

Sensible Observability Checklist

Introduce your organization to the best observability practices



Part 1: Productivity

1	New product features turn into actual software at a satisfactory pace	You can measure new feature development using the lead time metric.	<input type="checkbox"/>
2	There are no major roadblocks in the coding phase	Such coding roadblocks (e.g.: technical debt; underestimated jobs) exist in a phase called cycle time.	<input type="checkbox"/>
3	Code review time is measured and satisfactory	Non-technical issues such as poor task prioritization can affect code review time.	<input type="checkbox"/>
4	Testing functionalities is predictably swift	QA engineers who test functionalities can experience communication problems with developers.	<input type="checkbox"/>
5	There are processes for finding and eliminating development bottlenecks	Without such processes, your gains will be accidental and your system won't be able to become self-healing.	<input type="checkbox"/>
6	Each piece of code deployed can be linked to specific business needs	It's a straightforward way of making sure that your development prioritizes the company's requirements.	<input type="checkbox"/>
7	Each escaped defect is tracked and analyzed	To lower your escaped defects ratio over time, know where you stand right now and work from there.	<input type="checkbox"/>
8	It's clear how much work each developer has	Improving developer capacity management makes software delivery more efficient.	<input type="checkbox"/>

9	It's also clear how much work each developer can actually do	Giving too little or too much work to your developers makes them inefficient and they miss deadlines. Measure development velocity to avoid this.	<input type="checkbox"/>
10	Developers come and go at a predictable rate	High developer turnover makes it hard to maintain quality development and project knowledge.	<input type="checkbox"/>

Part 2: Reliability

11	It's easy to pinpoint the source of unusually high cloud costs	Breaking down your infrastructure costs by different services or instances is key to cost optimization.	<input type="checkbox"/>
12	Each new code release is observed for unexpected changes	A new code release may cause obvious errors. While everything can be fine on the surface, there are hidden changes that may cause cost bleeding.	<input type="checkbox"/>
13	The system is prepared for sudden spikes of traffic	A truly scalable system can adapt to any traffic spike automatically even when one is completely unexpected.	<input type="checkbox"/>
14	The system is scanned for security vulnerabilities	The usual cyber attack is undetectable. It may take a while to uncover it. By that time, the damage will be done.	<input type="checkbox"/>
15	The system focuses on events that matter like critical errors	Many critical errors begin with improper signals from your system. By observing them, you can prevent them from ever occurring.	<input type="checkbox"/>
16	There is an incident management process in place	Merging all of these best practices into a comprehensive strategy is what incident management is all about.	<input type="checkbox"/>
17	It's possible to release new changes to a selected user group	By doing so, you can test new features discreetly and roll them back if problems arise.	<input type="checkbox"/>
18	There's a central place for all system logs	It's a progressive strategy that will make it easier for your DevOps to use them in the future.	<input type="checkbox"/>
19	It's easy to break down and analyze each user request	User requests provide a wealth of data that allows you to trace the user journey and their behavior.	<input type="checkbox"/>
20	There are no unused resources that generate needless costs	The answer can tell if your cost optimisation efforts work or not.	<input type="checkbox"/>

Part 3: Marketability

21	Logs, metrics, and traces are used for user research	Logs provide event info (e.g. errors), metrics quantify system factors (e.g. speed), while traces help detect root causes.	<input type="checkbox"/>
22	The system provides sales and marketing with key data	Such system data can even be used to automate the optimization of campaigns such as PPC ads.	<input type="checkbox"/>
23	It's easy to record the reaction of users to new product changes	You should be able to break down your user data by time, source, or segment to know how they respond to new releases.	<input type="checkbox"/>
24	The system allows to compare different variants of new content or features	Built-in tools should let teams run content experiments (e.g. A/B tests) for selected audiences.	<input type="checkbox"/>
25	Dashboards clearly show when the system is under the heaviest load	This is known as load monitoring, which can let you adjust resource usage to current needs in order to minimize costs.	<input type="checkbox"/>
26	Seasonal differences in user behaviour can be observed and analysed	Business seasonality can be noticed quite easily, but what's hard is to tie it to other metrics such as resource usage at the right time.	<input type="checkbox"/>
27	Revenue data can be broken down in multiple ways	By breaking down revenue data by user segments, app features, or products, you can uncover useful user patterns.	<input type="checkbox"/>
28	The system provides the data needed to optimize conversion rates	One way to test if you collect and analyze the right data is to ask yourself if you have already used it to improve conversion rates.	<input type="checkbox"/>
29	There is a way to measure NPS over time	The Net Promoter Score requires user feedback. But in return, it provides metrics that can be used strategically to increase revenue.	<input type="checkbox"/>
30	The system complies with all the latest GDPR regulations	When you comply with GDPR, you help your users and yourself by contributing to strong security practices.	<input type="checkbox"/>

