

A Convizit White Paper

# Using AI to Capture Better Behavioral Data, Faster



# Introduction

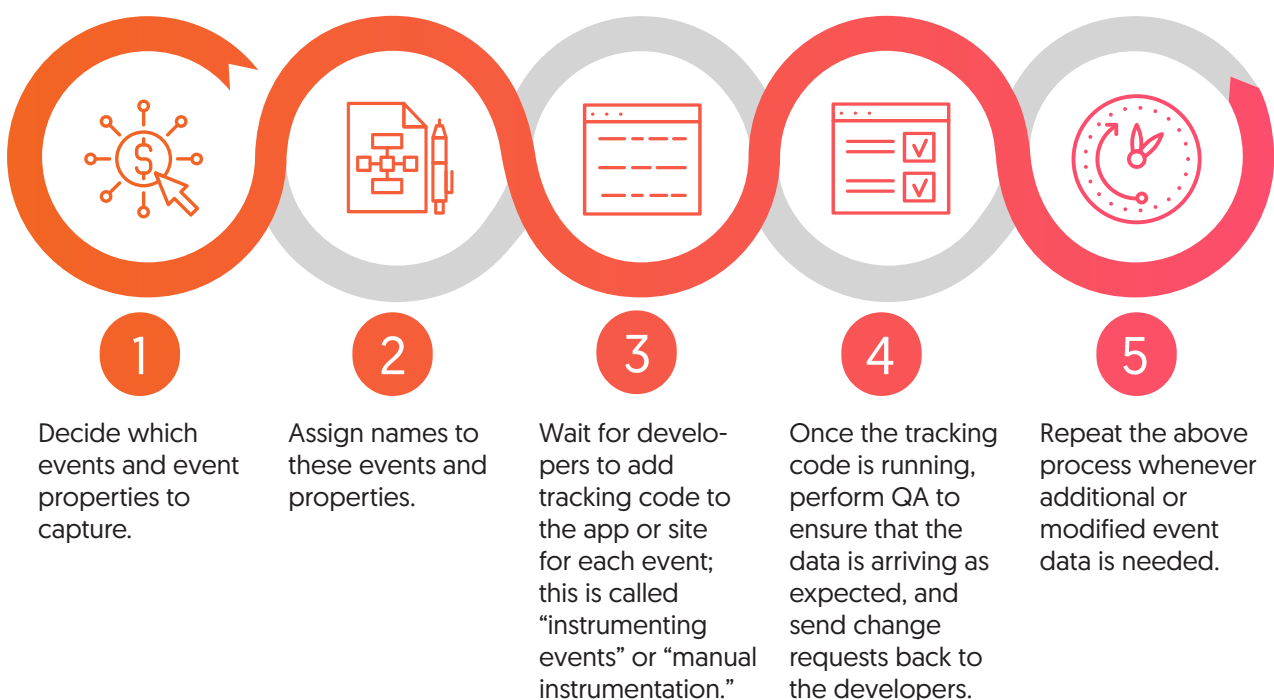
If you are reading this white paper, you already understand how important user behavior analytics is for accomplishing a range of business goals. What you might not know is that the conventional methods for capturing user event data are fraught with difficulties, costs and other shortcomings that have been a fact of life – until now.

This white paper describes the labor-intensive traditional approaches and their downsides, and then proceeds to introduce and compare a new approach, based on artificial intelligence [AI], that eliminates the weaknesses of the manual methods. The AI empowers you to achieve tangible business benefits from your user behavior data more quickly and more easily than was possible before.

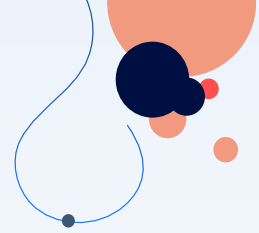
## How Organizations Typically Capture User Event Data in SaaS Applications and Websites

Most organizations implement a product analytics platform for their application or website to address goals in the areas of product management (improving the user experience, conversion rates, customer retention, etc.) or marketing (improving inbound campaigns, allocating ad budgets, etc.). To meet these goals, the org needs to collect and analyze user activity data. Specifically, this data consists of events (individual user actions, such as, “Played video” or “Image clicked”) and event properties (attributes that provide extra meaning for each event, such as, “Product name” or “Selected color”).

