



APM Buyers Guide 2021

Table of Contents

Executive Summary	4
What is APM: Overview	5
3 Types of APM monitoring tools	5
Understanding the “why” as fast as possible	5
Why transactions are slow or failing	6
Why Do You Need APM?	6
Better end-user experience	6
Greater customer satisfaction/meet SLA requirements	7
Higher developer productivity	7
Decreased reliance on costly tools	8
Faster innovation	9
Collaboration between Ops and Dev	9
Business Continuity/Reduced Downtime	10
ROI	10
8 Essential APM Features	11
1. Performance & Monitoring	11
Performance of every web request and transaction	11
Usage and performance of all application dependencies	11
2. Code-level performance profiling	12
Detailed traces of individual web requests or transactions	12
3. App and server monitoring and metrics	12

Table of Contents

4. Log Management	13
Bonus Feature: Structured Logging	13
5. Error Monitoring	14
Bonus Feature: View Logs & Errors in code profiling traces	14
6. Metrics	14
Application framework metrics	14
Custom applications metrics created by the dev team or business	14
Bonus Feature: App Scoring	15
7. Deployment tracking	15
8. Real User Monitoring	16
Top APM Tools & Solutions	16
Retrace	16
APM is Affordable For All Dev Teams	17
Advice on Choosing an APM Tool by Role	17
Developers/architects	17
Operations	17
About Stackify by Netreo	18

Executive Summary

Modern businesses are dependent on software applications. Ensuring that all of your organization's mission-critical applications are running optimally at all times is a top priority. The right application performance management (APM) tool will monitor your applications proactively and reactively so you can sleep better at night.

[Retrace](#) offers a developer-friendly cloud-based solution that fully integrates APM with error tracking, logging, and detailed traces of what your code is doing. Retrace allows dev teams to easily monitor, detect and resolve application issues before they affect the business to ensure a better end-user experience. The end goal with APM is to help developers understand every millisecond spent in their code.

Observations of the current software development market indicate that highly successful development teams are shifting left in order to reduce costs and time spent fixing issues in production. APMs like Retrace are vital to these teams for that purpose. A [Retrace](#) user testifies that “Stackify by Netreo has helped us improve the quality of our application in many ways. We are a lot more efficient, to a point where we are being proactive with issues rather than hearing about them from customers/customer support.”

There is a wide range of application performance management and application monitoring tools on the market available for developers, DevOps teams, and traditional IT operations. Historically, APM tools have been used by and designed for IT operations. They have been great at monitoring applications and alerting someone about performance degradation and a select few were able to pinpoint the root cause of problems.

Newer APM tools, like [Retrace](#), are designed to be used from development and QA to production in order to identify and fix performance issues and bugs earlier in the life cycle.

Since there is a lot of gray area as to what APM is and who it benefits within an organization, it can be difficult to compare features, costs, and how each tool could potentially work with your technology stack.

The primary purpose of this ebook is to provide readers with a framework to evaluate the potential benefits and costs of an APM solution. To better understand the benefits and necessary features, we've aggregated several internal and external sources in this ebook so you can make an informed decision.

We'll start by defining what APM is and the different types, share the primary benefits of using APM, list the top APM tools along with their features, and close with sharing tips for selecting a tool.

What is APM: Overview

The term APM is largely an industry or vendor-created term for anything that has to do with managing or monitoring the performance of your code, application dependencies, transaction times, and overall user experiences.

According to [Tech Target](#), “Application performance monitoring (APM) is the collection of tools and processes designed to help information technology (IT) professionals ensure that the applications users work with meet performance standards and provide a valuable user experience (UX).”

Since APM is a ubiquitous term for anything and everything performance-related, some vendors use the term to mean totally different things. APM can span several different types of vendor solutions.

3 Types of APM monitoring tools

- **App Metrics-based** – Several tools use various server and app metrics and call it APM. At best they can tell you how many requests your app gets, and potentially which URLs might be slow. Since they don’t do code-level profiling, they can’t tell you why.
- **Code-level performance** – Code profiling and transaction tracing tools such as Retrace are used to help developers and DevOps teams with code-level performance.
- **Network-based** – NPM tools measure application performance based on network traffic either via synthetic transactions or real user monitoring.

