WHITE PAPER

revenera

A Buyer's Guide to Software Usage Analytics:

Getting the Insights You Need to Build Better Products and Increase Sales Conversions

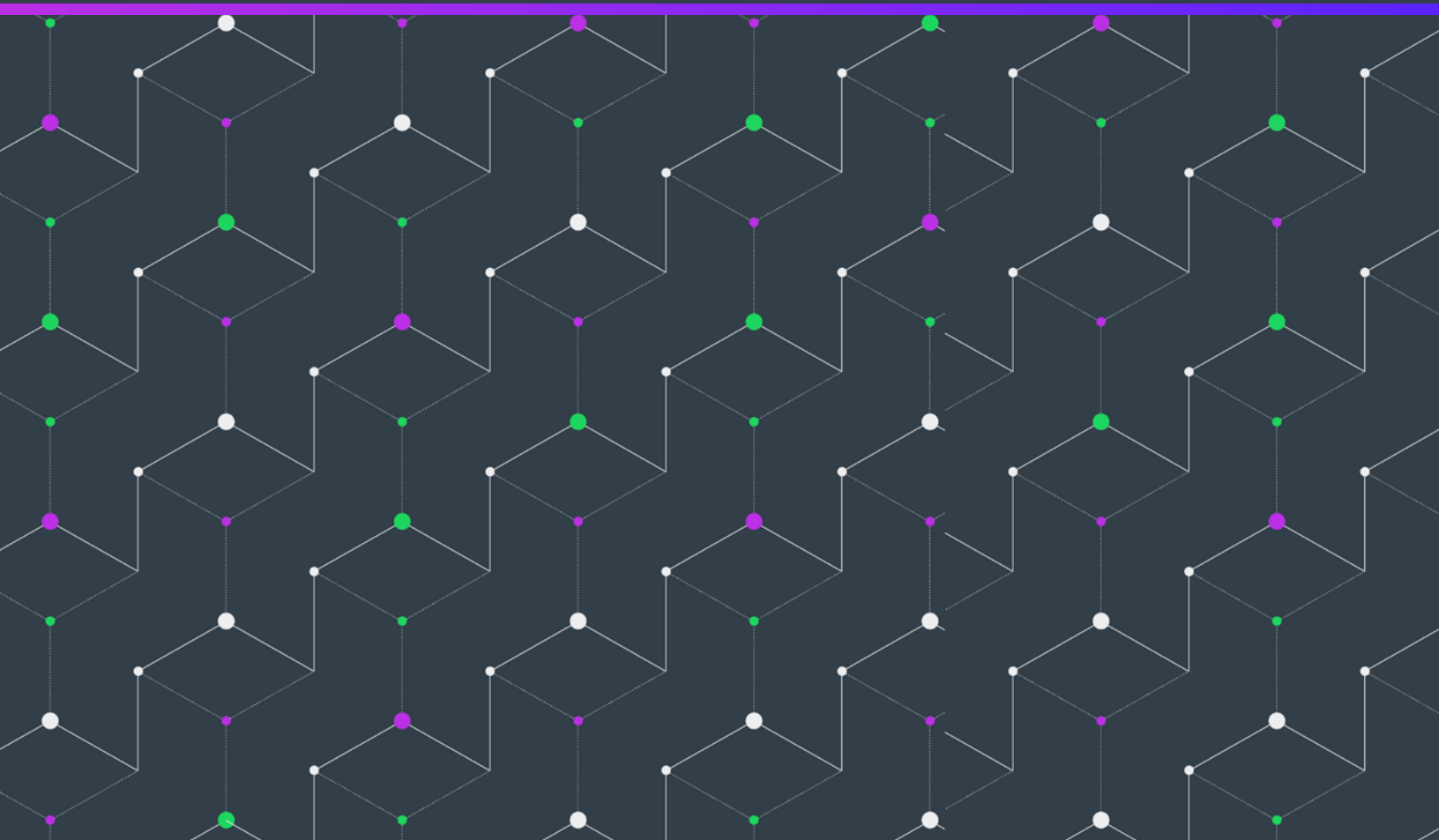

Table of Contents

Sustaining Profitability and Growth in the Software Business is More Challenging than Ever.

To succeed, you need to delight your customers and provide software solutions with exceptional value. This requires product management and marketing professionals to have the best possible understanding of how your customers are reacting to and engaging with your software. You can gain these actionable insights by integrating software usage analytics into your software product.

