



Observability:

A 21st Century Manifesto

Change Is the Only Constant

How many SREs does it take to define observability?

**All of them.
And they still won't agree.**

However you choose to define observability, you have to account for the one constant inherent in modern distributed software environments: **Change**.

Change in telemetry data types and sources. Change in application architectures and deployment methods. Change in engineering best practices and tools of choice. Change in numbers and types of distributed systems, containers, and levels of abstraction between application and infrastructure. Change in user consumption of your product.

Change in business that demands change in technologies.

At New Relic, we define observability very simply as how well you can understand your complex systems.

We believe that defining observability is much less important than the principles you apply to practice it successfully.

So instead of arguing about definitions, we spoke with developers and engineers about the topic and distilled observability down to these 10 principles.

10 Principles of Observability

01

Observability gives you **shared understanding**.

DevOps is all about eliminating silos between the engineers that build code and those who support that code in production, and observability gives those teams a common framework to take action on shared data.

“When you see a convergence of egos—monitoring people, observability people, and people holding the purse strings—observability offers you the shared understanding, the single source of truth, that can bring these egos together and work toward the same goals.”

Josh Biggley, TechOps Strategy Consultant, New Relic

02

Observability should let you instrument your system so that **all your data is always available** in real time from the same platform.

It's more critical than ever to see across your entire software system and get traceability through your full stack. That's why [94% of software leaders](#) believe that observability is key to developing software, instead of bolting it on as an afterthought.

That means being able to access all your metrics, events, logs, and traces—regardless of whether that data is from a vendor's agent, your homegrown solution, or an open standard—in one place.

“Every company, at its core, is a data company, and the most powerful tool for success is understanding that data.”

Zack Mutchler, TechOps Strategy Consultant, New Relic

03

Observability demands that you ingest **high-cardinality** data.

What's the point of collecting data if you can't analyze it on the fly? High cardinality gives you the flexibility to ask your data questions from all angles to address **unknown unknowns** and discover critical insights.

It's imperative that engineers are able to test their systems in production and ask questions to investigate issues they couldn't originally predict.

"[We can] make better decisions because [we have] the visibility we need to pinpoint issues and work proactively to resolve them. We're now able to fix problems we didn't even know we had."

Daniela Constanza Muñoz, Developer, [Banco de Credito e Inversiones](#)