Some other tools do monitoring based on server and application metrics, not code-level performance, and sometimes refer to their products as application performance monitoring solutions. Knowing your server CPU or average response of your web server is important and helpful, but APM aims to go way deeper.

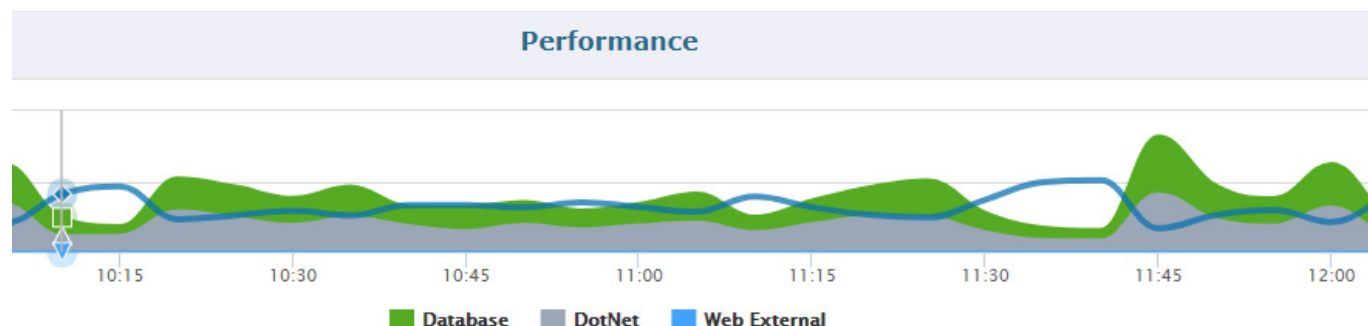
By leveraging [code profiling](#) and other data collection techniques, application performance monitoring tools can provide detailed transaction tracing.

Understanding the “why” as fast as possible

If you want to measure the performance of a web application, it is pretty trivial to parse the access logs and get an idea of how long web requests take. This would give you an idea about overall performance and which pages are slow. Unfortunately, it doesn’t answer the key question of why.

Why transactions are slow or failing

For example, a development or operations team can instantly tell from this visual that their database is causing some performance spikes. They can also leverage their APM to identify exactly which database query and web requests were affected.



Screenshot from Retrace

APM solutions can help identify common application problems quickly:

1. Track overall application usage to understand spikes in traffic
2. Find slowness or connection problems with application dependencies including SQL, queues, caching, etc
3. Identify slow SQL queries
4. Find highest volume and slowest web pages or transactions

Why Do You Need APM?

There are several ways APM tools provide a high return on investment (ROI). They can help with developer productivity, prevent application problems, reduce hosting costs, optimize performance and prevent costly downtime.

Most importantly, APM tools can help you sleep at night.

Here are some more reasons.

Better end-user experience

Users love fast software. It creates an overall higher sentiment for your product. It may be hard to quantify, but it helps with customer conversions and retention.

Slow performance can impact your bottom line. Amazon found every 100ms of latency cost them 1% in sales. We can all relate to trying to buy something online and stopping because it was taking too long.

It could be that the kids are screaming, it's time to go to dinner, or some other reason. We sometimes tell ourselves we will look at it later when we have more time. Many times we don't remember, or spend our money on something else. "I almost bought one of those" is the last thing any retailer wants to hear.

One of our clients provides small loans to their customers online. Their website was taking 10-15 seconds to load and couldn't figure out why. They were able to use Retrace to identify that caching was not working properly. After applying the fix, they were immediately able to see a substantial increase in business.

Greater customer satisfaction/meet SLA requirements

Many B2B companies have service-level agreements (SLAs) with their partners. These agreements typically have clear penalties written into them if their software is not online and working properly.

A short [outage by Amazon AWS](#) reportedly cost them 2% of their total revenues due to SLA credits. The five-hour outage likely cost them millions in revenue and caused problems for many clients, including Apple, Adobe and Netflix.

