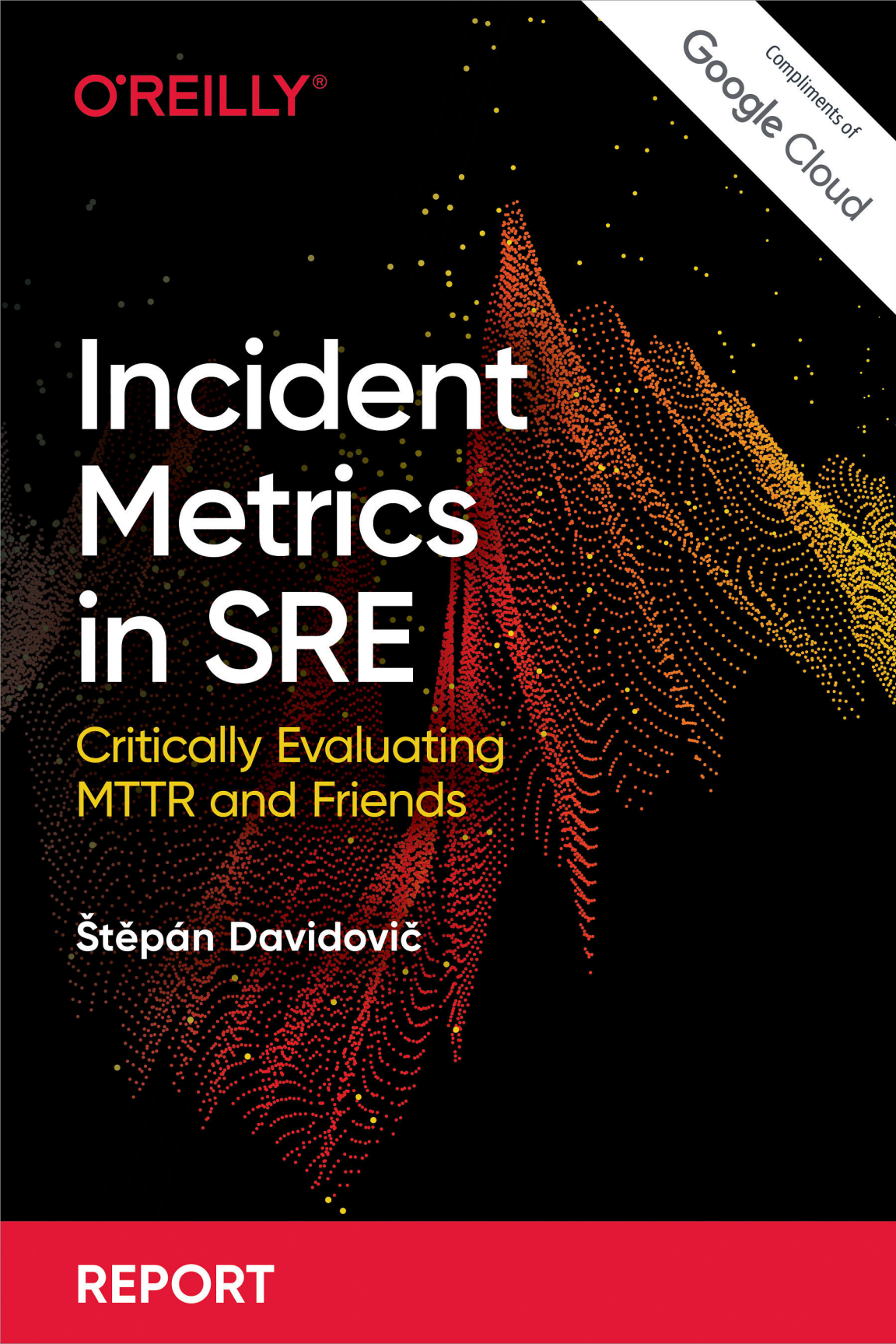
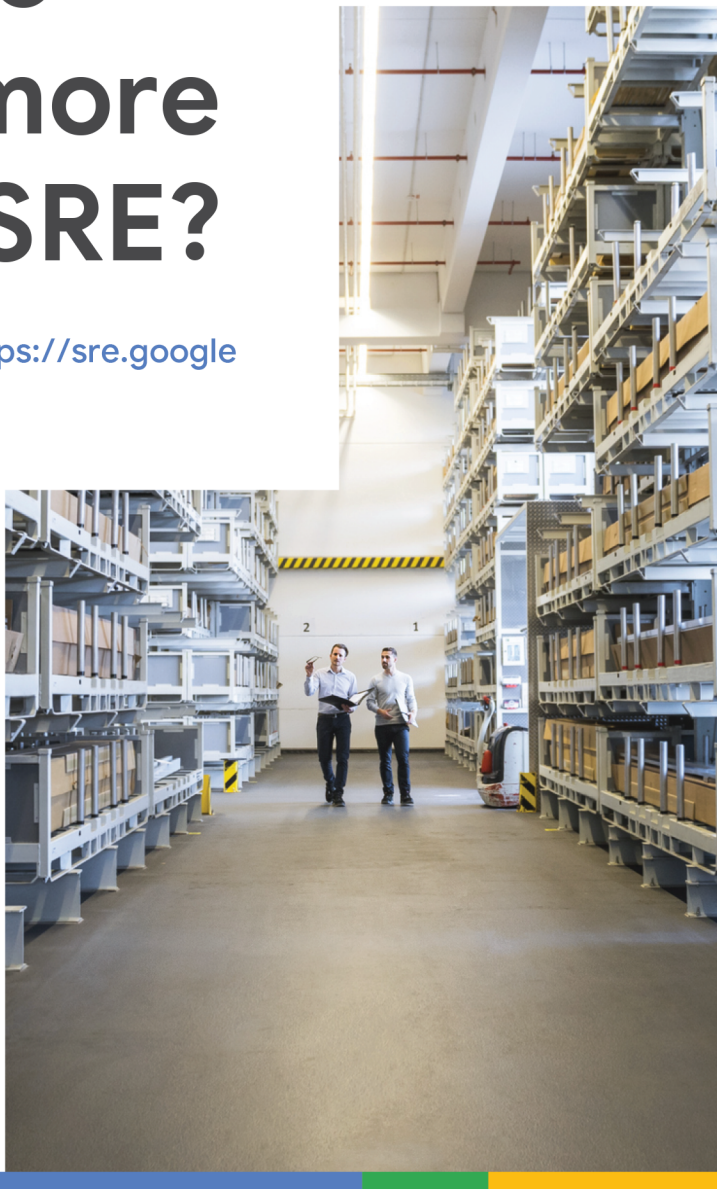

O'REILLY®
Compliments of
Google Cloud
Incident Metrics in SRE
Critically Evaluating MTTR and Friends
Štěpán Davidovič
REPORT

Google Cloud

Want to know more about SRE?

To learn more, visit https://sre.google



Incident Metrics in SRE

Critically Evaluating MTTR and Friends

Štěpán Davidovič

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

Incident Metrics in SRE
by Štěpán Davidovič

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Acquisitions Editor: John Devins
Development Editor: Virginia Wilson
Production Editor: Kate Galloway
Copyeditor: Shannon Turlington
Proofreader: Holly Bauer Forsyth
Interior Designer: David Futato
Cover Designer: Kenn Vondrak
Illustrator: Kate Dullea

March 2021: First Edition

Revision History for the First Edition
2021-03-19: First Release

978-1-098-10313-2

[LSI]

Table of Contents

Incident Metrics in SRE

Abstract

Measuring improvements as a result of a process change, product purchase, or technological change is commonplace. In reliability engineering, statistics such as mean time to recovery (MTTR) or mean time to mitigation (MTTM) are often measured. These statistics are sometimes used to evaluate improvements or track trends.