This white paper explains what software usage analytics is, what it can do for you, and what an optimal solution looks like. It will also help you answer a key question many software organizations ask: Should we build our own, or buy a third-party solution?

What is Software Usage Analytics?

"Software usage analytics" (a.k.a. "software analytics" or "runtime intelligence") involves the monitoring, collection and analysis of data related to software installations, and to end-user activity associated with them. By offering fact-based insights for decision-making that would otherwise be unavailable, software usage analytics helps you:

- Convert more prospects and upgrade more customers
- Improve marketing effectiveness
- Accelerate sales cycles
- Assess and optimize the performance of your sales channels
- Optimize product pricing and packaging strategies
- Evaluate feature usage
- Plan product roadmaps
- Establish feedback loops to optimize investments in early stage products or features
- Sunset legacy features or support for old or problematic platforms

Why Alternatives to Software Usage Analytics Don't Work

Software companies have attempted to understand how users are engaging with their products using diverse methods. These include web analytics tools such as Google Analytics, CRM systems, qualitative sales/ channel feedback, helpdesk support calls, and surveys. None of these is an adequate substitute for software usage analytics.

Web analytics tools such as Google Analytics track website user activity. Since they are focused on pageviews, however, they cannot be effectively used to track desktop applications. To track what happens after a product is downloaded, you must perform clumsy workarounds, capturing events within your application and manually mapping each of them to a pageview with a unique URL.

Each time you want to track an event with Google Analytics, you must make a web request. This requires your software to constantly maintain a live internet connection, generating internet traffic that can slow applications, degrading user experience or even raising suspicions of malware infection.

Moreover, you still don't understand how your software is actually being used. Google Analytics can't maintain user profiles, or generate reports spanning multiple sessions. Since you can't track an individual user's progress across sessions, you can't assess a user's behavior or engagement over their evaluation cycle or lifetime. Therefore, you can't analyze conversion funnels, or detect shifts in feature usage.

CRM systems have other serious limitations. They may require significant customization to serve software product managers, and rely on sales reps to enter new information on an ongoing basis. These customizations and updates may not be top priorities.

Even if customized, CRM systems still don't capture the realities of software usage. They don't illuminate feature usage, trialing experiences, customer base dynamics, or the customer's perspective on what aspects of the product fall short. A CRM system might reveal a downturn in renewals, but offer no actionable insight into the cause: pricing, competition, dissatisfaction with functionality or performance, or something else?

Other inputs, such as helpdesk support calls, capture only an unrepresentative slice of user experience. You learn little about users without "showstopper" problems, or how engagement with your product evolves over time. At best, these approaches produce inconclusive analyses, and at worst yield inaccurate results.

Software Usage Analytics: Answering the Questions that Matter Most

Software usage analytics can answer dozens of crucial 'How and Why' questions about user environments, prospect evaluation and buying cycles, product conversions, renewals and upgrades, feature usage, and software reliability.

Prospects

"Are people seriously evaluating my software, or just running it once and walking away?"

- How can I reach more of the right prospects?
- What is my prospect's evaluation experience?
- How many registered trial users actually install and run the software?
- How many trial users drop after install?
- Why do some stay and others drop? What are they doing during evaluation?
- How can I maximize the effectiveness of the trial period?
- How many people are running each version or edition of my software right now?
- Why do users run my software?

New Users

"What tasks do my new users experiment with first? Are they succeeding? Can I help?"

- How quickly do purchasers configure and start using my product?
- Which features are discovered most often by evaluators?
- Which features are used (or rarely used) after purchase?
- Does feature usage vary by types of user?

Conversions and Sales Cycles

"We had 1,000 downloads and only 50 sales. What happened to the rest?"

- What do trial users do immediately before purchasing or dropping my software?
- Are there patterns of delight or disappointment that lead to wins or losses?
- How long do prospects take to purchase? How can I shorten that time?
- How long do prospects remain in an expired trial before upgrading to paid versions?

Renewals and Upgrades

"We're trying a new offer to increase renewals of lapsed customers. Is it working?"

- How does feature usage vary between customers who renew or don't?
- How many times do users see my expiry notice before renewing? Do my expiry notices impact user behavior?
- Are customers seeing enough value in new releases?
- How quickly is my customer base adopting new versions?

Marketing and Sales Performance

"Why are users in Germany converting faster than in the US?"

