















The logo for OverOps, featuring the word "OverOps" in a white, italicized sans-serif font. The letter "O" is stylized with a blue-to-white gradient and a slight shadow effect.

OverOps

OverOps and APM Comparison



| Function | OverOps | Dynatrace | Additional Comments |
|---|---|--|--|
| Detect and categorize code errors as New, Resurfaced, or Critical such as unhandled? |  |  | OverOps sees the source code that is running, and detects when an exception is new or resurfaced, which gives valuable context to devs. This is evident when sending proactive alerts based on new exceptions, before users call in, and auto-route to devs to prevent user impact. |
| Deduplicate and aggregate unique code errors for faster tracking? |  |  | OverOps deduplicates code errors with counts and failure rates to measure impact to users. DT does not deduplicate errors down to unique code location. |
| Detect changes in code compared to previous release? |  |  | With code changes come inherent risk of impact to users. OverOps gives indications code has changed so devs know where code changes may have caused issues. |
| Automatically show all the code that ran for the full stack at time of error? |  |  | OverOps decompiles the entire stack trace automatically and overlays the stack with all variable data in one view, so devs can reproduce quickly to move on to fixing code and developing new features. DT users follow a manual process to download and decompile class files of a stack individually to view code that ran, without showing the lines of offending code or variable data in the same view. |
| Show all data (variables) that was processed at time of error for the entire stack trace, with no configuration or foresite required? |  |  | OverOps captures all variable data out of the box for the entire stack trace. Method args and params can be configured in DT by choosing specific methods settings as seen here . This requires foresight and instrumentation. |
| Automate code quality gates for code promotion in pre-production environments? |  |  | With tools like Jenkins, Bamboo, etc, OverOps can enforce code quality gates for QA and Dev teams before promoting code to production. For instance, stop a build when new errors occur during testing, or increasing errors, and other critical code errors. |
| Integration to common tools to automate troubleshooting of critical errors? |  |  | OverOps surfaces data in existing tools where dev teams spend their time, like APMs, logging frameworks, ticketing & incident systems, ChatOps tools, CI/CD, etc to fully automate existing troubleshooting process with minimal training. |

OverOps



Thank you