The Road to Clarity: Observability in Salesforce

pharos.ai/blog/1-road-to-clarity

Every Salesforce Org has a Blind Spot, Including Yours

Picture this: you're cruising down a desert highway on a 500-mile road trip in your car. But there's just one hitch...you have no dashboard.

For most of us, that would make for a long and anxious drive. We take for granted the critical information our dashboards provide: how fast we are going (speedometer), how hot our engine is (temperature gauge), and how hard it is working (RPMs).

All those gauges and lights, activated by strategically located sensors, give us a quick view into our car's current state, Some provide warnings of a perilous *future* state. Fuel gauge nearing E? Time to gas up, or there's a long hike in our future.

All that data, which is presented so efficiently that we can gather it in a glimpse, helps get us to our destination quickly and safely. Without it, the journey is much dicier. Eventually, we'll find ourselves stranded on the side of the road, going nowhere, calling for help.

Now let's talk about that dashboard in Salesforce

Have you found it yet? That beautiful information hub that lets you know when a flow has broken? You know, the one that shows you *exactly* which line of code is failing so you don't have to hunt for it manually?

Are you appreciating that holistic view that reveals when exception errors are spiking and who is impacted? How about the reports, especially the one that rolls up the health of your org for your weekly meeting with the boss?

If only. The sad truth is, there is no dashboard built into Salesforce.

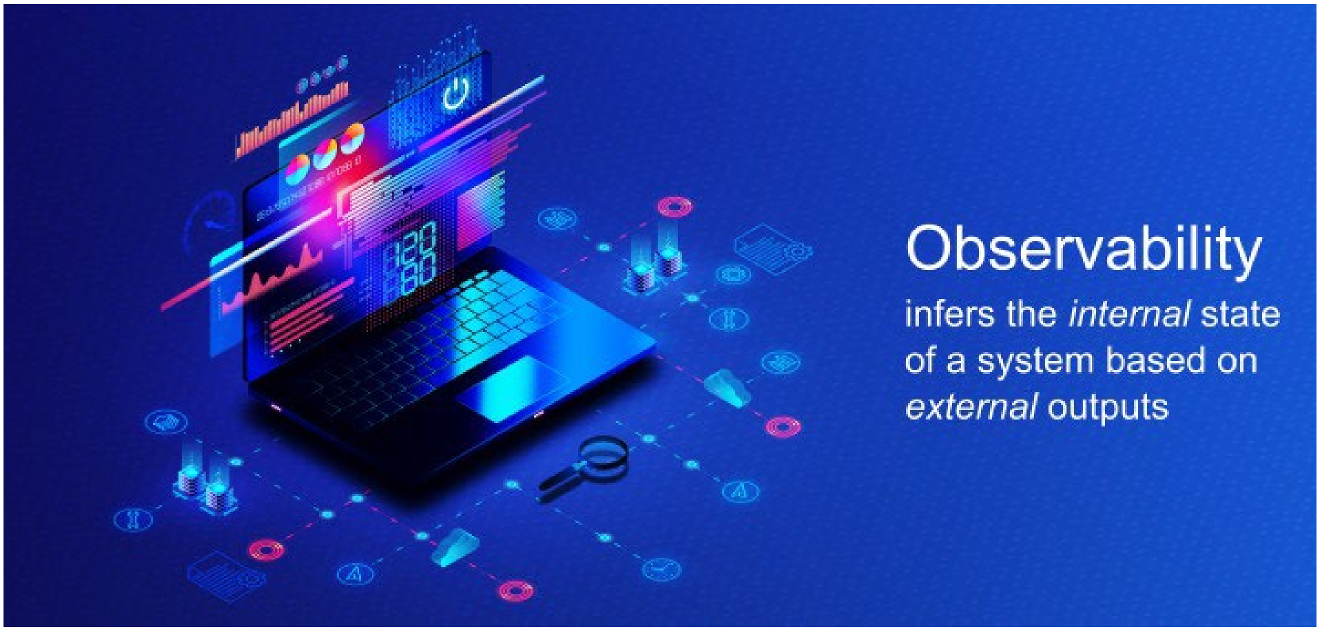
For many orgs, users are often the first to sound the alarms when things go awry in production. Even for those companies that have the luxury of a robust QA process in place, strange things happen in the real world.