In this report, I use a simple Monte Carlo simulation process (which can be applied in many other situations), as well as statistical analysis, to demonstrate that these statistics are poorly suited for decision making or trend analysis in the context of production incidents. To replace these, I propose better ways to achieve the same measurements for some contexts.

Introduction

One of the key responsibilities of a site reliability engineer (SRE) is to manage incidents of the production system(s) they are responsible for. Within an incident, SREs contribute to debugging the system, choosing the right immediate mitigation, and organizing the incident response if it requires broader coordination.

But the responsibility of an SRE is not limited just to managing incidents. Some of the work involves prevention, such as devising robust strategies for performing changes in production or automatically responding to problems and reverting the system to a known-safe state. The work also includes mitigation, such as better processes for communication, improvements in monitoring, or development of tooling that provides assistance during debugging of

the incident. As a matter of fact, there are products dedicated to improving the process of incident response.

You want your incidents (if you must have any at all!) to have as little impact as possible. That often means short incident durations, which I'll focus on here. Understanding how a process change or a product purchase shortens the durations of incidents is important, especially if there are real costs associated with the incidents. However, we can't jump to conclusions from a single incident; an analysis of a whole body of incidents is required.

A quick search with your favorite search engine might reveal many articles that state that MTTx metrics (including mean time to recovery and mean time to mitigation) should be considered the key performance indicators of your service's reliability. These articles are sometimes authored by high-profile companies with a track record of delivering their services reliably or providing reliability-related tooling. But are these metrics good indicators of reliability? In fact, are they indicators that can even be used *at all*? How can you tell?

When applying MTTx metrics, the goal is to understand the evolution of the reliability of your systems. But the reality is that applying these metrics is trickier than it seems, and these popular metrics are dangerously misleading in most practical scenarios.

This report will show that MTTx is not useful in most typical SRE settings, for reasons that apply to many summary statistics and do not depend on company size or strictness of enforcement of production practices. Whatever metric you choose to use, it is important to test that it can give you robust insights regardless of the shape of the incident duration distribution. There may not be a "silver bullet" metric that could serve as a general-purpose replacement where MTTx is currently considered, but you may have more success in measurement by tailoring the metric to the question at hand. I'll end this report by exploring some alternative methods for achieving these measurements.

Incident Life Cycle and Timing

Before analyzing incidents in aggregate, let me quickly introduce some language. Language may vary from company to company, but the underlying principles should be relatable.

Figure 1 demonstrates a simple timeline model of an incident that I'll be using going forward. In this model, the incident shows these key stages:

First product impact
: The first moment of severe impact to the product

Detection
: When the system's operator becomes aware of the ongoing problem

Mitigation
: When there is no longer severe product impact but the system might still be degraded in some way

Recovery
: When the system has been fully recovered into normal operation; recovery and mitigation are often the same stage, but sometimes they differ

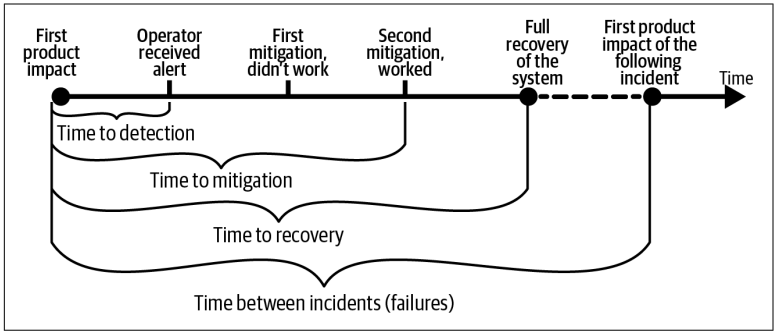


Figure 1. Simplified timeline of an incident, with key points highlighted.

I will be analyzing the incident durations and drawing conclusions about the usefulness of applying statistics to them. There are several publicly available repositories aggregating incident retrospectives, showing timelines and key events.[1] In this analysis, I am specifically looking at the time window during which an incident impacts users.

The incident timeline model in Figure 1 simplifies reality, as all models do. There are problems with what's been called "shallow

1 See, for example, "A List of Post-mortems!" and "Postmortem Index".

incident data."[2] An example problem with this model, in the context of this analysis, is the question, "Do you consider an incident mitigated if you've removed impact for 90% of users but 10% are still impacted?" What if 5% are still impacted, or 20%? Using this model, you need to make a binary decision. There is valid criticism that classification like this is often done subjectively and following inconsistent principles.

You may not mind these inaccuracies in your incident data. For many practical purposes—including the aggregate analysis I'll focus on—more precision is not necessary, and achieving it might turn out to be costlier than the potential benefits of improved data quality. Statistician George Box famously said, "All models are wrong, but some are useful," and I believe this model can be used to understand the viability of MTTR and similar metrics.

MTTR, MTTM, Oh My!

An incident might provide data about it, but you want to look at an aggregate. *Mean time to recovery (MTTR)* is a term often used in the industry.[3] Terms such as *mean time between failures (MTBF)* may also sound familiar, especially when considering the reliability behavior of hardware components.

MTTR, in this case, is defined as the mean duration calculated as time of recovery minus time of first product impact across all incidents eligible for such analysis. Similarly, *mean time to mitigation (MTTM)* is the mean duration calculated as time of mitigation minus time of first product impact.

Distribution of Incident Durations

To analyze the behavior of incident-duration statistics, you need data—ideally data from diverse settings to avoid drawing conclusions from just one company or just one product. I collected the public incident status dashboard data from three well-known internet companies (sized in the range of around one to two thousand

2 John Allspaw, "Moving Past Shallow Incident Data", Adaptive Capacity Labs, March 23, 2018.

3 "Mean time to recovery", Wikipedia.

employees). The distribution of incident durations is plotted in Figure 2.

I do not discriminate by incident type: if the company thinks the incident was worth publishing for end user consumption, I use it. The incidents' durations represent user-facing communication from the first impact to last. I will call it *time to recovery* for simplicity, but I acknowledge the imprecision. As *time to recovery* and *time to mitigation* are often the same durations, and I found them to both follow a similar distribution, this imprecision doesn't impact the analysis.

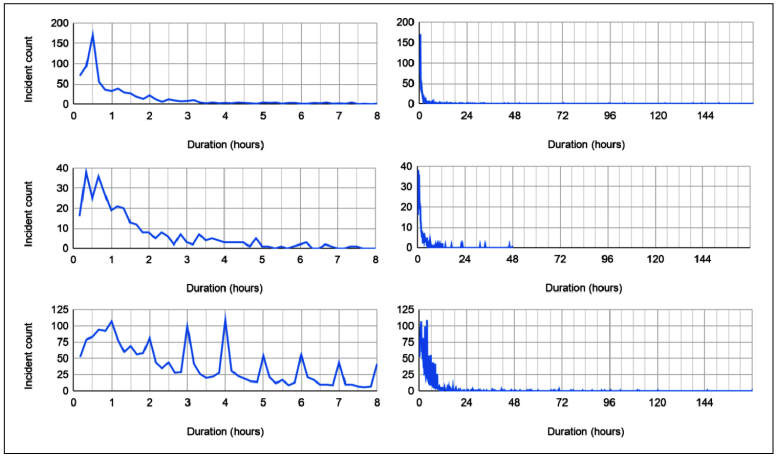


Figure 2. Distribution of incidents' durations with incident counts. Rows are, in order, Company A (N = 798; 173 in 2019), Company B (N = 350; 103 in 2019), and Company C (N = 2,186; 609 in 2019). Columns represent each company over a short and long time frame to show the tail of the distribution.