- Of those who downloaded based on my current campaign, how many installed and used the product? Did they convert?
- What usage trends exist—by region, product, version, language, or over time?
- Is sales performance (relative to trial installs) changing for specific regions, territories, channels, products, or versions?
- Should I customize product packages to reflect differing regional usage patterns?

User Environments

"Where should I focus my support and QA resources?"

- What resources will be available to my software? (e.g., laptops vs. desktops, memory, CPU/GPU, screen resolution, etc.)
- Which OS platforms are most popular and deserve the highest QA priority?

Feature Usage and Product Planning

"Should we invest in building out our collaboration features? Is anyone actually using them?"

- Is my market split into different groups using different capabilities?
- Are certain types of customers using only a narrow set of features?
- Are important features getting lost in my UI? Can I update the UI to better expose them? Or find better ways to educate my users about them?
- Which features and legacy platforms are rarely used and can be safely abandoned?
- How can we optimize our product roadmap and investments without alienating paying customers?

Reliability

"I'm hearing about more crashes—but is there a pattern to where they're happening?"

- Are new software reliability issues emerging?
- If so, are they appearing on specific hardware/software environments?
- How many customers are impacted? How should we prioritize remediation?

Collecting and Managing Data

A successful software usage analytics system is built upon a strong foundation of data. Your solution must efficiently collect and process massive amounts of usage and performance data, and store the data in a way that can be used conveniently for advanced reporting and analysis. Although many companies collect data, understanding when and what to collect, and investing in resources to store and manage that data can be daunting.

Don't underestimate the quantity or complexity of the data that must be managed. Even a few thousand installations can generate big data issues requiring significant expertise. Simple relational databases such as MySQL might store enough data, but fail to perform when you attempt to build scalable real-time reports. If you need to handle huge datasets with specialized NoSQL databases, you may need new skills.

Organizations often underestimate the staff resources required to manage home-built analytics systems. These often require the company to reserve one or two engineering team members to manage and indefinitely maintain reporting. This can become a significant investment, especially if not an organizational core competency.

Data collection considerations include:

- Does an SDK exist for handling the client side of data collection? Or must you create the code and infrastructure yourself?
- What kinds of data will you need, and can you collect it? For example, can you track a wide variety of application events?
- Can you capture data that is custom to your application, such as user-specified configuration settings, the state of your application when a specific event occurs, or user feedback sent via inapp messaging?
- What actionable metrics should you collect? Conversely, what is useless noise? (There's no point in collecting, storing, and processing data that doesn't add business value.)
- Must your solution maintain a constant Internet connection to collect usage data? Or can it intelligently cache data locally until it can be forwarded to a network server, without interfering with user experience?
- How will the system handle offline use, dark networks, or firewalls/web filtering gateways?
- How efficient, reliable, and secure is the clientserver communication protocol?

Data storage and management considerations include:

- Does an SDK exist for handling the client side of data collection? Or must you create the code and infrastructure yourself?
- What kinds of data will you need, and can you collect it? For example, can you track a wide variety of application events?
- Can you capture data that is custom to your application, such as user-specified configuration settings, the state of your application when a specific event occurs, or user feedback sent via inapp messaging?
- What actionable metrics should you collect? Conversely, what is useless noise? (There's no point in collecting, storing, and processing data that doesn't add business value.)
- Must your solution maintain a constant Internet connection to collect usage data? Or can it intelligently cache data locally until it can be forwarded to a network server, without interfering with user experience?
- How will the system handle offline use, dark networks, or firewalls/web filtering gateways?
- How efficient, reliable, and secure is the client-server communication protocol?

Analyzing the Data

Your goal is not merely to capture and display data. You need insights that your organization can act upon. Your product managers and marketers shouldn't have to write SQL queries, manually join datasets, or build complex Excel PivotTables to get these insights. Reporting should be simple and intuitive for non-experts, and operate in real-time so you can act immediately.

It should be easy to generate actionable reports that instantly call attention to trends and anomalies, and to augment these reports with context. For example, if you can combine the anonymous user profile details built over multiple sessions with a user's behavior in his latest session, you can often discover the implications of his latest actions. Is he following a pattern that has led others to drop the software? Can you intervene, and move him back towards purchase?

Visualization is often the most effective way to communicate software usage analytics insights. Many software organizations, recognizing the value of a powerful and flexible visualization dashboard, envision building an in-house solution based on a generic visualization framework. This may not be as easy as it seems.

First, your organization must master the framework's API and maintain those skills indefinitely. Second, additional licensing, server, and administration costs will be associated with your preferred visualization toolkit.