04

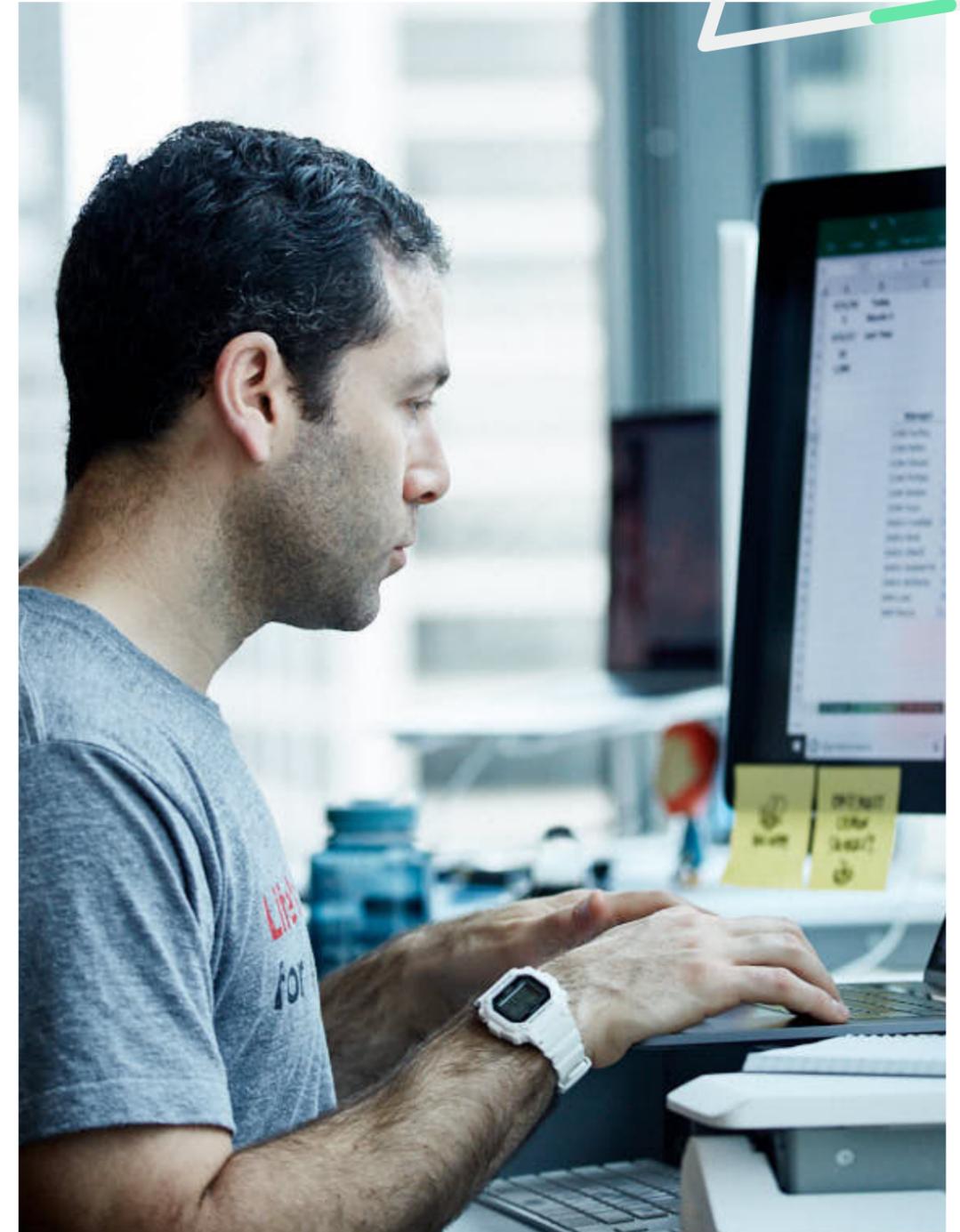
Observability gives your teams **confidence**.

The more quickly and effectively you understand your complex systems, the more confident you can be about embracing system change.

You have the confidence that everyone has access to the data they need to make informed decisions across the business. You have the confidence to scale as demand fluctuates. And you have the confidence to push new code, immediately see the effects, and react to any issues—before the end user realizes there was a problem.

"Observability means being able to make changes and not have to worry about breaking things. If you don't have confidence that you can make a change without affecting a bunch of different things, you're not going to be making those changes, and therefore you're not going to be innovating."

Leo Guinan, DevOps Engineer, Fuse by Cardinal Health



05

Observability enables **fireproofing**, not just firefighting.

The ability to connect and correlate signals gives you context into hotspots before they affect your customer experience. Understanding why critical incidents have occurred, and **proactively** preventing them from recurring, decreases downtime and improves MTTR, giving your team more time for innovation.

“We can see into the application performance, drill down into why the error rate spikes and when application latency shoots up, see which particular node is suffering. This is so valuable when a new build has been pushed out because it’s now possible to get a powerful view of where any instability is coming from.”

Chris Callaghan, Site Reliability Engineering Manager, [Royal Society of Chemistry](#)

Observability Outcomes: Software Leaders vs. Laggards

- 83% of software leaders experience fewer than 5 outages a month, compared with 3% of laggards.
- 75% of software leaders report an average MTTR of less than half an hour, compared with 1% of laggards.
- 78% of software leaders learn about service interruptions through observability solutions, versus just 12% for laggards.
- 89% of software leaders have adopted automated remediation, and only 5% of laggards have done so.
- Laggards learn of nearly half (48%) of their interruptions via customers, 15% more often than leaders (33%).
- Software leaders spend 77% of their time innovating, and only 23% fixing issues. Laggards spend double that—46% of their time—on troubleshooting.

