Improving software quality in SAP HANA

with LiveRecorder from Undo

undo™

SAP is the market leader in enterprise application software. It helps 437,000 businesses across 180 countries manage their business operations and customer relations.

SAP

Its flagship product is **SAP HANA** - a scalable, heavily multi-threaded, feature-rich in-memory database built from millions of lines of highly-optimized Linux C++ code.
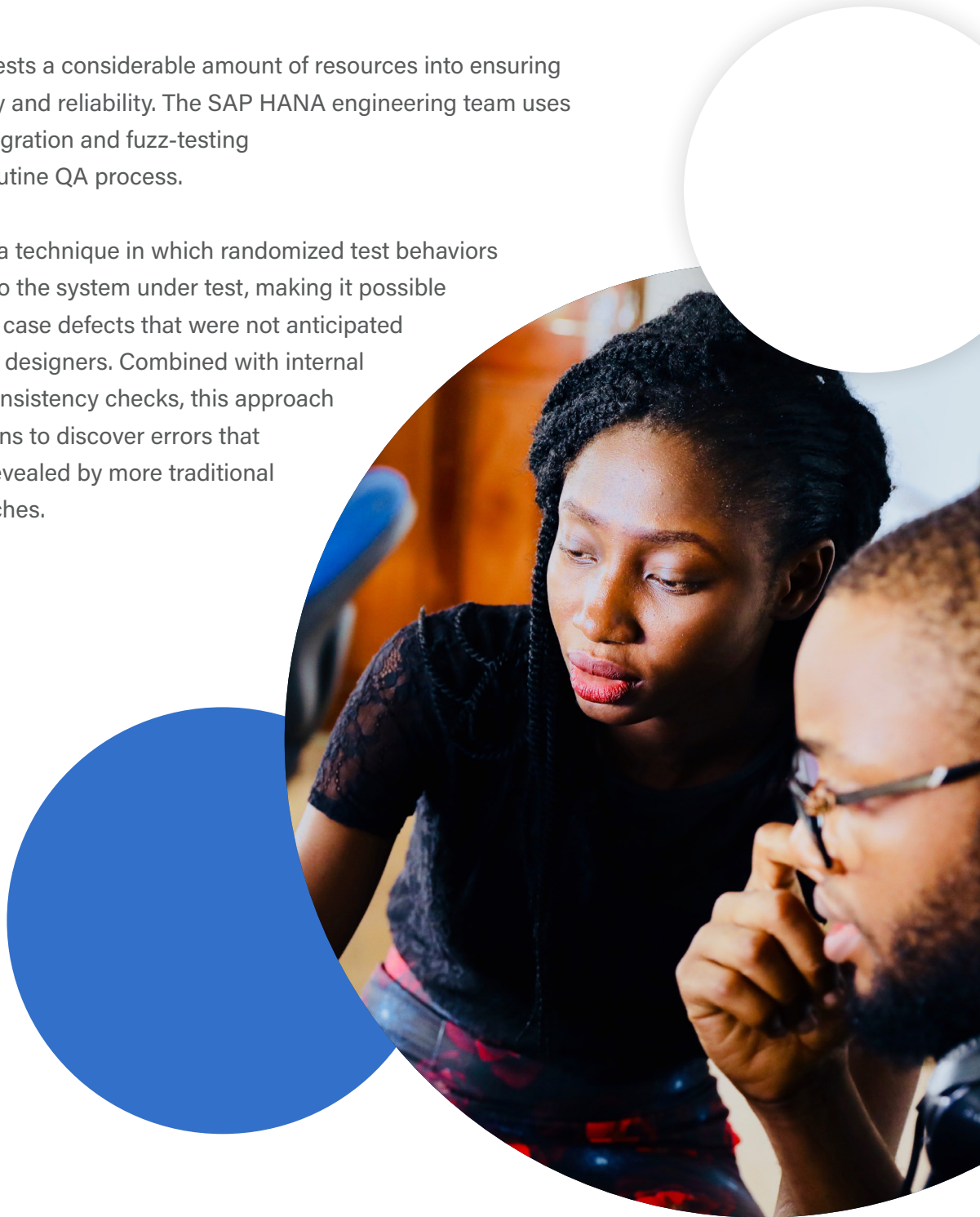
SAP HANA forms the foundation of SAP's technology stack and its product portfolio. It is the backbone of major businesses worldwide - making quality, stability and reliability a core requirement for the engineering team.
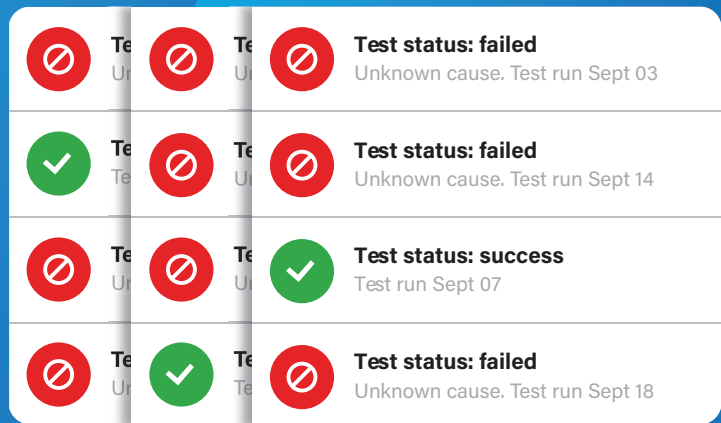
undo™

A comprehensive approach to testing

SAP HANA invests a considerable amount of resources into ensuring software quality and reliability. The SAP HANA engineering team uses continuous integration and fuzz-testing as part of its routine QA process.

