Catch Data Errors Before They Happen

ObservePoint

Would you make business decisions based on the advice of a psychic?

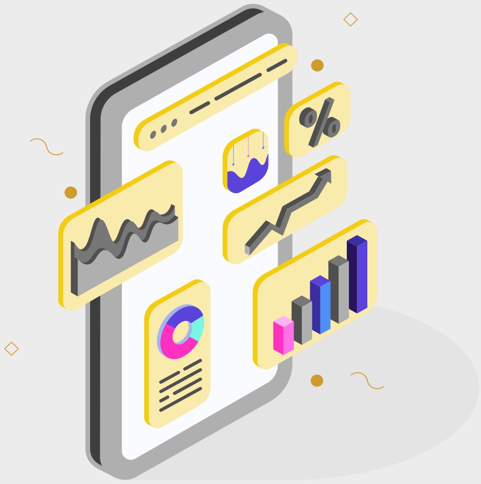
Hopefully not.

A reasonable professional would recognize that basing business decisions on Madam Perdita's predictions won't get you far.

But here's the clincher: You, the reputable, data-driven professional, are as vulnerable to misguided advice as a psychic's patron if you don't have mechanisms and processes in place to validate the accuracy of the data guiding your business decisions.

Whether you're an analyst, marketer, decision-maker, or all of the above, the accuracy of your implementation, and the data collected from that implementation, has a significant influence on your ability to make effective decisions—and therefore your credibility.

Don't lose that credibility, and don't miss out on business opportunities because of faulty data. Here's how you can catch analytics errors before they happen and become your company's data-driven champion.



The Reactive Approach

For Catching Data Errors

Let me tell you a story that might sound familiar.

Following what appeared to be a successful website release, a marketing team met to review their results. The analyst pulled up the reports to show his boss what kind of performance increase they were seeing but, instead, saw that their top metrics had dropped noticeably. How could this be?

The analyst opens a key metric report only to find it's completely empty. For some reason, the implementation hasn't been collecting this data point since the launch. With considerable effort, the analyst manually combs through the analytics implementation to find the problem: conflicting JavaScript in the new release compromised the analytics implementation.

The team has lost critical data for accurate analysis of the new site's performance, and the analyst lost credibility with his supervisor. On top of that, if he hadn't noticed the error (not all data errors are obvious), the decision-maker could have made a bad call on bad information.

Many companies are still testing their analytics implementations after a release (or not at all) and unsurprisingly face similar scenarios—discovering they've been collecting inaccurate data that is unusable. Then they lose even more time, data, and confidence trying to figure out what happened. The reactive approach is much less than ideal—it's crippling and costly.

The Proactive Approach

For Catching Data Errors

The best way to catch analytics errors before they skew your data is to proactively test your implementations before, during, and after release.

While this can be done manually (by combing through lines of code to investigate individual tags one page at a time), this method requires a lot of man hours and might not even be completed in time for the next release, making manual testing both time-intensive and expensive.

The optimal way to perform and maintain this scale of testing is with an automated data governance solution that uses Audits, Journeys, and Rules to:

- Ensure your implementations are deployed and functioning properly
- Test against predefined rules to determine whether or not the data is correct
- Alert you to tagging errors—including missing, duplicate, or piggybacking tags—that corrupt, inflate, or compromise your data

Data Governance Tools:

- **Audits** automatically scan large portions of your site to identify tagging errors. After an Audit runs, errors are aggregated into clear reports so you can quickly identify and correct where tags are missing or misfiring.
- **Journeys** scan small sequences of steps, such as a booking path or a shopping cart process. This allows you to continuously test implementations on your most critical user paths and ensure they're always functioning.
- **Rules** are user-defined logic that tell your Audits and Journeys what to test against. If you expect to see specific tags (and specific values in those tags) on certain pages or after different events, you can set up a rule to verify that data was collected correctly in those instances. When a rule fails, you will be notified of what went wrong.

By using a data governance solution to test throughout the development cycle and beyond, you can avoid the reactive process described earlier and proactively uncover tagging errors before they wreak havoc on your data collection.

1. Implement a testing process for before and after releases.

When you are gearing up to launch a new release, there are a lot of hands touching and making changes to your site. Often, these changes cause tags to be relocated or even disappear from your pages, making testing a critical step in preventing data from being lost or misreported.

So it's critical to first establish a process for testing your analytics with each release to ensure error-free implementations. Doing so requires taking stock of:

- **People:** Determine who will be involved, what their roles will be, and how much time will be required of each team member.
- **Documentation:** Ensure up-to-date documentation of the tagging on your site, either by manually creating a tagging plan or via automation, so you can quickly test against your current implementation.
- **When to test:** Test before and after each planned release, and then monitor for unplanned changes on a continuing basis.
- **What to test:** Prioritize testing the most important pages of your site, including critical conversion paths, top pages, and your data layer.
- **How to test:** Testing can be done manually, by crawling the code on each page to locate errors, or testing can be automatically performed with an automated solution (more on this below).

With an automated solution, you can execute an analytics test plan by creating multiple Audits to run immediately before and after release. By testing the most critical pieces of your implementation before moving into production, you can verify that everything is running as expected once you get into production. This greatly increases the likelihood of a successful update to your analytics implementation.