Staying in Front of Trouble

What Salesforce is missing is full-throated *observability*.

Once an obscure concept that existed in the realm of engineers and mathematicians, observability has gained traction in recent years. Companies like Splunk, Dynatrace, and IBM are applying its principles to complex systems, where it has become a mission-critical best practice outside of Salesforce.

Simply put, observability means being able to infer the *internal* state of a system based on its *external* outputs. For example, if your car's temperature gauge is going up (external output), you can infer there is something wrong with the water pump or radiator (internal state). With this information, you know that you'll need to take your car to a mechanic, and your mechanic will know which parts need attention.


Observability
infers the *internal* state of a system based on *external* outputs

In software systems, observability involves collecting, analyzing, and visualizing data from various system components. The output takes the form of *logs*, *metrics*, and *traces*, which give vital insights into system performance and user interactions. Staff leverage these insights to rapidly identify and resolve issues, optimize resources, and improve a system's reliability and user experience.

Good News...Bad News...

If you’re responsible for some aspect of your Salesforce org, the good news is that Salesforce does have logging and tracing built in, so observability within your ecosystem is possible.

The bad news? As every admin knows, the way Salesforce reports errors can be challenging to manage.

On the front end, users encounter those infamous red boxes, loaded with vague and unhelpful messages that lead to frustration and bring work to a halt. On the back end, Salesforce sends lengthy and cryptic emails. Sometimes they are sent to admins. Sometimes they are sent to the wrong contact or into a black hole.

The volume of those emails can be prolific. We know of companies that are receiving more than a thousand exception error emails each day. Those emails are largely ignored.

This presents a big problem for any business, a problem that extends beyond IT.

One goal of observability is to provide an early warning system so admins can address errors before they become problems (or revenue-threatening catastrophes). Consider this real-life example:

Customers were filling out and submitting order forms for large, complex purchases. Or so they thought. In reality, the orders lingered in an unprocessed state for days. No one noticed. It wasn't until customers began complaining that the issue escalated, but by then it was too late. Many customers had given up and taken their business elsewhere. The result was a significant amount of revenue lost unnecessarily because a back-end error wasn’t quickly and properly identified.

These are the kinds of issues observability is designed to prevent.

Ending the Troubleshooting Time-Sink

Making the most of limited staff time is another major benefit of observability. It’s not uncommon for troubleshooting to take hours if not days. Typically, the process runs like this: a user reports an issue, maybe submits a ticket. The Salesforce admin reaches out to get more information, asking how and when it happened. Support staff might then meet with the user to try and recreate the error. It’s like a murder mystery, except not nearly as entertaining and not always solved.

With good observability processes in place, errors can be identified instantly, even before a user raises the issue. What’s more, the cause can be pinpointed down to the line of code that is triggering it. Issues that once took days to research can be resolved in minutes.

This is why anyone involved with the development and administration of Salesforce should care about observability – and push for it. The lack of holistic and actionable observability is the platform's Achille's heel. As Salesforce becomes increasingly mission critical and more complex, the need becomes greater.

That's why observability should be on every Salesforce org’s roadmap, whether home-grown or achieved through the native capability Pharos provides. More than just log analysis or application performance monitoring, observability will improve your processes, detect the “unknown unknowns” that threaten revenue, radically shorten troubleshooting time, and provide for an environment that operates much more efficiently.

Like the dashboard on your car, observability will make your Salesforce journey easier and safer – and keep your org off of the shoulder of the road.

__Pharos__ is the leading expert for observability within the Salesforce ecosystem. If you’re interested in learning more about observability and how to bring it to your Salesforce org, let us know. We’ll be happy to introduce you to additional resources.

Meet with a Pharos Expert