The conventional approach to capturing this user behavior data is painfully manual, and typically includes these steps:



# The Downsides of Capturing Events Using Manual Instrumentation



The above process is still the one most commonly used today. Despite its ubiquity, it exhibits some major disadvantages, even when implemented flawlessly:

## Very long time-to-value



Planning every event-tracking decision in advance, relying on busy developers to implement tracking code and manually validating everything is a very time-consuming process. Business users are usually frustrated by the limits and delays involved. These delays in driving improvements incur a concrete cost on the organization.

## Lack of business user autonomy



Business users cannot independently make changes to the data being captured. Ongoing changes – such as adding/altering event details and selecting how events are sent to destination platforms – force business users to wait for technical staff, busy people who have many other priorities.

## Mysterious and duplicated event data



Unless every captured event is well documented, business users won't understand exactly which user action actually generated each event they see in the data. Rare is the organization that is able to maintain accurate, explicitly descriptive, up-to-date and understandable event data documentation over time.

## Unreliable data over time



By their nature, applications and websites change frequently. This causes event tracking to “break” with surprising regularity, following updates. The result is reports and funnels that stop working and/or that present missing or erroneous information. Bad data leads to bad decisions which leads to bad business outcomes.

## Never-ending event tracking maintenance



Multiple people need to be involved with updating and validating event tracking every time an application or website changes. Because it is rare for an organization to be on top of this consistently and comprehensively over time, data quality and completeness usually deteriorates.

## Incomplete user behavior data

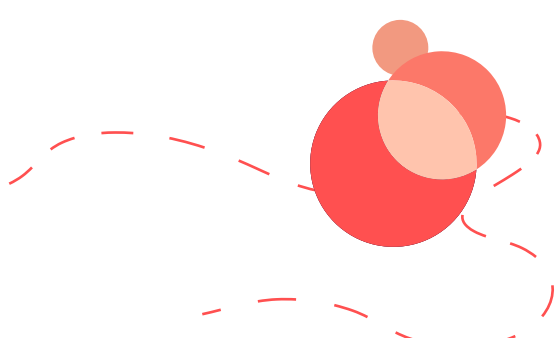


Instead of having easy and immediate access to all events and properties, the conventional process captures only a small subset of actions (and even fewer event properties). Furthermore, there is no access to historical data [i.e., user activity that occurred before tracking for those particular events was implemented].

## Platform lock-in



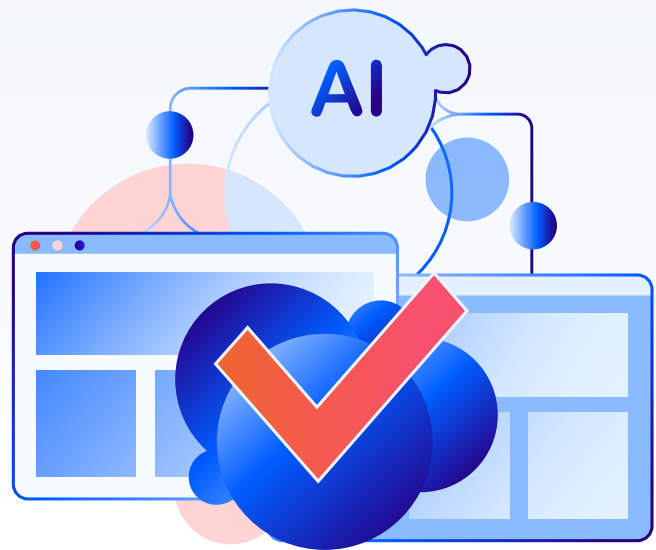
When instrumenting events using a particular platform, the data is proprietary to that platform. If the org wants to change the analytics or CDP platform they're using, for example, they will usually need to re-code event capture for the new system.