I also collected incident data from Google (Figure 3), and Google's data set—in my analysis—represents a very large company focused on internet services. The Google data set was collected over a one-year period—shorter than any of the data sets shown in Figure 2—but it also contains internal incidents (e.g., those impacting only developer productivity). I cannot share the numbers, but Google's incident data set is several times larger than any of the three public data sets, as expected given the company size.

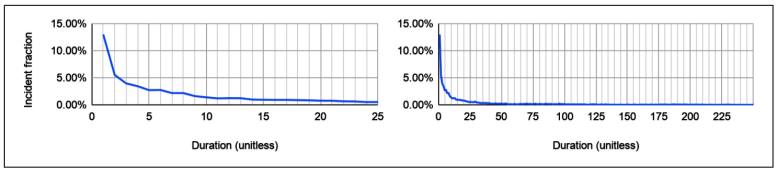


Figure 3. Distribution of incident durations at Google from 2019, obfuscated.

The key observation is that the incidents follow a positively skewed distribution in each case, with the majority of incidents resolving quickly. Figure 4 shows that the distributions roughly approach lognormal (or gamma) distribution, but I have not attempted probability distribution fitting of the empirical data. All data sets show a huge variance in the incident durations. This matches my experience: most incidents are resolved fairly quickly, but some are more complex and lingering events, and a handful are disastrous "black swan events."[4]

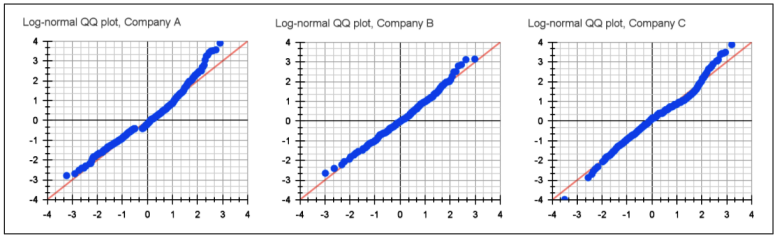


Figure 4. Lognormal Q-Q plot of the incident durations, showing how they approach lognormal distribution.[5] Note that this cannot be used to conclude that lognormal distribution is the best fit. It is included for illustration only.

I excluded incidents shorter than three minutes and longer than three days from the public data sets, which were around 1–2% of each data set. Manual inspection of arbitrarily chosen incidents from the data sets confirmed that these outliers are valid, and I also know from incident retrospectives that there are impactful multiday incidents even longer than that.[6] But I felt that including unusually

4 Laura Nolan, "What Breaks Our Systems: A Taxonomy of Black Swans" (video), SRECon19 Americas, March 25, 2019.

5 "Normal probability plot", Wikipedia.

6 See "A List of Post-mortems!" and "Postmortem Index".

long incidents—even if they happen in practice—could cast avoidable doubt on the analysis.

From this empirical data, you see the distribution of incident durations, but it would be wrong to judge the companies' reliability practices from the incident counts or their durations. These data sets come from companies with different business models, reliability needs, and incident communication practices.[7]

Analyzing Improvements

All right, you've got a clear picture of what your incident durations look like. Now it's time to make your incidents shorter!

Imagine you are offered a reliability-enhancing product that helps you shorten the mitigation and resolution time of incidents by 10%. For example, a daylong incident shrinks to a little over 21 and a half hours. You are offered a trial to evaluate the product. How can you tell that the product delivers on its promises? This report explores the use of MTTR and similar metrics, so that's the metric I'll use.

I chose this artificial scenario intentionally because it applies to many real-world scenarios. Whether you are changing a policy, developing software, or introducing a new incident-management process, the objective is often to shorten your incidents and try to evaluate the success of the change.

Deciding MTTR Improvements

So how are you going to test that the product has actually delivered on its promises? An intuitive test is pretty straightforward: "If every incident's duration decreases as stated, we are able to tell the improvement in the MTTR metric."

This is still quite imprecise, though. What does "we are able to tell the improvement" mean? At the end of the day, you often need to make a binary decision. In this scenario, you need to decide whether the product is successful and purchase it or not.

7 Notice, for example, that Company C has incident durations often aligned with whole hours, and this manifests as spikes on the graph.

To gauge whether a product delivers on its promise of shortening the incident duration by 10%, you could set a threshold of a 10% decrease in MTTR compared to before you began using the product. A looser test is to require any improvement at all. You would decide that the product is successful if you see any shortening of incidents at all, regardless of magnitude.

You want to have a crisp understanding of how you expect the metric to behave and be confident that the chosen metric (such as MTTR) faithfully measures what you want it to measure. There would be real and severe risks and costs if you were to rely on a poor metric. These can be direct, such as purchasing a product for the wrong reasons, but they can also be very subtle. For example, your employees' morale may suffer upon realizing that their incident-management efforts are evaluated using unproven or suspect metrics.

Simulating MTTR in Parallel Universes

You live in your one universe, so you get only one go at evaluating the product in this scenario. But intuitively, you know that incidents vary, and you want to be reasonably sure that what you're seeing isn't just a random fluke.

To become reasonably sure whether that's the case, you can do a Monte Carlo simulation of the improvement process.[8] Assume that the incidents follow the empirically observed distribution of the obtained data sets and evaluate what kinds of improvements you would see after a certain number of incidents—and with what confidence level.

The simulation process is simple:

1. Randomly draw two samples, with size N_1 and N_2 (where $N_1 = N_2$ to get a perfect 50/50 split), from the empirical distribution of incident durations.
2. Modify the incident durations in one of the populations, in this case by shortening it by 10%.

8 A simulation done by repeated sampling to model a behavior—in this case, the behavior of incident resolution times.

3. Calculate MTTR for each of the groups, i.e., $MTTR_{modified}$ and $MTTR_{unmodified}$.
4. Take the difference, observed improvement = $MTTR_{unmodified} - MTTR_{modified}$. (A negative difference means MTTR is worsening.)
5. Repeat this process 100,000 times.

You are doing two samples, with size N_1 and N_2 where $N_1 = N_2$. The 50/50 split gives the strongest analysis; I will briefly touch on *why* in "Analytical Approach" on page 18.

Simply put, you visit thousands of parallel universes where you simulate that the product delivers on its promises and compare the resulting MTTR against the incidents that weren't treated. Mechanically, this can be done using tools such as a Python script and a CSV file with the data or a sufficiently capable SQL engine, and does not require any specialized tooling or additional knowledge.

You are now operating on probabilities, so you need to add one more condition to your test: some tolerance of random flukes. Let's say that you're tolerating up to 10% of these parallel universes to mislead you. More formally, you might recognize this as requiring statistical significance $\alpha = .10$. This is arguably a generous value.