Source: [Deeper Than Digital](#) - New Relic

06

Observability is a **socio-technical** system.

It goes beyond systems and software, and runs through the whole organization—different lines of business, teams, roles, and *people*.

It lets you see how all of your applications—as well as their underlying services and systems—relate, so you can understand dependencies across organizational boundaries, and troubleshoot and solve problems faster.

It's visibility into people and processes, not just technology.

“Observability encompasses the whole socio-technical system—the human side, the sensors—there’s no line of demarcation between the technical side and the people running it. If you try to separate them into neat little buckets—well, it’s impossible, and the effort to do it undermines your effort to understand what’s going on and identify where to improve.”

Beth Long, Sr. Software Engineer, New Relic

07

Observability lets you **move faster**.

It gives you context into relationships and lets you respond much more quickly to incidents.

“Managing our growing microservices environment requires observability ... a single pane of glass where we can understand what’s going on in the underlying infrastructure, our individual applications, and all our microservices. Not only can we make sure we’re meeting our service level objectives, but when things go wrong, we can resolve the issue faster. ... It lets us move fast without the wheels coming off.”

Matthew Tapper, Lead Site Reliability Engineer, [Culture Amp](#)

08

Observability is **an infinite loop**.

It's a journey, not a destination; **constant improvement**, not wholesale change. If you believe you've reached "perfect" observability, you are wrong. If you don't continue adapting and improving your observability practice to keep up with changes in business requirements and best practices, it will eventually be insufficient.

The good news is a successful observability practice will improve the inner workings of your entire business. The bad news is you're never done working to maintain it.

"How we deal with today's problems is going to open up a whole new set of problems, and keeping on top of that is the big challenge in the next few years."