# A New AI-driven Approach Eliminates the Weaknesses and Empowers Business Users

Recent technological breakthroughs leveraging AI are now enabling a streamlined and radically more productive approach for user behavior data capture, one that turns the conventional process on its head:

**Instead of starting with a large investment of time, planning and technical setup, AI delivers automation, speed, agility, accuracy, reliability and business-user autonomy.**



With little or no manual effort, this new, AI-enabled methodology begins delivering accurate, property-rich and ready-for-use event data to your analytics platforms of choice, within days. In a game-changing shift better attuned to real-world dynamic responses, product and marketing teams can begin actually using their analytics tools and working on their data-driven projects almost immediately.

Business users can customize how event data is captured and delivered (e.g., selecting and renaming events and properties) without the involvement of developers or other technical people. There is no need to add identifiers to on-page elements inside the app/site. Changes to the event data are applied, and available for use, instantly. Over time, as the user interface of the application or website changes (as they always do!), no user behavior data is lost.



# | The Fundamentals of AI-Driven User Behavior Data Capture

Instead of relying on human effort to add event-generation code or element identifiers to an application or website, the AI-driven approach combines a few technologies that, together, allow the software to “understand” how users interact with an app or site. These technologies, many aspects of which are made possible using machine learning technology, include:

- **Smart auto-capture** tracks every user interaction with every on-page element, and intelligently extracts the most relevant event “value” at the time of the action.
- **Intelligent per-element property extraction** determines which on-screen content presents relevant property data for each event.
- **Automatic element grouping** determines when multiple on-page elements are essentially the same event [even across pages], and treats them as a single event, based on UI functionality and page context [rather than on CSS selectors]. This works regardless of how the page is coded, or if the display is rearranged in different scenarios [e.g., for different screen sizes].
- **Automatic tracking continuity** enables the software to determine that an element remains the same essential element, even after changes to the coding of the element itself or the page around it [e.g., location, CSS styles/classes, IDs, DOM page structure], and thus to continue correctly sending the event and its properties.

The resulting structure of all event data is exposed in a UI that allows business users to review and customize the behavioral data being streamed to their analytics platforms. They obtain this required functionality – which serves setup, QA and data-understanding purposes – without reliance on technical staff or the manual, labor-intensive efforts required today:

