

## Chapter 5. Conclusion

As my friend Brian Knox, who manages the Observability team at DigitalOcean, [said](#),

*The goal of an Observability team is not to collect logs, metrics, or traces. It is to build a culture of engineering based on facts and feedback, and then spread that culture within the broader organization.*

The same can be said about observability itself, in that it's not about logs, metrics, or traces, but about being data driven during debugging and using the feedback to iterate on and improve the product.

The *value* of the observability of a system primarily stems from the business and organizational value derived from it. Being able to debug and diagnose production issues quickly not only makes for a great end-user experience, but also paves the way toward the humane and sustainable operability of a service, including the on-call experience. A sustainable on-call is possible only if the engineers building the system place primacy on designing reliability into a system. Reliability isn't birthed in an on-call shift.

For many, if not most, businesses, having a good alerting strategy and time-series based "monitoring" is probably all that's required to be able to deliver on these goals. For others, being able to debug needle-in-a-haystack types of problems might be what's needed to generate the most value.

Observability, as such, isn't an absolute. Pick your own observability target based on the requirements of your services.