Amazon reportedly had to refund 10-30% in service credits. Many vendors offer service credits based on how bad the SLA was missed. Here is an example of a table that defines how a breach in SLA is handled.

Percentage Uptime	Percentage Credit
98% or greater but < 99.7%	2%
97% or greater but < 98%	3%
96% or greater but < 97%	5%
94% or greater but < 96%	10%
90% or greater but < 94%	50%
Less than 90%	100%

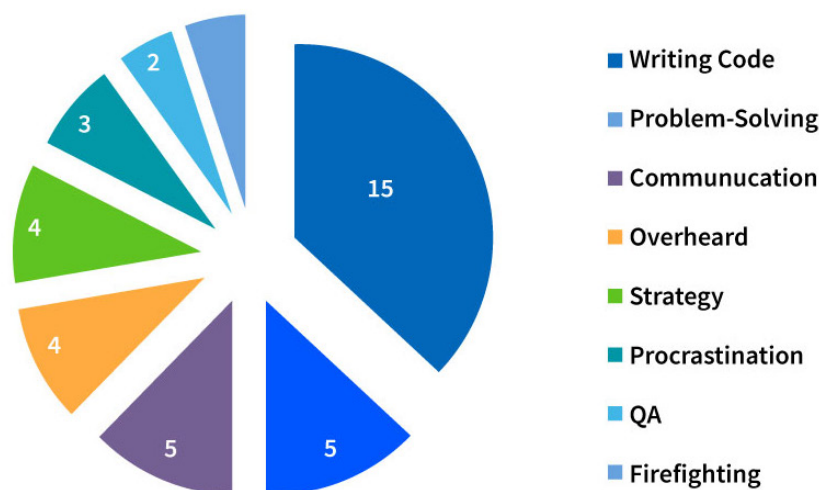
Higher developer productivity

Software developers are expensive. They are also a highly limited resource in today's economy. It is important to keep them working on innovating new products that can grow your business. [Developer tools](#) that make them more productive are highly valuable.

Solving production problems can be very hard and time-consuming. APM tools are designed to help developers quickly identify application problems.

If APM can save your developers a few hours of time a month, it is easy to see how quickly it pays for itself. ZeroTurnaround's [developer productivity report](#) showed that the average developer spends at least a couple hours a week firefighting production problems.

Mapping the Developer Work Week (median hours/week)



Source: ZeroTurnaround's Developer Productivity Report

Decreased reliance on costly tools

APM products are very helpful for measuring the performance of your applications and helping to identify opportunities for improvement. A SQL query tweak here, some code refactoring there, and you might be able to lower your hosting costs through some optimizations.

For example, at Netreo one application ran on about 20 servers. By using Retrace to identify potential performance optimizations, developers were able to refactor some code and reduce the number of servers by 50%. That simple change saved \$2,000 a month.

APM tools can help you understand how your applications use SQL databases, Elasticsearch, web services, and much more. We hear all the time from clients that they had no idea how many [SQL queries](#) their application was running or how slow the queries were. A little performance tuning around [application dependencies](#) can improve overall performance and allow you to scale down those dependencies.

Faster innovation

Every company today should be terrified of disruption by a faster, more innovative competitor.

In fact, as you read this article, Amazon is busy deploying a new release every second, and odds are they're already planning to move into your industry. Any company that doesn't keep a healthy fear of disruption — and speed up their own innovation — will be swallowed up soon enough.

Collaboration between Ops and Dev

The goal of DevOps is collaboration and getting developers more involved in the deployment process and application monitoring. APM tools, like [Retrace](#), can help give developers the insights they need to troubleshoot application problems, without giving them administrator level access.

When development and Ops teams use the same toolset to track performance and pin down defects from inception to the retirement of an application, this provides a common language and faster handoffs between teams.

APM tools, like Retrace, can be used in development, QA, and production. This keeps everyone using the same toolset across the entire development lifecycle.

In a non-unified environment, each team would need to reproduce errors, recapture the logs, and reanalyze the data in their own toolset.