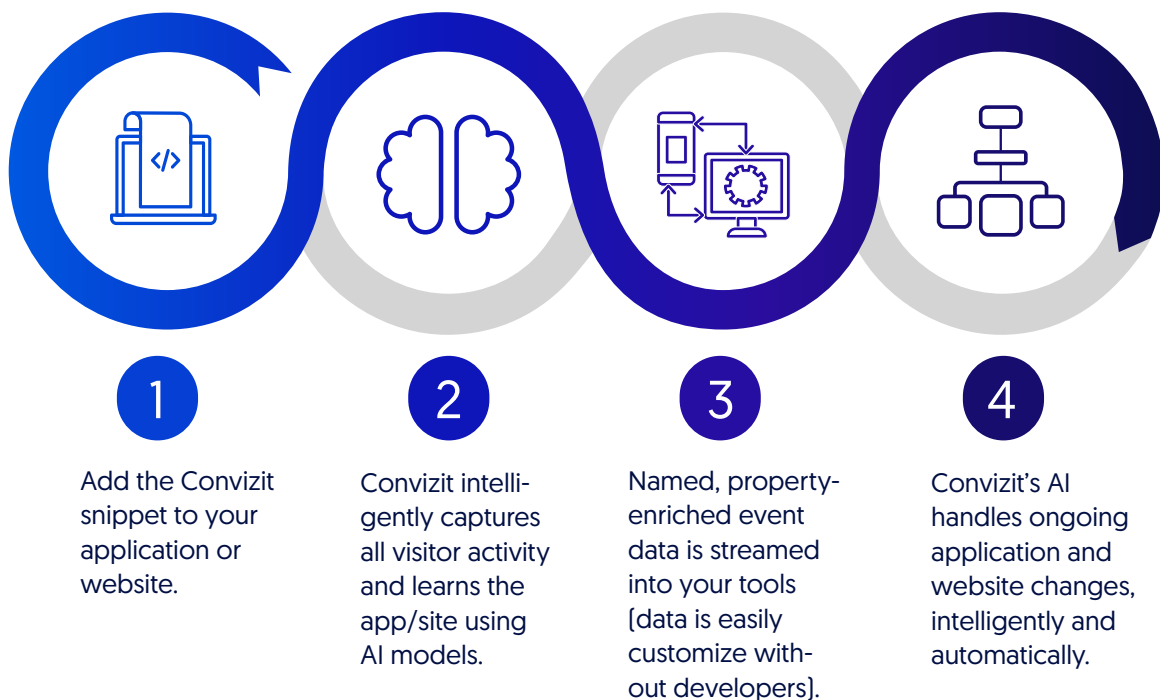
- **Review a catalog of all captured event types.**
- **Identify, in the live app/site, the on-page element that generated each particular event.**
- **Interactively preview the exact event and property data generated for any click on the page.**
- **Rename events and properties.**
- **Select/remove properties for individual events.**
- **Decide which events and properties are sent to each destination platform.**

The rich event data is streamed directly into any analytics, BI, CDP or marketing platforms – wherever and however it is needed.

# | Fast Deployment, Easy Customization

The process of deploying and customizing AI-driven user event capture is straightforward:

- You add two small code snippets to your website/application – one that collects all user activity and one that links user identities. The heavy lifting happens elsewhere, eliminating many of the problems described above. Within minutes, deployment (and, thus, developer involvement) is complete: every user event and event property is intelligently collected and structured.
- The system automatically groups together disparate elements if they are functionally identical (e.g., various Product Images in different areas of a product page, in different layouts of the same product page, and even across product pages will all be automatically grouped together into a single “Product Image click” event, each containing the event and property values that fully identify the particular element that was clicked). This grouping dramatically streamlines tasks, such as selecting or naming events and properties, as one change can be applied to numerous on-page elements. Grouping also avoids the possibility of duplicate or inconsistent event data.
- You can begin using this “default” dataset in *any* analytics and marketing platforms within days of system deployment. [The AI’s machine learning and modeling process typically requires a modest 3-5 days to learn an app/site, depending on its complexity and the amount of actual user activity.]



- Business users have autonomous access, via a friendly, no-code visual tool, to identify the source action of every event, to review how the events are being captured, to select which events are delivered to which analytics and marketing platforms, to select which properties each of those events contains, and to customize how events and properties are named. Different permission levels allow some users to only review the data, others to edit the way events are delivered, and others to serve as admins.
- AI-driven “tracking continuity” technology ensures that event tracking remains complete and consistent following aesthetic or functional coding changes to the user interface of the site or application. Reports and funnels will remain intact and consistent. [Learn more about this topic in our [Tracking Continuity](#) blog post.]
- You can easily “back-fill” historical data [for any event that was not previously included in the data stream] into any destination platform, all the way back to the day the AI event-tracking system was deployed.
- Application/website performance is not affected. Because the lightweight script is only 60KB in size, and functions completely asynchronously, there is no impact on user experience, page load times or SEO rankings.

## | Different Use Cases Demand Different Methodologies

While deep user behavior analysis benefits from having as much event and property data as possible, most analytics and marketing use cases do not require the tracking of every single user action. Whereas a complete behavioral dataset is invaluable for data scientists and advanced analysts looking for correlations in the data or deep user journey analysis [for example], most organizations begin with more modest analytics goals.