Scenario simulation and evaluation

For this scenario, I picked two samples of incidents of equivalent sizes (N_1 and N_2, where $N_1 = N_2$). I chose $N_1 + N_2$ equal to the number of incidents in the year 2019 (Table 1).[9] This was 173, 103, and 609 incidents for Company A, B, and C, respectively.

Table 1. Incident count, mean, and variance across the three data sets.

	Company A	Company B	Company C
Incidents (all)	779	348	2157
Incidents (2019)	173	103	609
Mean TTR	2h 26m	2h 31m	4h 31m
Standard deviation	5h 16m	5h 1m	6h 53m

9 As of late summer 2020, I felt that just using the last 12 months could lead to an unusual data set, swayed by world events.

Having performed the simulation, I plotted it out to see what happens (Figure 5).

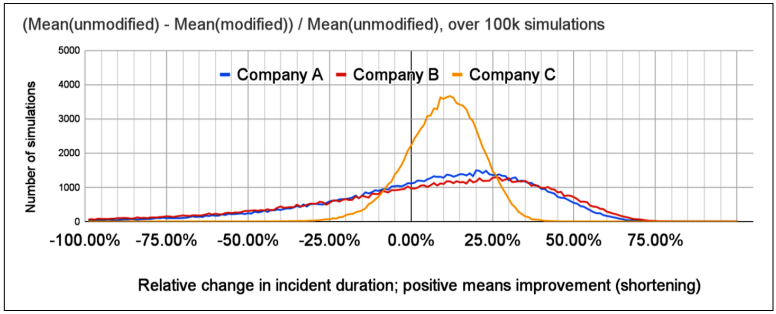


Figure 5. Distribution of simulated changes to MTTR if improvement actually happened, as relative improvement.

Yikes! Even though in the simulation the improvement always worked, 38% of the simulations had the MTTR difference fall below zero for Company A, 40% for Company B, and 20% for Company C. Looking at the absolute change in MTTR, the probability of seeing at least a 15-minute improvement is only 49%, 50%, and 64%, respectively. Even though the product in the scenario worked and shortened incidents, the odds of detecting any improvement at all are well outside the tolerance of 10% random flukes.

Change in statistic without changing incidents

To make matters worse, there's a good chance that you'll see a significant reduction in your MTTR that goes *beyond* what the product promised. This can be demonstrated more clearly by running the same simulation as before, but in this case, the product does nothing to change the incidents. Replace Step 2 with `new_duration = old_duration`.

And sure enough, Figure 6 shows that there's a 19% chance that there is a half-hour improvement (or better) of MTTR in Company A (and 23% for Company B, and 10% for Company C)…even though in this simulation, you did not change anything about the incidents.[10] In other words, even if the hypothetical product did

10 For this particular situation, where incidents were shortened by 10%.

nothing for you, you would think it had and decide to purchase the product.

NOTE

A cynical response to this finding would be to start selling a fake incident-shortening product. The business would set its prices to be profitable when a fraction of customers see the advertised improvement just by chance and they purchase the product. I do not endorse such a business plan. However, it definitely highlights the problems that can stem from using low-quality metrics.

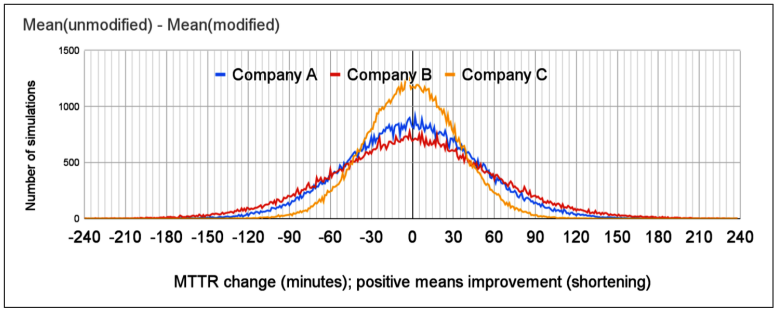


Figure 6. Distribution of simulated changes to MTTR if there was no change to the incidents.

We've learned that even without any intentional change to the incident durations, many simulated universes would make you believe that the MTTR got much shorter—or much longer—without any structural change. If you can't tell when things aren't changing, you'll have a hard time telling when they do.

Changing the Thought Experiment

The previous scenario started by assuming that there's a product that can reduce the incident duration and you want to understand how that change would manifest in the observed MTTR. But in practice, forecasting and modeling a prospective improvement is very difficult.

That can be solved by turning the question around, as I've already done. Instead of looking for a particular improvement, look at the change in the observed MTTR (or other statistics) if there's no structural change to your incidents. In other words, your incident

durations keep coming from the same distribution (not changed by any incident-handling improvement), and you evaluate the typical change in the statistics.

From here on, I will simplify the discussion and focus only on the scenario of showing what the change in MTTR can be if nothing changed the incidents, foregoing the analysis of improvements. Consequently, what's most interesting is the *shape* of the resulting distribution: put bluntly, we want to know how flat it gets.

Better Analysis by More Incidents

You might have an intuition about why you see such a wide range of possible changes of the observed MTTR: there's too much variance in the incidents. There's a statistical basis for this intuition.

The central limit theorem tells us that the distribution of sample sums tends toward a normal distribution as the number of samples increases.[11] You can see some evidence of that in the previous analysis (such as in Figure 6), where the distributions are somewhat normal looking. While you cannot automatically assume that the resulting distribution is always normal (more on that later), it also means that the variance converges to

$$\sigma^2_{sample\ \ mean} = \frac{\sigma^2_{incidents}}{N}$$

in the limit. In line with your intuition, this indicates that the variance seen in the observed MTTR value decreases as the sample size (i.e., the incident count) increases. That can easily be demonstrated. Table 2 shows 90% confidence intervals for MTTR for several incident counts.

Recall that you are drawing two samples from the incident-duration distribution. So if you are trying to find out how good an analysis you can make with N incidents total, you draw two samples with size N_1 and N_2, where $N_1 = N_2$.

11 See the chapters "Sampling Distribution of the Mean," "Sampling Distribution of Difference Between Means," "Testing of Means," and others in *Online Statistics Education*, project leader David M. Lane, Rice University.

Table 2. 90% confidence intervals for difference of two MTTRs calculated from two randomly sampled sets of incidents ($N_1 = N_2$) across 100,000 simulations.

	Company A	Company B	Company C
Mean TTR of original data	2h 26m	2h 31m	4h 31m
Incidents in 2019	173	103	609
$N_1 + N_2 = 10$	mean difference ≅ 0 90% CI [−5h41m; +5h42m]	mean difference ≅ 0 90% CI [−5h25m; +5h18m]	mean difference ≅ 0 90% CI [−7h4m; +7h15m]
$N_1 + N_2 = 100$	mean difference ≅ 0 90% CI [−1h44m; +1h44m]	mean difference ≅ 0 90% CI [−1h39m; +1h39m]	mean difference ≅ 0 90% CI [−2h16m; +2h16m]
$N_1 + N_2 = 1{,}000$	mean difference ≅ 0 90% CI [−33m; +33m]	mean difference ≅ 0 90% CI [−31m; +31m]	mean difference ≅ 0 90% CI [−43m; +43m]

As the number of samples goes up, the standard deviation goes down, and that improves your ability to detect smaller and smaller changes as significant. In the original scenario, you were evaluating a product offering a 10% reduction in the incident duration; even at one thousand incidents, that would still fall into the 90% confidence interval. In no case do you get to a confident value even with a year's worth of data.