How one Fortune 500 company ran their analytics test plan

When one Fortune 500 finance company was preparing to update the most trafficked user path on their site, they created 42 separate tests to conduct before and after the release. Because of the testing process, they were able push to production efficiently and effectively, resulting in a successful release.

2. Test your implementation in a staging environment.

Before you deploy your analytics implementation onto your live site, you need to test it on a staging site.

Companies who use a staging environment to test their analytics technologies (about 74% of companies) are more likely to catch errors before they are deployed to the live site—where those errors can negatively impact customer experiences, threaten revenue opportunities, and inhibit accurate data collection.

While working within your staging environment, you should run Audits to scan specific sections of your site for potential tagging errors and use Journeys to ensure there are no errors present on your most critical user paths. Taking these precautions will give your technology teams sufficient time to resolve issues before a website or app goes live.

Hint: If you don't have access to a staging site for testing analytics, talk to your IT department, because it's worth the effort to get one set up.

Understanding the Development, Staging, Production Model

In the context of release testing, this model allows you to test implementations before going live and risking public malfunctions that interrupt data collection or customer experiences. These are different environments along the web or app development process, and each environment is used as a testing ground to prepare for the next one:

1. **Development Environment** is the initial experimentation environment where you can build the minimum viable product of a website or app.

2. **Staging Environment** is where properties are prepared to be seen by the public. As mentioned, you should conduct major testing and adjustments here before going live.

3. **Production Environment** is the live version of your site or app and is completely accessible to the public. Any updates to your production environment should be tested in the staging environment first to ensure functionality.

A data governance solution can also monitor tag performance in the production environment (the live version of your site) and notify the appropriate stakeholders if anything goes wrong.

“I can say without a doubt that since we started, with every pending release that was going to production, ObservePoint has caught an issue for us.”

- Senior Analyst, Analytic Technology, Travel Industry ObservePoint Customer

3. Monitor data collection over time.

Once you’ve pushed to production, the work isn’t done. Websites are highly dynamic creatures. New pieces are constantly added and configurations are frequently changed, exposing you to potential breaks that cannot be foreseen.

In order to ensure your implementation continues to catch errors before they result in data problems, you’ll need to set up a monitoring system that verifies your implementation continues to function as expected, notifying you when something goes wrong.

Using an automated data governance solution, you can set up regular Audits to monitor all digital properties for errors and run regular Journeys on the most important conversion paths. By setting up Rules that define which tags should be on which pages, you can automate this process further and be notified whenever a Rule fails. By leveraging ongoing testing, you can consistently trust your data and catch errors before they destroy your credibility.

How one digital analytics team used ObservePoint for ongoing validation

“We invested decent marketing dollars into creating video testimonials, so we implemented analytics on those players to see who watched and what they did as a result. One day, without telling us, the development team made a very minor change to improve the video on mobile and it caused the entire analytics package deployed on the player to break. But because we had an ObservePoint audit running, we caught it that same morning and had it fixed by the end of the day.”

- Manager, Digital Analytics Product Team, Education

Getting Buy-in for a Data Governance Solution

The earlier in development a company properly applies data governance, the less costly and risky it is. However, getting buy-in for a new solution may require presenting a business case that demonstrates the value. Here is a basic outline to get started:

- **Problem/Opportunity:** Errors are 100x more expensive to fix in the production environment. Fixing errors in dev environments instead will significantly reduce costs and improve productivity. Focus on a specific issue based on experiences you've had with IT or DevOps personnel. Get them involved.

- **Solution:** Propose the use of a data governance solution. Show how your solution fits in with your data governance and development objectives to increase productivity by drastically reducing time spent QA testing and minimizing errors at each stage of development.

- **Business Value:** Tie the value proposition of the solution back to your company's long-term goals and objectives, such as correcting issues so you have trustworthy data to make more effective decisions that increase revenue.

Proactively Catch Data Errors to Save Time and Money

The cost of ineffective data governance is much higher than the cost of successfully governing your data. With a 432% ROI, ObservePoint is an automated, enterprise data governance solution that is well worth the investment to validate your data is being collected as expected.

In addition to Audits, Journeys, and Rules, the following benefits of ObservePoint can help you quickly and efficiently catch errors in every phase of development:

- **Testing in Secure Environments:** One of the challenges you might face when testing your pre-production environments is the maze of approvals, authentication, whitelisting, proxy servers, and VPN networks. Thankfully, ObservePoint supports all of those technologies, allowing you to audit in any environment you need to.

- **Automated Release Testing:** To make things even easier, you can automate release testing for before and after regularly scheduled releases. Just schedule your audits and journeys to run before and after the release time, and you won't have to manually run your analytics test plan.

- **Comparison Auditing:** Also related to release testing are comparison audits, in which you can compare the results of a pre-production audit to a production audit and see what changed between environments. This expands your ability to identify changes and potential breaks.

As you follow these guidelines, you will set your implementations up for success in multiple environments and feel confident in your data quality. To see how ObservePoint can help you catch errors and ensure data accuracy, schedule a demo.

Schedule Demo