Most important, a visualization framework merely lets you convert raw data into attractive charts. By itself, it doesn't add intelligence or context, or answer the questions that matter. For that, you need actionable interactive reports designed to answer specific questions, with easy drilldowns that make it intuitive to explore data in greater detail. Building such actionable reports will require time, and knowledge of the framework. It will also deep knowledge of what each potential user needs to know, both now and in the future, as they start asking more advanced BI questions.

If you're considering a third-party solution, ask:

- Does its built-in visualizations and reports answer the business and technical questions I care about?
- Are there easy point-and-click tools my product team can use to customize new reports on their own, whenever they need to answer a new BI question?
- Can my product team easily filter or segment to uncover differences between different kinds of users (e.g., home vs. corporate, novice vs. experienced, licensed vs. trial?) Or must they depend on technical experts to perform these tasks?
- Will my reports run responsively in real time? (With huge datasets, reporting can be painfully slow unless the analytics engine, supporting backend and queries, has been carefully optimized.)

If you're considering creating your own solution, ask:

- What would it take me to develop and maintain the infrastructure and reporting system myself?
- What will be my time to market? How quickly will I benefit from new intelligence?
- As our needs change, how will we handle customizations? Who will do this?
- Can I cost-effectively guarantee real-time performance and scalability?

Choosing a Solution: Additional Considerations

Beyond data, several additional issues are key in determining whether to build or buy your solution. Assessing these will help you select a solution that can be implemented quickly, work reliably and cost-effectively, and is easy to manage.

Solution environment, performance and support

Adding software analytics capabilities to your applications goes beyond adding simple data collection and egress functionality. Be sure to consider these factors as you choose the best approach.

- Was the solution designed specifically to support my client environment? (For example, data collection/analytics systems built for web applications require complex workarounds to collect data from desktop applications. Even after these workarounds are implemented, they may not capture all the data you need, or work as flexibly as you want.)
- Does the solution's code or communication requirements compromise performance or user experience?
- Is all specialized functionality built in? (For example: runtime tracking, geolocation, proxy support, handling of firewalls, or offline caching when Internet connectivity is unavailable.) Or are you prepared to build, test, and integrate this functionality yourself?
- Who will provide for hosting, secure backup, updates, and administration?
- Does the system have automatic failover built-in? What failover plans are in place?
- Who will manage changes between product releases, including updates to client-server communication without breaking backward compatibility with older clients?
- How will we accommodate usage spikes and uneven traffic patterns?
- Who will monitor servers and running services to ensure the system is working properly with no impact on users?

Integration

Prioritizing application development resources is a balancing act between addressing known issues and adding new features.

- How long will it take to integrate analytics into your products and your infrastructure? How quickly can you get actionable data and insights?
- How much code must be integrated? How complex is this?
- Is integration non-disruptive to existing code and user experience?
- How will you test your system for load, security, and other factors?
- How much QA will be required before you can ship telemetry code in your live product?
- How easy will it be to integrate telemetry for new product features?

Determine how easy it will be to integrate your data, reporting, and visualizations with external products or systems. Here again, ease of integration directly affects time to value. Ask your potential partner or in-house developer:

- Does a fully-documented API exist, or must we continually rely on the solution's builders?
- Can I easily integrate reporting data into existing or new dashboards, via iframe or other means?
- Can I integrate visualizations or raw data into existing solutions such as CRM dashboards, quickly use analytics data in other business processes, parse it with other tools, or aggregate it with intelligence from other systems?

Security and privacy

As they consider software usage analytics, most software companies are deeply concerned with privacy and security. They have serious legal and regulatory responsibilities to customers and prospects; failures to abide by privacy regulations can damage the business and its reputation. In considering whether to build or buy, ask:

- Do you have confidence that privacy and security will be professionally handled in alignment with best practices and all legal requirements where you operate?
- Who will implement, test, and integrate privacy and security features? Is this built into the solution? If you're considering building your own, do you have the specialized security and legal expertise?
- Does the system capture and maintain data anonymously by default to support strict privacy regulations such as those in the EU?
- Are IP addresses used only for geolocation, or are they stored in the database? (IP addresses are regarded as user-identifiable information; storing them may breach privacy laws in some jurisdictions.)
- Are license keys or other sensitive data encrypted in transit?
- If a user wants to see the data that's being collected, can your solution permit this?
- Can you easily turn tracking on or off based on each user's opt-in/opt-out?
- How will your solution prevent staff (or others) from seeing reports they shouldn't see?
- Can you provide timely access to specific reports for whoever needs them? Or does the system create bottlenecks by limiting all report generation to just a few authorized users?