Similar results for Company A and Company B are coincidental. The two companies are providing very different services, but as shown, they happen to have similar mean incident duration and standard deviation. If you were to consider incidents from just a single year, the data differs a lot: Company A's mean incident duration is 4h 35m, while Company B's is 2h 38m. Their other statistics, such as median, also differ more than their means.

Even with a high number of incidents (higher than the yearly tally), the variance is still too high, and Figure 7 shows that even a sizable change of observed MTTR stays in the 90% confidence interval. While increasing the number of incidents would help get a better signal, it would go against the overall objective of reliability engineering.

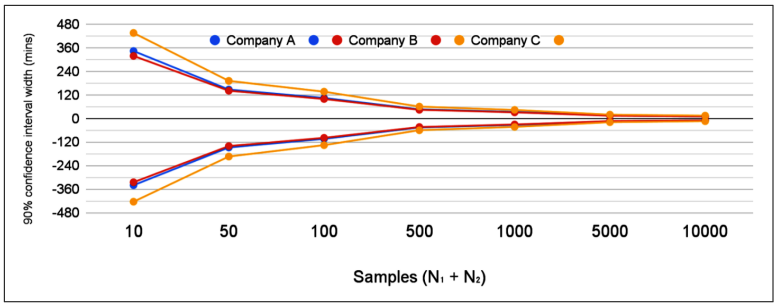


Figure 7. Decrease of width of 90% confidence interval as sample size increases.

Going Beyond Your Means

A frequent and rightful point of criticism of the arithmetic mean is that it's sensitive to outliers. Even though the most egregious outlier incidents have been excluded (recall that incidents shorter than three minutes or longer than three days were excluded), the point still stands. You might easily consider other statistics, so let's explore them.

Median and percentiles

Median is frequently used to avoid a few far outliers skewing the resulting measure too much, and it can be used here, too—most incidents don't last several days.

It's important to remember that if you're going to analyze medians, you also need to adjust what you're looking for. If you're looking for any kind of relative difference, it should be relative to the median. Testing against a fraction of MTTR, for example, would be quite misleading.

As Table 3 shows, even at N = 1,000 incidents, the generous 90% confidence interval is still large relative to the median statistic and encompasses the discussed target of 10% median TTR. The difficulty is not specific to the "mean" in MTTR; median TTR isn't helping us either.

Table 3. 90% confidence intervals for difference of two median TTRs calculated from two randomly sampled sets of incidents ($N_1 = N_2$) across 100,000 simulations.

	Company A	Company B	Company C
Median TTR of original data	42m	1h 7m	2h 50m
Incidents in 2019	173	103	609
$N_1 + N_2 = 10$	mean difference ≅ 0 90% CI [−1h46m; +1h46m]	mean difference ≅ 0 90% CI [−2h13m; +2h12m]	mean difference ≅ 0 90% CI [−4h8m; +4h7m]
$N_1 + N_2 = 100$	mean difference ≅ 0 90% CI [−29m; +29m]	mean difference ≅ 0 90% CI [−29m; +29m]	mean difference ≅ 0 90% CI [−1h20m; +1h19m]
$N_1 + N_2 = 1{,}000$	mean difference ≅ 0 90% CI [−11m; +11m]	mean difference ≅ 0 90% CI [−9m; +9m]	mean difference ≅ 0 90% CI [−29m; +29m]

The higher percentiles, such as 95th percentile, perform much worse. Intuitively, this makes sense. The higher percentile incident duration will be swayed by the worst incidents, which are also the rarest. As a result, they see a very high variance. A few values are listed in Table 4.

Table 4. 90% confidence intervals for difference of 95th percentile TTRs calculated from two randomly sampled sets of incidents ($N_1 = N_2$) across 100,000 simulations.

	Company A	Company B	Company C
95th percentile TTR of original data	10h 45m	8h 48m	12h 59m
$N_1 + N_2 = 100$	mean difference ≅ 0 90% CI [−12h19m; +12h22m]	mean difference ≅ 0 90% CI [−8h34m; +8h36m]	mean difference ≅ 0 90% CI [−12h29m; +12h30m]
$N_1 + N_2 = 1{,}000$	mean difference ≅ 0 90% CI [−5h23m; +5h25m]	mean difference ≅ 0 90% CI [−3h18m; +3h17m]	mean difference ≅ 0 90% CI [−3h33m; +3h32m]

While the results for Company A and Company B were fairly similar in MTTR for these percentile measures, you can see the impact of the differences between their incident durations.

Geometric mean

Another aggregate statistic you might be interested in is the geometric mean, which is calculated as $\sqrt[n]{x_1 \cdot x_2 \cdot x_3 \ldots x_n}$. This is especially appealing given the fact that the incident duration distribution isn't too far off from lognormal distribution, and so the geometric mean is to lognormal distribution what arithmetic mean is to normal distribution. As before, this can be simulated quickly (Table 5).

Table 5. 90% confidence intervals for difference of two geometric means calculated from two randomly sampled sets of incidents ($N_1 = N_2$) across 100,000 simulations.

	Company A	Company B	Company C
Geometric mean TTR of original data	54m	1h 9m	2h 24m
$N_1 + N_2 = 100$	mean difference ≅ 0 90% CI [−24m; +25m]	mean difference ≅ 0 90% CI [−27m; +27m]	mean difference ≅ 0 90% CI [−56m; +56m]
$N_1 + N_2 = 1{,}000$	mean difference ≅ 0 90% CI [−7.2m; +7.2m]	mean difference ≅ 0 90% CI [−8.5m; +8.7m]	mean difference ≅ 0 90% CI [−18m; +17m]

As yet, we are not getting good enough results at a practical number of incidents. With one thousand incidents, the 90% confidence interval just about makes it past a 10% change in the metric.

Sum incident duration

You might be interested in reducing the total incident duration, instead of the typical incident duration. The argument is intuitive: you want to offer a reliable service, but the reliability of the service isn't defined as much by the mean incident duration as it is by the total unavailability.[12]

We've already done this analysis once! The arithmetic mean is the sum of incident durations divided by the incident count, and so you can simply multiply the results of the MTTR simulation by $N/2$ (that

12 Depending on your business, this reasoning might be flawed. Consider that having a single one-hour-long incident a month might impact your users (and your business) very differently than 60 one-minute-long incidents. This same concern also applies to the commonly used service level objective language.

is, the number of elements in either of the two samples) and get the results of a simulation with the sum. To trivially confirm this, I generated a handful of simulations with the sum, showing that the confidence intervals are equal to MTTR confidence intervals multiplied by the corresponding N (Table 6).

Table 6. 90% confidence intervals for the difference of two incident duration sums calculated from two randomly sampled sets of incidents ($N_1 = N_2$) across 100,000 simulations.

	Company A	Company B	Company C
$N_1 + N_2 = 100$	mean difference ≅ 0 90% CI [−87h; +87h]	mean difference ≅ 0 90% CI [−82h; +82h]	mean difference ≅ 0 90% CI [−113h; +113h]
$N_1 + N_2 = 1{,}000$	mean difference ≅ 0 90% CI [−275h; +274h]	mean difference ≅ 0 90% CI [−260h; +259h]	mean difference ≅ 0 90% CI [−359h; +357h]

The number of incidents has a huge impact on the observed value of the sum. Let's briefly look at the incident counts next.