Beth Long, Sr. Software Engineer, New Relic

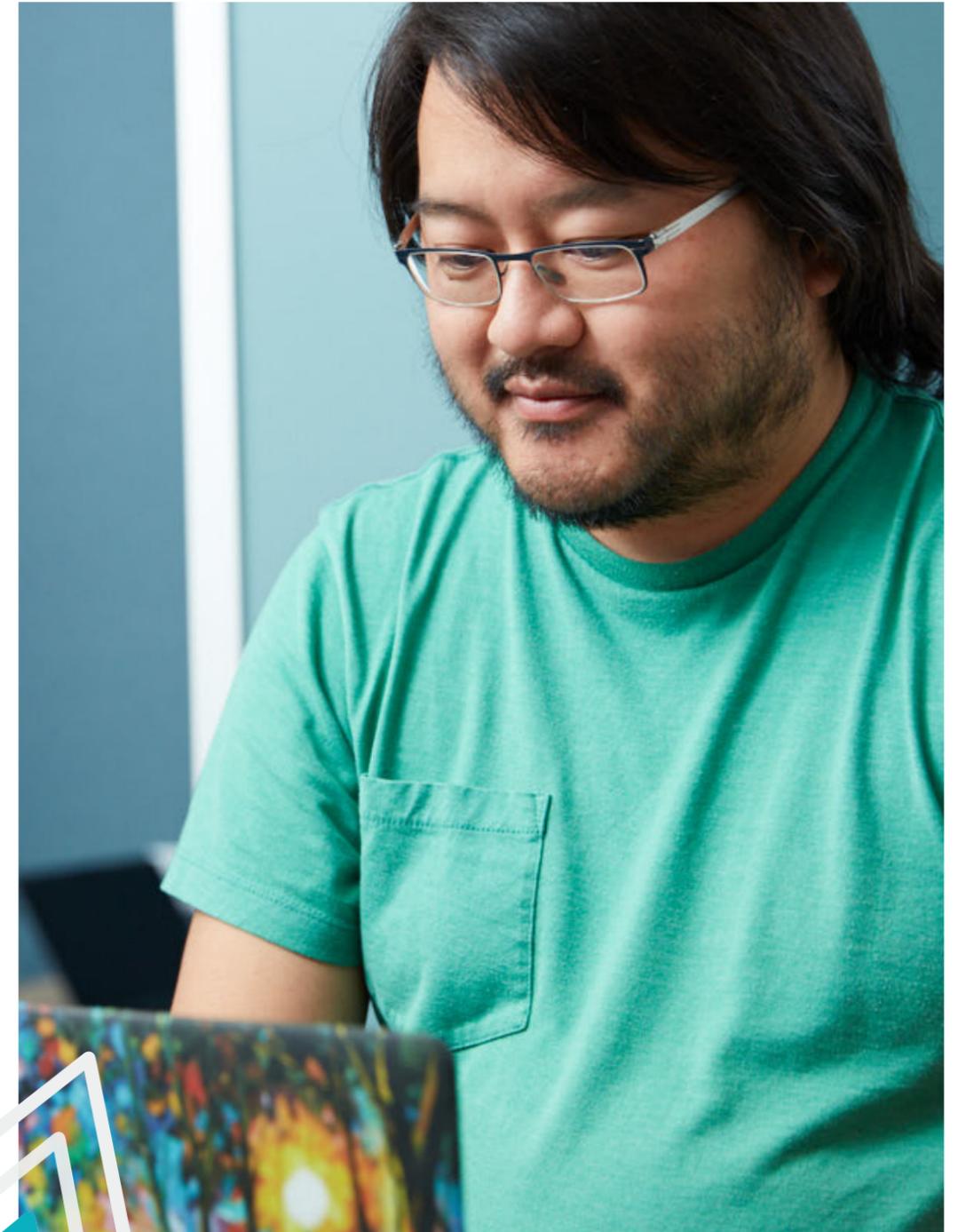
09

Observability is **leaving work at work** and **sleeping better** at night.

It's beating alert fatigue *and* physical fatigue— with the peace of mind that your system is **resilient** enough to weather issues without complete failure, and with alerts prioritized so you'll be interrupted only for something critical.

"Not having visibility over a system you're responsible for—you will lose sleep at night. And you will feel powerless because you don't have the information you need."

Beth Long, Sr. Software Engineer, New Relic





10

Observability is a **superpower**.

It lets you rethink how you're doing a job and understand how software changes—connections between your tools, applications, and infrastructure—affect the end user. It empowers you to improve that experience without jeopardizing system health.

“Observability allows me to be a data storyteller by having access to all of the data. It allows me to be a better leader, galvanize necessary support within the organization, and achieve goals.”

Josh Biggley, TechOps Strategy Consultant, New Relic

6 Questions You Should Be Asking

Observability is like democracy—you can always do it better.

Start where you can and move at a pace that works for your organization.

To move from monitoring (a passive approach that tells you when something is wrong) to observability (an active approach that lets you understand *why*), these are the questions you should be asking of your system, your leadership, and your SREs.

Q.1 Are you ingesting metrics/events/logs/traces from every source, whether open source, in-house, or proprietary?

Q.2 Does your platform give you the flexibility to collect high-cardinality data that allows you to ask and answer the most specific and ad hoc questions?

Q.3 Are you able to dynamically interrogate the data you collect—*especially* questions you didn't know you needed to ask when initially setting up instrumentation?

Q.4 Can you connect datasets to deliver custom insights that inform key business metrics?

Q.5 Can you surface dynamic connections automatically, to sense patterns in the data through analytics and curated visualization experiences?

Q.6 Do you have the willingness at every organizational level to invest in a platform that provides the visibility to uncover the reality of your situation?

Or to put it another way—does your leadership want data that confirms what they think, or do they want data to know the truth of your situation?

Change the Way You Change

Traditional businesses struggle to adapt to new needs and new challenges, because today's complexity makes it so much harder to separate the signal from the noise.

It's hard to see how people, teams, and systems all come together to operate.

But the stronger your observability practice, and the more it evolves as your system evolves, the better positioned you are to overcome that complexity.

New Relic. Observability made simple.

[Sign up for a free New Relic account.](#)



© Copyright 2021, New Relic, Inc. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. 02.2021