The AI-driven approach offers the best of both worlds: for those use cases where having a complete dataset is beneficial, getting every event and property is available, very quickly and easily.

On the other hand, it is also easy to initially focus on only 10-20 events [as is typically required for essential reporting, dashboards and optimization], by using the visual tool to select just the events you want to start with. The beauty here is that you can easily select which events you want to focus on *from within an existing dataset of all events actually occurring in your app/site*; there is no need to delay time-to-value by pre-selecting which events to track, hoping you chose the right ones, and waiting for developers to code your tracking plan.

## | Empowered, Autonomous Business Users

An invaluable benefit here is *business user autonomy*. Product, marketing and analytics people know what they want to know [and are smart enough to let the data tell them about other things they should probably be looking at]. Properly selecting and naming events remains paramount, but the “reverse” AI-driven approach allows business users to do this on their own, without reliance on technical staff or waiting months to begin working with their data.

Additionally, and very importantly, business users can now begin their work based on a readily available, comprehensive behavioral dataset that represents what is actually occurring in the app/site. Even before sending their events into particular destination platforms, they can review and understand the big picture of which user events are occurring, which are occurring most often, which properties are available for each, and so forth.

Having this extensive basis from the get-go provides an important level of guidance towards selecting the most important events and funnels to analyze, as opposed to relying only on *a priori* assumptions regarding what’s seemingly [or hopefully] most important.



## | Added Bonus: Intrinsic Data Governance

Another powerful advantage of the AI-driven approach is the dramatic improvement in *data governance*: AI-generated event data is more reliable and usable without the need for extensive manual data governance efforts.



With event tracking implemented via code or tagging tools, substantial efforts are required for defining, documenting, debugging and reverse-engineering the event data that is being captured. It is common to have duplicate events, mystery [unknown] events, discrepancies between how events are documented and actually captured, and missing events. The reality is that solving these types of data challenges is not what humans do best.

On the other hand, AI-driven event capture solutions exhibit “intrinsic data governance,” by completely eliminating most of the problems and challenges that are the natural results of the traditional approach. In short, not only does AI-driven event capture save a huge amount of time and effort for both tracking implementation and ongoing tracking maintenance, it provides higher-quality, more robust, more trustworthy and more reliable event data over time. [Learn more about this topic in our blog post, [Intrinsic Data Governance](#).]



# What About Tagging Tools?

This white paper would not be complete without mentioning “tagging tools.” Instead of actually adding in-line code to capture event data for user actions, these solutions capture events by relying on the identification of specific on-page elements using technical identifiers, such as HTML tag ID values or CSS selectors. Their client-side code is installed in the application or website and captures events when a user interacts with any elements thus identified.

There are two categories of tagging tools for capturing on-page user events:

## Tag Managers

The screenshot shows a 'Trigger Configuration' window. Under 'Trigger Type', 'Click Link' is selected. It includes options for 'Wait for Tags' (Max wait time 200 milliseconds) and 'Check Validation'. Below, it shows conditions: 'Page URL matches RegEx .\*' and 'This trigger fires on Click URL does not contain [redacted].com'.

## Visual Taggers

The screenshot shows a 'Track User Action' configuration window. The 'Event Name' is 'CTA Clicked'. Under 'Properties', 'title' is set to 'get\_the\_app', 'action' is 'Enter a Value', and 'location' is 'Enter a Value'. Under 'Event Triggers on', 'Only on certain pages' is selected, with the URL 'https://[redacted].com'.

Tag managers provide a UI for more technical users to define event triggers based on identifiers that exist [or must be added] to the webpage’s HTML code. Visual taggers, which are designed for less technical users, provide a “visual tagging” UI tool in which the user clicks on various elements in a SaaS application or website to select them for event capture.