Counting incidents

This report discusses whether you can detect improvement in handling of incidents, focusing on analyzing how an incident is resolved. Going from having an incident to not having an incident at all is outside the scope of this paper.

However, since I've gathered all this data, at the very least I can take a brief look at the data sets to understand the behavior of the incident counts over time. I will not attempt a deeper analysis here.

The incident count is just as erratic as incident durations. Even aggregated to whole years, as shown in Figure 8, the values jump around wildly. At the resolution of months or quarters, it is even worse. At best, some egregious trends can perhaps be gleaned from this graph: Company C has seen a steep increase in its incident count in 2019 (the trend continues into 2020, not shown in graph) compared to years prior. This trend is only apparent at a multiyear time scale, which is especially visible when compared to the erratic trends of Company A and Company B.

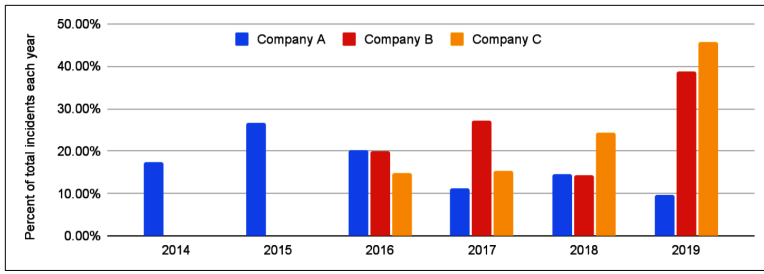


Figure 8. Incidents per year, as proportion of total incident count, per company. Incomplete years (year 2020 when the data was collected and the first year of each data set) were excluded.

But this trend might not be reflective of systemic reliability at all. Could it be because of a change in the usage patterns due to external world events? Or was there a product portfolio change? Or could it be due to a change in incident reporting with the same production events, for example, changing regulatory requirements? I can only speculate, but such factors—which are often unavoidable—may impact and even invalidate your own analysis, in your own company.

Other arguments against counting incidents have also been presented in the past.[13] I will not spend more time trying to analyze this data further, but I am looking forward to any future work focused more on this topic. Now that we have taken a quick look at the incident counts, let's use this knowledge as we go back to our topic of analyzing the shortening of incidents.

Analytical Approach

So far, I've been using Monte Carlo simulations. However, you can also take an analytical approach. Could you rely on the central limit theorem and calculate the confidence intervals rather than simulate them? Well, sometimes.

The central limit theorem says that the distribution of the sample mean will tend toward normal distribution in the limit. However, with incidents being an infrequent occurrence, there might be so few of them that the central limit theorem does not even apply yet.

13 Rick Branson, "Stop Counting Production Incidents", *Medium*, January 31, 2020.

Quite possibly, your team or company might not have enough incidents to have the distribution of sample mean turn normal.

One way to test for this is to run a simulation to generate a normal probability plot (Q-Q plot) of the distribution of the sample mean.[14] In Figure 9, I have done just that for data from Company A. With a higher sample size (such as a year's worth of incidents), the plot tends toward a normal distribution. But for as few as three months' worth of incidents, it is quite skewed away from a normal distribution. It can be misleading to assume that the durations are normally distributed and impact the subsequent calculations.

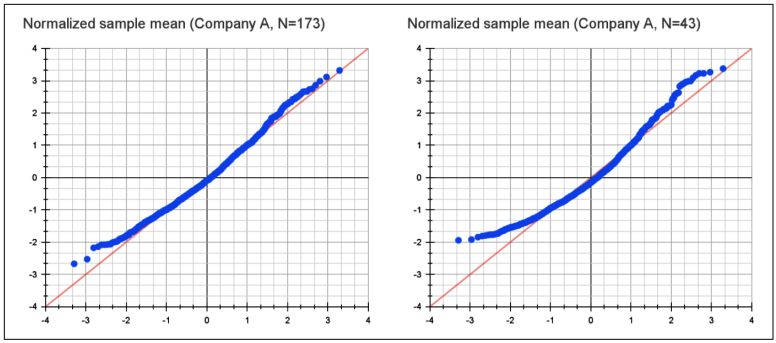


Figure 9. Normal probability plot for sample mean incident durations for Company A, generated from 1,000 simulations, for as many incidents as there were in year 2019 and in roughly one quarter of the year.

Once confident that the sample mean distribution is normal, you can use standard tools such as z-test or t-test to establish the confidence interval.[15] We are specifically interested in the difference between the two distributions, and since they are drawn from the same population, the mean difference (and therefore the mode of the normal distribution of the sample population difference) will tend to zero as we've seen it do in our simulations. The more interesting value is the standard deviation, which dictates the confidence intervals.

14 "Normal probability plot", Wikipedia.

15 See the chapters "Sampling Distribution of the Mean," "Sampling Distribution of Difference Between Means," "Testing of Means," and others in *Online Statistics Education: A Multimedia Course of Study*, project leader David M. Lane, Rice University.

The variance of the sample mean converges to:[16]

$$\sigma^2_{sample\ \ mean} = \frac{\sigma^2_{incidents}}{N}$$

And the variance of the difference of two normal distributions is:[17]

$$\sigma^2_{A-B} = \sigma^2_A + \sigma^2_B$$

For this case, where variance and sample size are the same for both sample mean normal distributions, this gives:

$$\sigma^2_{A-B} = \frac{2}{N}\sigma^2_{incidents}$$

This also explains the previous assertion that a 50/50 split is the best choice, since a different ratio of sample sizes would lead to a greater variance and therefore worse results.

You can then apply a two-tailed z-test. You can expand the z-test formula; knowing that the distribution mean is 0, you are looking for a given change in MTTR and also expanding it with the variance calculation:

$$z = \frac{\Delta MTTR}{\sqrt{\frac{2}{N}\sigma^2}}$$

You can also turn it around: you can look up the corresponding z-score (the z-score for a two-tailed test at our $\alpha = .10$ is ~1.644) and find the confidence interval of the MTTR change:

$$\pm\Delta MTTR = \pm z\sqrt{\frac{2}{N}\sigma^2}$$

16 See the chapters in *Online Statistics Education*, as well as "Distribution of the sample mean", Wikipedia.

17 Eric W. Weisstein, "Normal Difference Distribution", from *MathWorld*—A Wolfram Web Resource, updated March 5, 2021.

For Company A, the standard deviation of the incident durations is 5h 16m, and a sample of $N_1 = N_2 = 100/2 = 50$ is used to calculate the 90% confidence interval:

$$\pm \Delta MTTR = \pm 1.644 \sqrt{\frac{2}{50}(5h16m)^2} = \pm 1h44m$$

This result corresponds to the 90% confidence interval seen in the simulation results.

Although you can sometimes use equations to do the incident statistic analysis, I favor the simulation approach. I have found that it's easier to discuss the topic with a simulation that can be easily followed than with an equation. It also offers a lot more flexibility in what is modeled and analyzed. An analytical solution to calculating 95th percentile time to recovery might be quite challenging, but in simulation, it was a one-line change.