**Review
your score**

Audit your observability ops or other cloud areas. Introduce the right change with a certified engineering team that delivers 30 cloud projects per year. [Learn more](#) →

Glossary

Agent	A program that performs a specialized task in the background. They are the workhouse of data-driven systems that are responsible for automated jobs such as collecting system-level metrics, moving data between tools and services, or authenticating API clients.
Canary deployment	A deployment scheme in which the traffic is split between the newly deployed version and the previous one. The split may concern only specific user segments and the new version may be withdrawn fast. A very useful tactic for content experiments and user research.
Cost bleeding	When your system lacks sensible observability, you might be unaware that some paid for resources are redundant or that some functions generate avoidable costs. Each of these costs can be small on its own, which makes them hard to spot. But they do add up.
Cycle time	This metric measures the time it takes your developers to turn user stories into actual working code. It's the biggest part of the lead time, and it's great for assessing your technical prowess.
Developer capacity	Defines the number of hours a single developer can work in a given iteration. Measuring developer capacity makes it possible to assign work in a way that ensures efficient development and contributes to the wellness of your employees.
Escaped defects ratio	Some errors and mistakes escape sneak past your QA engineers and developers. How many is too many? That depends on the nature of your project. In order to limit software issues, measure your escaped defects ratio over time.
Incident management process	Sensible Observability puts emphasis on processes. Solving a single problem without a process only makes a small difference. Using incident management, you can design processes for working with various types of data and eventually create a self-healing system.
Lead time	If there is something wrong with your development that makes your time-to-market lengthy and your bills higher than they should be, you can find the cause by studying lead time. Lead time can be used throughout the product development process.
Load monitoring	Load monitoring refers to the ability to calculate server resources needed to handle the job at hand. By applying the load metrics, you can even scale your resource usage automatically in response to traffic. That way, you can maintain product availability in the cloud at the lowest price point.
Logs	These are text files that contain data about system performance and users, including server errors or access records. The key is to find the logs that actually contain crucial information to extract insights from a flood of generic messages.
Metrics	Systems also generate metrics. While logs report events, metrics quantify system performance. They include information on network traffic, user signups, or CPU usage. Metrics play a part in all observability and monitoring-related activities.
Net Promoter Score	A market research metric that quantifies customer loyalty. Through surveys, you learn how likely your clients are to recommend your business. Some argue that NPS is a predictor of company growth. If you want to measure it, it's best to make it a process and compare progress over time.

Request processing time

The speed at which your system processes requests is crucial for good UX. What you might not know is that different parts of your system can respond at varying speeds. Without sensible observability, it may be difficult to pinpoint isolated sources of subpar performance, especially with distributed systems.

Sensible Observability

Sensible Observability refers to a system that keeps track of its internal signals, making it easier to diagnose or even predict problems based on them and apply measures to remove errors or optimize performance.

Traces

When your user enters a page, or starts a checkout process, they send a request. Traces track them as they travel through the system. This data helps uncover technical issues along their way. Traces are especially useful in the case of distributed systems in which requests travel across services.

Well-Architected Framework

The AWS Well-Architected Framework is a set of best practices designed to produce healthy infrastructures. One of its 6 pillars (beside security, reliability, performance, cost optimization, and sustainability) is operation excellence that's based on monitoring AWS services to improve and repair processes continuously.

Velocity

While capacity refers to the amount of time your team has, velocity has to do more with their efficiency. It's the amount of work (story points) your team can deliver in a single iteration. By measuring velocity, you can approximate the speed at which business value can be delivered in the next sprints.

Zero downtime

It's one of the biggest reasons you need observability! The term applies when your website or application is never down or even unstable regardless how much traffic it handles or the changes you make. Achieving zero downtime requires both strong foundations as well as the use of proper deployment tactics, such as the aforementioned canary deployment.

“We’ve seen a **double-digit increase** in NPS.”

This comes from Pet Media Group, which has been working with our cloud engineers for 3+ years. Learn how to use your cloud better to drive business change. Or ask how to improve your observability.

[Visit tsh.io](https://tsh.io)

