OverOps

Identify and Prevent Errors Missed by Testing and Static Analysis

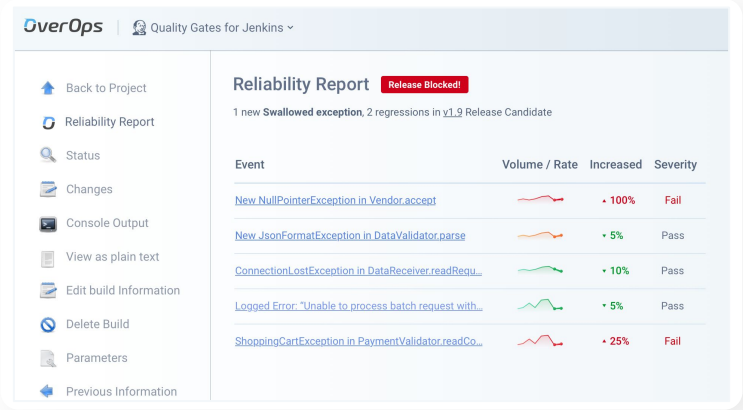
OverOps adds runtime quality gates to your build pipeline, blocking critical issues before customers are affected

Static Analysis
scans your source code for bad coding practices

Automated Tests
tell you when predictable issues occur

Runtime Code Analysis
tells you when and why code breaks **at runtime** - no foresight required

OverOps analyzes code as it executes to identify critical errors and resolve them in minutes - even if they weren't tested for or logged



Identify
critical errors that were not tested for

Prevent
bad quality code from being promoted

Resolve
issues with complete variable state

Integrates with

sonarqube · Bamboo · TC · Jenkins · And many more...

OverOps

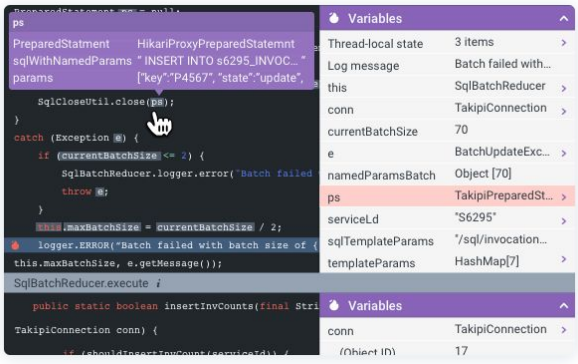
Ensure Code Changes Don't Impact Customer Experience

Introducing Runtime Quality Gates:

- New Errors
- Slowdowns
- Critical Exception Types
- Resurfaced Issues
- Increasing Error Rates
- Total & Unique Error Volume

Block Unreliable Releases

- **Identify** critical errors and slowdowns in every release
- **Report** on prioritized issues and their root cause
- **Block** unreliable releases from being deployed to production



Know Why Code Breaks

- **Exact state of the code** across the entire call stack
- **Code variables and objects** - 10 levels deep into the heap
- **Environment state** - CPU and memory state, thread state, and environment variables

See how OverOps stops bad code from being promoted: **www.overops.com/product/prevent**