You may also be interested in modeling different changes and situations. What if the proposed incident shortening is more complicated than just 10% reduction? Maybe you expect different reductions depending on the incident class? And what if the SRE team consisted of werewolves, and they only started working on an incident after the full moon is over? Your scenarios might not be quite so fantastic, but simulation makes them easier.

Large Company Incident Data Set

The previous analysis has highlighted one thing: the variance goes down as the number of samples goes up. Google is a company with about one hundred times as many employees as the three anonymized companies, and it has significantly more incidents than those companies as well. Does that help get a confident incident metric?

We'll analyze the Google incident data the same way as we have the other companies' incident data. We can take advantage of having a richer data set (thanks to internal metadata) and break down the data a bit further.

Figure 10 shows the distribution of incident durations for all significant incidents and for the most severe incidents. Both data sets also include internal incidents, such as the ones that affect only Google employees and their productivity, or even events that are completely invisible to any user, internal or otherwise. The data set of most severe incidents has a higher proportion of user-facing ones (which would, for example, be listed on service status dashboards).

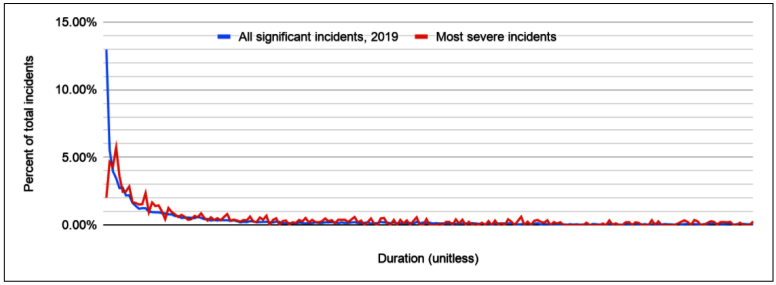


Figure 10. Distribution of incident durations for all Google incidents in 2019.

Except for an elevation of very short incidents in the broader incident set, the graph shows that the two distributions look roughly alike. The set of all Google incidents is approximately 15 times larger than that of the selected user-facing Google services, which is also why the company-wide distribution graph appears smoother.

In the case of the three public data sets, excluding incidents longer than three days removed ~1% of incidents, but both Google data sets had quite a few incidents lasting more than three days. As before with the public data set, it would be wrong to draw conclusions about reliability due to different incident tracking. I have tried both: a cutoff at three days and excluding the top 5% of incidents by length. The resulting confidence intervals of the relative MTTR differed only slightly, and the conclusions were the same. Table 7 has data for simulation with a three-day cutoff, consistent with the other simulations.

Table 7. 90% confidence intervals for difference of two mean TTRs and median TTRs calculated from two randomly sampled sets of incidents ($N_1 = N_2$) from Google incident data sets across 100,000 simulations; the number of incidents corresponds to a fraction of a year's worth in each data set.

		Google 2019 incidents	
		Most severe incidents (often, not always, user facing)	All significant incidents (often not user facing)
Incidents in 2019 (approximate relative size)		1 * X	15 * X
Mean TTR	$N_1 + N_2 = ¼$ year	mean difference ≅ 0 90% CI [−35%; +35% of MTTR]	mean difference ≅ 0 90% CI [−11%; +11% of MTTR]
	$N_1 + N_2 = ½$ year	mean difference ≅ 0 90% CI [−25%; +25% of MTTR]	mean difference ≅ 0 90% CI [−7.6%; +7.6% of MTTR]
	$N_1 + N_2 = 1$ year	mean difference ≅ 0 90% CI [−18%; +18% of MTTR]	mean difference ≅ 0 90% CI [−5.3%; +5.4% of MTTR]
Median TTR	$N_1 + N_2 = ¼$ year	mean difference ≅ 0 90% CI [−53%; +52% of median TTR]	mean difference ≅ 0 90% CI [−20%; +20% of median TTR]
	$N_1 + N_2 = ½$ year	mean difference ≅ 0 90% CI [−35%; +35% of median TTR]	mean difference ≅ 0 90% CI [−14%; +14% of median TTR]
	$N_1 + N_2 = 1$ year	mean difference ≅ 0 90% CI [−25%; +25% of median TTR]	mean difference ≅ 0 90% CI [−10%; +10% of median TTR]

Mathematically, the number of incidents in one year's worth of data of all significant incidents (I can't share the numbers, but it is more than the 1,000 from our previous tests) helps get more confident results, in line with what was found previously. However, you need to be mindful of the data you're looking at and the tests you are applying. It turns out that while mathematically true, this finding is not particularly useful in practice.

The data set of all incidents includes a wide variety of incidents, ranging from user-facing serving system failures to long-standing processing pipeline problems, network configuration, and corporate device software installations—often invisible to end users. For some

incidents, the time to resolution can be quite high as well (e.g., the incident is inherently long or can wait until after the weekend), pushing up the MTTR value.

I have no practical development that would promise this level of incident duration reduction over such a wide gamut of incidents. The ability to confidently detect changes as "small" as 5.3% in the mean after a year's worth of incidents is not strengthening MTTR's position as a practically useful incident statistic.

Is It About Data Quality?

The challenge in aggregate incident analysis does not appear to be about incident metadata quality. The efforts to improve the accuracy of metadata collection are unlikely to cause any dramatic changes. While inspecting the Google-internal incident metadata, I found no major improvement in the incident duration analysis for teams with more stringent incident-reporting expectations (e.g., teams with direct SRE support or running highest-availability, revenue-critical services). All three public incident data sets also show roughly similar behavior.

You can also verify this question by generating a completely synthetic distribution of incidents. If you make an assumption that the incidents are following a certain distribution (e.g., gamma or log-normal), you can choose the parameters such that it "looks right" in your subjective judgment and evaluate it.

This method can be applied to any distribution. However, that should be done with caution. It is likely not realistic to assume that the incident durations are, for example, normally or uniformly distributed. Drawing conclusions from the analysis of such distributions would be misleading.

And That's Why MTTx Will Probably Mislead You

Distributions such as the collected incident data (and perhaps the incident data of your company, too) have such a high variance that neither mean nor median nor a sum is going to be a good aggregate statistic to understand the trends in your incidents. Their variance is inherent to the incident problem domain, and so is the small sample

size. Having enough incidents that would allow for a robust analysis of incident durations, as in the three sample data sets, is undesirable. The analysis here was performed in ideal conditions, and the real-world performance is likely worse.

There is a difference between mitigation and recovery for reliability purposes, but in the scope of this analysis, it does not matter.[18] I call it "MTTx" because the actual measurement does not matter to the analysis, as long as it follows similar distribution properties and sample size (i.e., the incident count). Many other incident metrics, such as the time to detection, suffer from the same problem.[19]

This means that MTTx is a bad fit for typical practical analysis to evaluate the impact of a typical change on TTx:

- It is a poor measure of the overall reliability of your system. Reaching this conclusion alone does not require this analysis, and I can summarize one of the arguments made in *Implementing Service Level Objectives*: if you doubled the incident count while the incidents follow roughly the same distribution, your system's reliability has clearly worsened, but your metric has not changed a lot.
- It does not provide any useful insights into the trends in your incident-response practices. The simulations showed the amount of change you can see even if nothing changed about the nature of your incidents.
- Improvements in incident management processes or tooling changes cannot have their success or failure evaluated on MTTx. The variance makes it difficult to distinguish any such improvement, and the metric might worsen despite the promised improvement materializing.

