in the pipeline. |