While these tools purport to obviate the need for developers in the event-capture process, this is frequently not the case in the real world; it is often necessary to add identifiers to the HTML code of individual elements in order to reliably and explicitly track user interactions with them. Furthermore, they introduce a number of new problems and difficulties that lead many organizations to look for a superior solution.

The primary downsides of using a tagging tool versus the AI-driven approach are summarized in the table on the next page.

# Manual Tagging Tools vs. Automated AI Capture

## Tag Managers and Visual Taggers

### A completely manual process

The user needs to explicitly select and name each individual on-page element to be tracked.

### Developers are still needed

It is often necessary to add IDs to elements' HTML code in order to accurately capture their CSS selectors.

### No properties are collected automatically

The user needs to manually find and specify the CSS selectors for each property to be included.

### Duplicated events are common

Because event triggers are based on CSS selectors, there are usually multiple ways to identify and name the same event, leading to erroneous or duplicated data.

### Tracking is fragile

Even small changes to the page/element code [CSS selector] can completely break event definitions.

### Heavy tracking maintenance required

Without ongoing and diligent maintenance, data is often lost when changes are made to the app/site.

## AI-driven Automated Event Capture

### A mostly automatic process

All events are captured and ready for use, effortlessly; you can choose and customize events from a "catalog".

### No developers required

Tracking on-page user actions does not require engineering resources.

### All available event properties are captured automatically

You can easily choose which properties are sent with which events to which destination systems.

### No duplicate events

Automatic grouping based on functionality and context (rather than on CSS selectors), makes it impossible for duplicate events to appear.

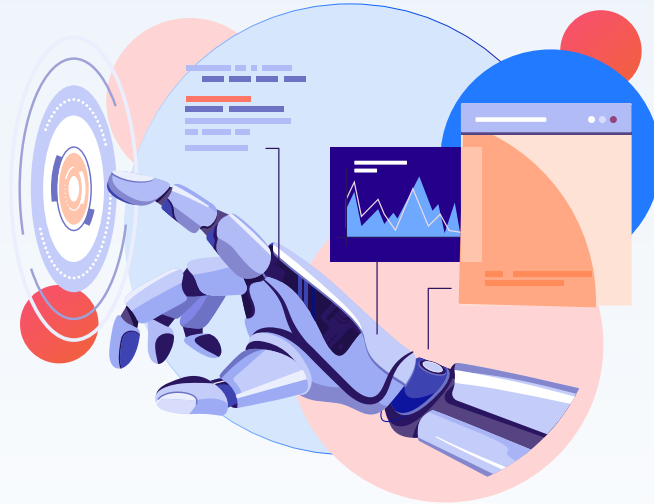
### Strong tracking continuity

Intelligent, automatic understanding of website changes over time means no lost event data.

### Zero maintenance required

All app/site changes are automatically captured, structured and delivered.





## | Conclusion

AI-based solutions are becoming more and more commonplace in our lives. Fields such as knowledge search, speech recognition, image generation, self-driving vehicles and industrial automation are being revolutionized by AI.

Artificial intelligence is especially useful when applied to challenges that involve processing and understanding large amounts of data in relatively unstructured environments. This perfectly describes the challenge of automated user behavior capture – and underscores why there has never before been a sufficiently accurate and reliable solution in this field.

Organizations looking to implement user behavior tracking in their SaaS application or website should consider using the automated, AI-based approach described here to capturing user event data, instead of the completely manual code-based and tagging-based approaches. Here is a summary of the primary advantages:

- Much faster time-to-value [days, not weeks or months]
- Property-enriched event data readily available for every user event
- Complete business user autonomy [no reliance on developers]
- Clearly understandable and usable event data
- Consistently accurate event data [both initially and over time]
- Strong tracking continuity following app/site changes [with zero tracking maintenance]
- Historical event data back-fill
- No platform lock-in [a single source of rich event data for every platform]

To learn more about how AI-driven event tracking can be a game-changer for your organization, **contact Convizit** or **schedule a meeting** today.