These outcomes apply to typical reliability engineering situations, such as the incidents on web services. The default position should be to reject MTTx metrics for purposes like the ones above. However, there are exceptions. One exception would be if you have quantities that enable aggregate MTTx analysis. A real example is a large-scale

18 Jennifer Mace, "Generic Mitigations: A Philosophy of Duct-Tape Outage Resolutions", O'Reilly, December 15, 2020.

19 Alex Hidalgo, *Implementing Service Level Objectives* (O'Reilly, 2020).

hardware purchase, such as hard disk drives. The company Backblaze regularly publishes statistics about hard disk drive reliability on a per-model basis, reaching tens of thousands of devices per model.[20] Additionally, there are greater similarities between the hard drives of the same model than between the incidents. Similarly, the quantity and the lower variance are the reasons why you are able to confidently see changes in the mean latency of a typical serving system.[21]

Another exception would be truly dramatic changes, such as cutting the incident duration to just 20% of what it used to be. As shown, you will likely be able to confidently detect it in your data. However, you will probably be able to detect it in a lot of other ways, too, and might not want to employ the otherwise still problematic MTTx metric.

Better Analysis Options

The challenge with MTTx can be summarized as choosing the wrong metric to look at. The behavior of the metric is such that it defies analysis attempts.

However, another challenge with the metric is that it fundamentally might not be measuring the thing you are actually interested in. When we talk about MTTR improvements, we often mean to ask, "Have we gotten more reliable?" or perhaps, "Have we gotten better at responding to incidents?" Choosing a metric that more accurately represents your decision goals is an important topic covered in other literature as well.[22]

I did not find any "silver bullet," a metric that promises to be generally as applicable as MTTx is often claimed to be. However, we can look at some ways to choose a better metric for specific contexts.

20 "Hard Drive Data and Stats", Backblaze.

21 Although other statistics, such as higher percentiles, are often a better measure for latency of serving systems.

22 Douglas W. Hubbard, *How to Measure Anything*, 3rd ed. (Hoboken, NJ: John Wiley & Sons, 2014).

Tailor Your Metric to the Question

I used simulation to test whether a product impacts MTTx. However, that's not what any product or process change in *reality* does. Instead, it improves some aspects of an incident. Perhaps it is the incident communication process or perhaps it is the automatic incident analysis tooling that suggests the hypothesis.[23]

As noted earlier, an incident is a collection of steps of varying durations. These steps have been studied in various publications.[24] If you are improving one step of the journey, including all other steps in the aggregate makes your ability to understand the impact of the change worse.

Trying to analyze the individual behavior of each and every incident is likely not practical. You cannot rely on humans entering metadata, and you are unlikely to be able to tightly observe each incident. Instead, a practical solution can be user studies on a select sample of incidents. These studies can be constructed to focus on just the aspects of the incident you are interested in and can surface richer understanding than an aggregate statistic can ever hope to. Constructing these studies correctly is not always trivial, and expert advice is recommended if possible. With that in mind, some literature is helpful in establishing a low-effort user test, and I have successfully applied the lessons learned in building practical systems.[25]

Consider Direct Reliability Indicators

Perhaps your question is, "Is our reliability getting better or worse, as a company?" This is where the concept of *availability* comes in. In SRE practice, the familiar language for this would be service level indicators (SLIs) and service level objectives (SLOs). Ideally, these should represent the measure of user-perceived reliability of your product, and the SLOs should be set as the objectives that are the right business trade-off. Often, neither is exactly true, and sometimes they are far away from this ideal.

---

23 Andrew Stribblehill, "Managing Incidents," in *Site Reliability Engineering* (O'Reilly, 2016).

24 Allspaw, "Moving Past Shallow Incident Data"; Charisma Chan and Beth Cooper, "Debugging Incidents in Google's Distributed Systems: How Experts Debug Production Issues in Complex Distributed Systems", *Queue* 18, no. 2 (March–April 2020).

25 Steve Krug, *Rocket Surgery Made Easy* (Berkeley, CA: New Riders, 2010).

If your SLIs and SLOs are as true to your business goals as possible, this still does not automatically mean they can be analyzed using aggregate statistics, such as sum error budget burned per year. Given the breadth of how an SLI (even one with close to ideal properties) can be implemented, any answer provided here would likely not be broadly relevant. I have not done any analysis in this space, but it would make for interesting future work. You may be able to do it within your company easily, with the tools previously discussed.

Depending on your business, another measure may be the total count of opened support cases, or customer phone calls as a result of service unreliability, or perhaps some more advanced composite metric.

Put Your Chosen Metrics to the Test

There might be better approaches than suggested here, and I'm looking forward to future work in this field. The salient point is that the analysis should focus on the thing you are actually interested in; you should choose your metrics wisely.

Reliability incidents are varied, and so are the questions that need to be answered about measuring reliability. The key thing is to look at your metrics with a critical eye. Are they really measuring what you meant to measure? Are they robust in the face of randomness? Do you have evidence to support your answers?

The same tools that I've used to investigate MTTx can be used for another metric you might be considering. The process is much the same: determine what level of change is meaningful to you (this depends on the metric, but also on your business), and then analyze whether you can confidently see it in the data.

Conclusion

I have demonstrated that even in a favorable analysis setup, MTTx cannot be used for many practical purposes where it has been advertised to be useful, such as evaluating reliability trends, evaluating results of policies or products, or understanding the overall system reliability. The operators of systems, DevOps or SREs, should move away from defaulting to the assumption that MTTx can be useful. Its application should be treated with skepticism, unless its applicability has been shown in a particular situation.

The problem is not specific to the metric being an arithmetic mean; I've demonstrated the same problem with median and other metrics. It is a consequence of the typically low volume of incidents and high variance of their durations. This distribution has been observed on practical data sets from three anonymous companies, as well as the obfuscated data set from Google.

Instead of analyzing the overall incident statistics using MTTx, you can focus on more narrow questions of the incident life cycle, more closely aligned with what you are trying to evaluate. That can lead to a different choice of metric or a different measurement process altogether. A better choice of metric should give better and more robust decision processes. An example of this might be measuring and studying the time to detection specifically, or time spent on some common incident-response activities.

Perhaps there are other statistics that can be used to glean more value. Perhaps the variance of the incident durations is itself useful, as it could attest to consistency in the ability to respond. Whatever the case may be, one thing is certain: you should think critically about your metrics and put them to the test (perhaps using some of the tools referenced in this report). Go beyond relying on assumptions or intuitions or industry trends, and seek evidence that the metrics you have chosen can be used to indicate what you wish them to.

Acknowledgments

The author would like to thank Kathy Meier-Hellstern for her review, advice, and suggestions; Ben Appleton for his review of this work as well as of some initial work leading up to this text; Michael Brundage for further review and for inspiring additional analysis; Scott Williams for further review; and Cassie Kozyrkov for her work to make statistical thinking an increasingly accessible topic.

About the Author

Štěpán Davidovič is a site reliability engineer at Google. He currently works on internal infrastructure for automatic monitoring. In previous Google SRE roles, he developed Canary Analysis Service and has worked on both a wide range of shared infrastructure projects and AdSense reliability. He obtained his bachelor's degree from Czech Technical University, Prague, in 2010.