Traditionally, it would be like this:

- Ops, using APM, notices something trending slow.
- Ops then opens a ticket with all the details they can extract and sends it down the pipeline.
- Either the development lead will review the ticket, ensure it has all the information she wants and then dispatch it, or ask for more information, which Ops probably doesn't have.
- Developer accepts the ticket, reads the long log files, reviews any screenshots, and then manually searches other environments to see if the error is happening anywhere else. If not, she will need to reproduce the error to make sure they can increase logging levels to diagnose the error so they can triage and then resolve it.

With a unified toolset used across all environments, the scenario runs more like:

- Ops, using APM, finds an anomaly.
- Ops notes the date/time and opens a ticket with just that information. The development lead immediately dispatches the ticket with no review required.
- The developer, using the APM tool, finds the anomaly immediately and then starts to extract the information she will find useful.
- In the same tool, the developer also has the ability to search for other instances of that error and even increase the frequency and level of logging in production to capture additional information points to aid in triage, and ultimately root cause analysis.

The developer is productive on the issue in minutes, not hours or days. Application performance management tools, like Retrace & Prefix, help development and operations teams better collaborate across all deployment environments.

Business Continuity/Reduced Downtime

**NOTE: The following information is excerpted from Application Performance Management via PCMag's Business Software Index*

Ensure that your APM can assist you with preventing unplanned downtime while minimizing planned downtime.

You need to understand your user's experience and business processes to ensure that you can discover as many issues as possible before your customers are aware of them. That's the easiest way to avoid negative experiences for your customers. This includes being able to determine the root cause when a transaction isn't completed. Did the customer abandon the transaction, or did the transaction fail, causing the customer to abandon their cart?

Your APM needs to be able to deliver actionable data to your IT team, so having the capability to analyze raw performance data and convert it into usable information is important. That means that your customer click rates and click response times become reports on what users clicked on, how quickly apps responded to the clicks and a click-stream analysis.

ROI

It has been said that nearly every business is now a software business in some form or another. That means that the reliability and performance of their software applications are critical to their success. Unfortunately, many [APM](#) tools have been very expensive and targeted at only large enterprises.

The price of APM for 20 servers can range from \$500 to \$6,000 a month. Some vendors also require being paid annually. It is common for us to hear from customers that they can try our product, [Retrace](#), for a few hundred dollars a month, or pay another vendor \$25,000 and they are stuck in an annual contract.

APM solutions [can be affordable](#) and have a priceless return on investment (ROI) if used to their full potential. In the next section, we are going to discuss some of the key features of good APMs and how they provide an excellent ROI.

Check out the [ROI Calculator](#) to see how you can improve while saving money with Retrace.

8 Essential APM Features

For developers, APM is really all about data, and I mean lots of data. But they need more than data, they need actionable insights from that data so they can quickly get to the root cause of what is causing application problems.

Here are some of the key features that most of them support.

1. Performance & Monitoring

Performance of every web request and transaction

At the heart of APM you have to be able to measure the performance of every web request and transaction in your application. You can then use this to understand which requests are accessed the most, which are the slowest, and which ones you should add to your backlog to improve.

Knowing the performance of every web request is just the start though. You could potentially get that from a web server access log. The real key is understanding the why.

Usage and performance of all application dependencies

Why your application is slow usually comes down to a spike in traffic or a problem with one of your [application dependencies](#), like databases, web services, caching, etc.

It is very common to have these types of problems:

- A particular [SQL query is slow](#)
- SQL database server is down
- External HTTP web services calls are failing
- Noisy neighbors in the cloud causing problems

As one example, we recently had some issues accessing our CRM's API. They were throttling us and the only way we would have ever known is because we track all of the exceptions and can see in our APM that those affected transactions were also failing.

2. Code-level performance profiling

If you want to understand why your application is slow, throwing errors, or has weird bugs in it, you have to get down to the code level. Knowing that a certain web request doesn't work is important and actually pretty easy. Figuring out why it doesn't work is hard, sometimes really hard.

By tracking what your application is doing all the way down to the [code level](#), you can potentially gain way more insights about what is occurring:

- What key methods in your code are even being called?
- Which methods are slow?
- Is your app slow due to things like JIT, garbage collection, etc?
- What dependencies are being called?