Communicating with customers through the product

If your software usage analytics solution supports one-way and two-way in-application messaging, you can communicate directly with users to solve problems, build awareness, highlight features, provide tips, ask questions, convert, upsell, and cross-sell.

In planning for in-app messaging as part of your analytics solution, consider these issues:

- Can you target in-app messages to specific users or groups of users? For example, could you create a 48-hour sale for users based on how long it's been since they began a trial?
- Can you automatically trigger messages based on the user's activity? For instance, can you call attention to features a prospect hasn't tried?
- Do you have control over exactly how, where, and when your message appears?
- Can marketers and product managers create automated campaigns without specialized technical skills, or support from the engineering team?
- Can you collect in-app user feedback through bilateral communications? For example, can you run surveys or collect freeform text feedback from within your application UI?
- Can you link user feedback to user profile data? For example, if a user reports that "your software is slow," can you match that against the type of machine and OS he is running?

Choosing Your Best Partner

Building software usage analytics capability requires a wide range of technical expertise. Few software companies possess the entire range. Given the need to prioritize development of features customers want, many choose to partner with an expert rather than attempting to build from scratch, or to cobble together a solution from diverse libraries and applications.

Experience matters. If you are considering a partner, ask:

- How many software usage analytics solutions have you implemented? How do they compare with mine?
- Have you worked with large, well-known customers? Have you managed large datasets spanning multiple geographies worldwide?
- What experience does your data analytics team possess?
- How will you ensure the performance, reliability, scalability, and security I need?
- How does your offering reflect what you've learned from your customers?
- What is your roadmap for future development? How can I influence it?
- How long have you been specialized in the analytics business? Are you creating your first iteration of software usage analytics, or have you already learned from experience through multiple iterations?

Revenera Usage Intelligence: Your Best Solution, Right Out of the Box

To help you evaluate Usage Intelligence, we've summarized its key features and advantages:

Data collection and management

- Track unique installs and uninstalls
- Track customer events, including session starts, stops, specific feature usage, usage patterns, and trends
- Collect in-app user feedback
- Track software exceptions as they occur, with full stack trace info and detailed environment data for replicating and fixing problems before customers report them
- Track license keys in use; identify abused keys and/or multiple installations sharing a key
- Lightweight, robust client SDK for handling data collection and client-server communication in distributed applications; already deployed to 40MM+ installations
- Intelligent caching reduces call-homes generated by each client, while supporting offline data tracking, with auto-resubmission when the client returns online
- Supports distributed C/C++, .NET, Obj-C and native Java applications on Windows, Macintosh, and Linux

Analysis and intelligence

- State-of-the-art dashboard with comprehensive built-in reports and visualizations, each answering a question that is important to your business
- Interactive Conversion Funnel highlighting user behavior throughout their evaluation lifecycle
- Churn, engagement and user loyalty reports for various user segments
- Compare product adoption and activity between versions, editions, and builds
- Analyze cohorts, trends, and historical data
- Identify most and least popular features
- Learn about users' hardware/software environments and geographic location
- Generate real-time reports with optimized performance
- Generate custom reports easily via point-and-click filtering and segmentation (no need to waste time coding your own dashboard)
- Collaborate in teams and share reports with granular user access control

Integration

- Powerful web reporting API for offline data export or integration with third party solutions
- Every report can be easily shared or embedded into your CRMs (such as a Salesforce dashboard) by simply copying a URL
- Risk free evaluation with just 30-minute integration
- No disruption to user experience or product performance

Security and privacy

- Extensively tested security and privacy features
- Anonymous data capture without storing IP addresses or user-identifiable data
- License keys and sensitive data encrypted in transit
- Log data stored in clear text on the client, giving end-users transparency into what data is collected
- Developer can switch off tracking for opt-out users

ReachOut™ in-app messaging and feedback collection

- Target by geolocation, language, product version/edition, build, license status, running time, days since install, OS types, or hardware profiles
- Trigger messages based on feature usage and application engagement
- Full control over message appearance and timing
- Support for surveys and in-app user
- No coding required to create or execute campaigns
- 100% visibility and delivery rate guaranteed

Revenera provides the enabling technology to take products to market fast, unlock the value of your IP and accelerate revenue growth—from the edge to the cloud. **www.revenera.com**