Detailed traces of individual web requests or transactions

Troubleshooting problems in production are very difficult. [Transaction tracing](#) makes this a lot easier by being able to see details about exactly what is happening in your code and how that affects your users.

Traces can contain these types of data:

- Web request info like URL, etc
- Who the user was
- What dependencies did your code call (SQL, caching, HTTP calls, etc)
- Logging statements
- Application errors
- Key methods in your code

Seeing all of this data in a single trace can short circuit having to attempt reproducing a problem in QA. Getting to the root cause can be nearly instantaneous with an APM solution that collects detailed traces.

3. App and server monitoring and metrics

Application problems can occur for a lot of reasons like CPU, memory, etc.

Thanks to virtualization and the cloud, a server going down isn't near as common these days. However, it still does happen and is something you need to monitor for. It is also critical to monitor things like [server CPU and memory](#).

A lot of modern web applications are not usually CPU-bound, but they can still use a lot of CPU and it is a useful indicator for auto-scaling your application in the cloud.

4. Log Management

Whenever something goes wrong in production, the first thing you hear a developer say is “send me the logs”. Log data is usually the eyes and the ears of developers once the application is deployed.

Developers need access to their logs via a centralized logging solution like a log management product. Fortunately, [log management](#) is an included APM feature in [Retrace](#). Most APM solutions don’t support the #1 thing developers want to see... their logs!

Bonus Feature: Structured Logging

If you haven’t used structured logging, you are missing out! The goal of structured logging is to log “properties” or “objects” so that you can later search for those fields, or do more advanced analytics on them.

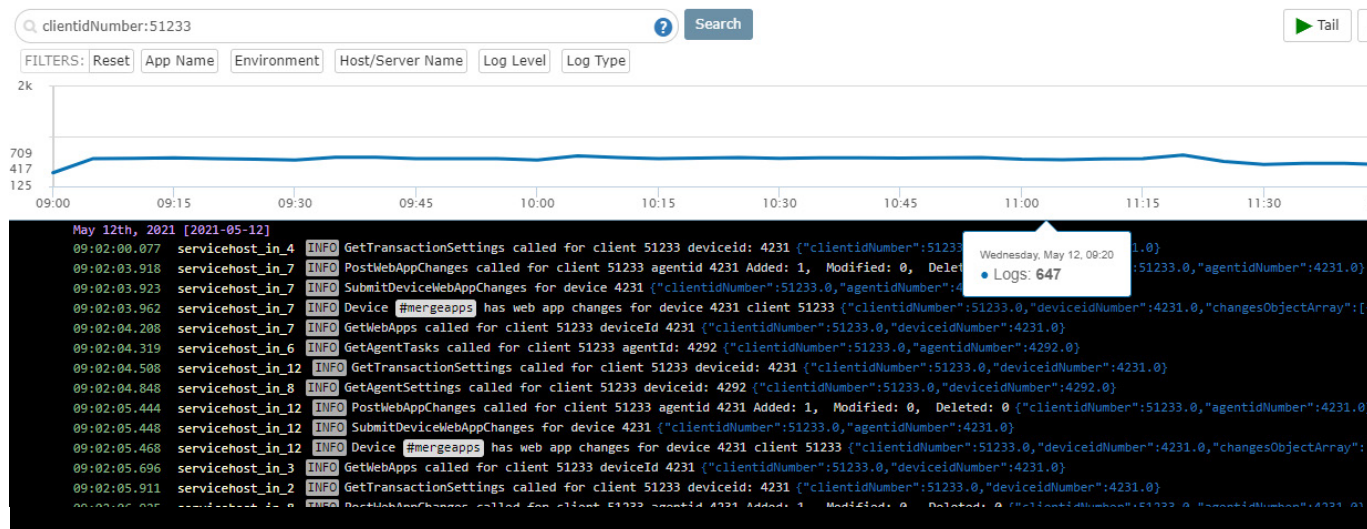
For example, at Netroo we use this to always log the clientid along with our logging messages.

```
log.debug("Incoming metrics data", new {clientid=54732});
```

This enables us to search our logs to only see log messages that are filtered down by that clientid. This makes it a lot easier to troubleshoot issues specific to a certain client.

5. Error Monitoring

The last thing we ever want is for a user to contact us and tell us that our application is giving them an error or just blowing up. Errors are the first line of defense for finding application problems. Developers are responsible for finding and fixing errors.



Ideally, dev teams will find an error before our customers call to complain because odds are most of them won't even call to tell you. They will just go somewhere else.

Excellent [error tracking](#), reporting, and alerting are absolutely critical to developers in an application performance management system. We would highly recommend setting up alerts for new exceptions as well as for monitoring overall error rates.

Anytime you do a new deployment to production, you should be watching your error dashboards to see if any new problems have arisen. Odds are, you will find some type of new errors that you can then quickly identify and hotfix.

Bonus Feature: View Logs & Errors in code profiling traces

At the heart of Retrace is a powerful code profiler. It tracks specific methods in your .NET, Java, PHP, Python, Ruby and Node.js applications to help understand the performance and behavior of your code. Retrace provides some of the most detailed code profiling traces of any APM solution you can find.

One of the most powerful features you can get in an APM tool is the combination of your logs with detailed code profiling. Your logs can provide a great deal of context to what happened within a web request or transaction.

6. Metrics

Application framework metrics

Server metrics like CPU and memory are interesting, but for developers, application metrics can be a lot more valuable for true application performance monitoring. Developers need to monitor metrics around things like [garbage collection](#), request queuing, transaction volumes, page load times, and much more.

Developers can monitor a wide variety of [Windows Performance Counters](#) and [JMX](#) MBeans. It can also be critical to monitor things like Redis, Elasticsearch, SQL, and other services for key metrics.

Custom applications metrics created by the dev team or business

Standard server and [application metrics](#) can be very helpful for monitoring your applications. However, you may get way more value by creating and monitoring your own custom metrics. At Netreo we use them to do things like monitor how many log messages per minute are being uploaded to us, or how long it takes to process a message off of a queue. These types of custom metrics are easy to create and can be very useful for [application performance monitoring](#).

Bonus Feature: App Scoring

Are we better or worse off with this release? We heard these concerns from our clients and wanted to provide a single metric to answer this common question.

[App Scoring](#) is a proprietary metric that expands on Retrace's deep performance insights, combining many factors of an application's performance into a single "letter grade" benchmark score. Users of Retrace can now see at a glance how their application is performing over time.

App Score Detail x

C **nopCommerce (Prod Sandbox)**

Your App has scored average overall. There are different actions you can take to improve your score.

Here is the breakdown of your App Score:

A **New / Regressed Errors**
 This score is a measure of how many new and regressed errors have occurred in your App recently. To improve this score, find and fix issues earlier in the development phase via unit and integration testing. [Show Me](#)

A **Satisfaction**
 This score is the measure of satisfaction of user requests. You can see much more information about Satisfaction on your App Dashboard. To improve this score, you need to get your pages to load faster and reduce page failures. [Show Me](#)

A **App CPU**
 This score is a measure of how well your App is performing on a server. To improve this score, ensure you write efficient code and make sure your App is correctly scaled. [Show Me](#)

F **Errors / Request**
 This score is a measure of total errors over the number of requests. To improve this score, reduce the total number of failures and exceptions that are occurring in your application. [Show Me](#)

App Score Summary

ExampleScheduledTask Test C	DefaultAppPool Sandbox A
Sandbox Ruby Sandbox A	Node Express Showcase Sandbox B
Sandbox Python Sandbox A	Sandbox PHP CLI Sandbox B

7. Deployment tracking

Deployment tracking gives you the ability to see when deployments happen, what environment they happened in, and how they affected your application's performance. It provides visual indicators on your timelines within your APM when these events took place, and you can easily use them to drill down into metrics that will give you an exact idea of what is happening in your application. These metrics can lead to quicker troubleshooting, or just give you the proof that some things aren't always the developer's fault!

Deployment tracking is a necessity since nothing unites or divides a team like a deployment straight out of your nightmares.

8. Real User Monitoring

Real User Monitoring ensures that you have end-to-end monitoring. By stitching front end and back end code together, you can see the entire picture of what's happening with your application both on the server and on the client side. This ensures decreased load times so your users remain happy.

Real User Monitoring also helps your developers pinpoint exactly where to focus for reduced time to resolution. But how?

Retrace's Resource breakdown graphs can quickly help you identify if your images need to be optimized or if your stylesheets and scripts need to be minified or cached. Developers can also use the segmentation information within Retrace to monitor load times based on browser, geography, and device type. This makes it easy to identify opportunities to improve the overall experience by pinpointing optimal locations for Content Delivery Networks (CDNs).

Top APM Tools & Solutions

Retrace

[Retrace](#) is an [affordable SaaS APM tool](#) designed with developers in mind. It enables granular insights through detailed code-level transaction traces for easy troubleshooting. Retrace also combines errors and logs into a single trace view unlike any other APM tool on the market.

With Retrace in non-production and QA in addition to production environments, users are often able to resolve issues proactively before they reach production.

In addition to Retrace, Stackify by Netreo also offers a free code profiling and tracing tool, [Prefix](#), for developers to use on their workstations to write better code before committing it.

- Languages: .NET, Java, Python, Ruby, Node.js, PHP
- Unlimited users for full team collaboration
- SaaS based
- Integrated error and log management
- Detailed code-level transaction traces
- Includes application metrics, server monitoring, and real user monitoring
- Easy to install and use

Cost: Starts at \$35/mo. See our [pricing page](#).

APM is Affordable For All Dev Teams

Traditionally, application performance management tools have been an expensive luxury item that only large IT enterprises could afford. Many APM vendors still cater to the larger enterprises, still charging \$2,000-\$4,000 per year per server. Ouch!

Most APM solutions are very complex to configure and use. So much so that development teams don't even use them. They end up being expensive traffic lights and dashboards. Some vendors have put a huge focus on [making their products affordable](#) and very easy to use so they can be available to development and operations teams of all sizes. Our product, [Retrace](#), starts at just \$99 a month.

Advice on Choosing an APM Tool by Role

Developers/architects

[Cristian Vanti](#) is a performance-oriented solution architect with over 20 years as an IT professional in several different roles. He's passionate about bleeding-edge technologies, fast paced environments, and challenging projects.

NOTE: The following information is excerpted from [Choosing the Right APM – A Fool with a Tool is Still a Fool](#) via LinkedIn.

“When a company decides to buy a tool, it must create value, satisfy specific needs, and ultimately solve problems...”

What still surprises me is that the performance culture isn't yet widespread, and often managers buy software or services that are very appealing or trendy, but aren't actually an element of any strategy.

Web performance is a war that must be fought every day. Every day customers ask for new features and expect quicker systems. You can't think that a tool like Application Performance Management is a magic wand that can solve all your problems forever. First comes the strategy, then the budget, and then, only then, you can look at the market to choose your tools. This is a process we often help our customers to understand.

Operations

[Karun Subramanian](#) is passionate about IT operations. His website is dedicated to supplying useful information and tools to effectively manage your Linux, DevOps, and APM environments.

NOTE: The following information is excerpted from [APM Selection Guide: How to choose the right Application Performance Management System](#) via [KarunSubramanian.com](#).

“Do NOT begin the search of an APM solution unless you have clearly defined the requirements...”

You may be thinking, “This is not a software application that we are developing. This is a monitoring tool! What do you mean by requirements?” Well, consider the following questions:

- Do you need deep insights such as code-level diagnostics?
- What are the various types of technologies you need to monitor? PHP? Ruby? Java? Node.js? Python? .NET? Mainframe?
- Do you need end-to-end visualization with end-user experience monitoring?
- Do you need to build custom dashboards for your IT Operations folks to use?
- Do you need SAAS (software as a service) solution?

About Stackify by Netreo

We built a set of APM tools to tell us how, and why, applications fail. From the workstation and pre-production to deployment, when our 1300+ customers spend less time fighting technology they spend more time releasing it, and those new applications make the world a better place for all of us.

APM Questions? Email us at info@netreo.com. We'd love to hear from you.

Learn more about [Retrace](#) and start your Free trial today!

Discover the power of our Free code profiler [Prefix](#) today!