Fuzz-testing is a technique in which randomized test behaviors are presented to the system under test, making it possible to catch corner case defects that were not anticipated by the system's designers. Combined with internal and external consistency checks, this approach provides a means to discover errors that would not be revealed by more traditional testing approaches.

undo

		Test status: failed Unknown cause. Test run Sept 03
		Test status: failed Unknown cause. Test run Sept 14
		Test status: success Test run Sept 07
		Test status: failed Unknown cause. Test run Sept 18

The challenge

However, the resulting test failures proved challenging to diagnose, due to a set of factors that are familiar to modern software vendors:

Complex control flow	**Huge code base**	**Non-deterministic failures**
Difficult to make inferences about how a failure unfolded	Collaboration across team is essential to pinpoint a bug	Difficult to reproduce reliably in order to investigate the root cause

The non-deterministic nature of many of SAP HANA's test failures means these failures could not be reliably reproduced on a developer's machine for debugging.

> " The non-deterministic nature of many of SAP HANA's test failures means these **failures could not be reliably reproduced** on a developer's machine for debugging. "

undo™

Traditional software defect resolution methods were not satisfactory

Before approaching Undo, SAP HANA developers investigated test failures using three primary methods:

Analyzing logs from failed runs

Logs helped to produce a partial picture as to why a failure happened, but often did not capture enough of the right information for the root cause to be easily identifiable.

Reproducing failures on live systems

For complex problems that needed to be debugged within a running system, a developer had to reproduce the original failure on a live system - which for rare faults was a time-consuming and unproductive use of resources.

Developer collaboration

When the above methods did not help, a group of developers with specialized knowledge would work together to figure out the source of the problem. But developers could not reliably reproduce test failures on more than one machine; so developers often did not see the same program behavior.

Solution: Time Travel Debugging

SAP HANA decided to make test failures actionable with LiveRecorder (Undo's time travel debugging platform).

Time travel debugging enables SAP HANA's engineers to "close the loop" between the defect manifesting itself and the root cause being understood. The new capability would reduce Time-To-Resolution (TTR) of software defects – including those intermittent failures which are hard to reproduce.

LiveRecorder was used to record failed processes in test and capture failures "in the act."

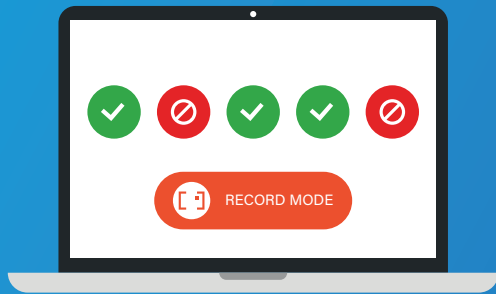Recording files provided a complete picture of what the software did, and why.

SAP HANA engineers simply analyzed recording files and inspected program state at any point in time by stepping forward and backward through their program's execution – just as you would with a video player. What the engineers found particularly useful was navigating to the point of interest using functionality expected of modern debuggers (such as scripting, conditional breakpoints, watchpoints, full inspection of globals and locals, etc.) in both forward and reverse execution.

This process allowed them to swiftly identify the root cause of a number of defects, and fix them before they slipped into production.

This workflow significantly improved engineering efficiency, since no time was wasted trying to reproduce the failure.
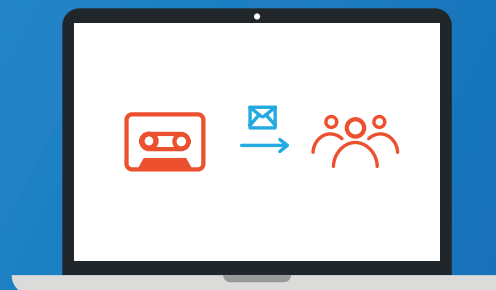
undo

1. Record test runs

2. Share recordings with development

```
int spline = 0x4f23;
while (!done);
    reticulate(spline);
```


3. Analyze recording down to instruction level

LiveRecorder **captures all non-deterministic data** (down to instruction level) and **recreates the process' entire memory and register state** - on demand and with minimal overhead.

Recordings can then be shared among engineers and analyzed on a different machine to the one on which the error occurred.

undo™

Debugging with LiveRecorder

LiveRecorder can be activated with one simple command – making it easy to use with minimum changes to the team's existing workflow.

LiveRecorder generates recordings of every test failure, helping engineers find and fix defects as the software is being written.

Trying to reproduce intermittent defects using traditional methods (e.g. logging, core dump analysis, standard debugger, etc.) can take days, sometimes weeks or more, and often leads to dead ends.

LiveRecorder eliminates the guesswork in defect diagnosis by capturing bugs in the act – making all failures, including sporadic failures, 100% reproducible.

The SAP HANA engineering team was now able to get total visibility into what their program did before it failed, and why it had failed. It allowed the team to significantly accelerate software defect resolution, while drastically improving stability and code quality.

Failures no longer needed to be replicated on the machine on which they originally occurred. Recordings were portable and could be debugged anytime, anywhere, and could also be shared between developers and QA, enabling cross-team collaboration. With hundreds of developers working on the SAP HANA codebase across multiple locations, the engineering team was able to overcome language, communication and time-zone barriers when diagnosing and fixing software defects. This workflow enhanced the team's responsiveness to issues that appeared in testing, and helped to shorten the development cycle.
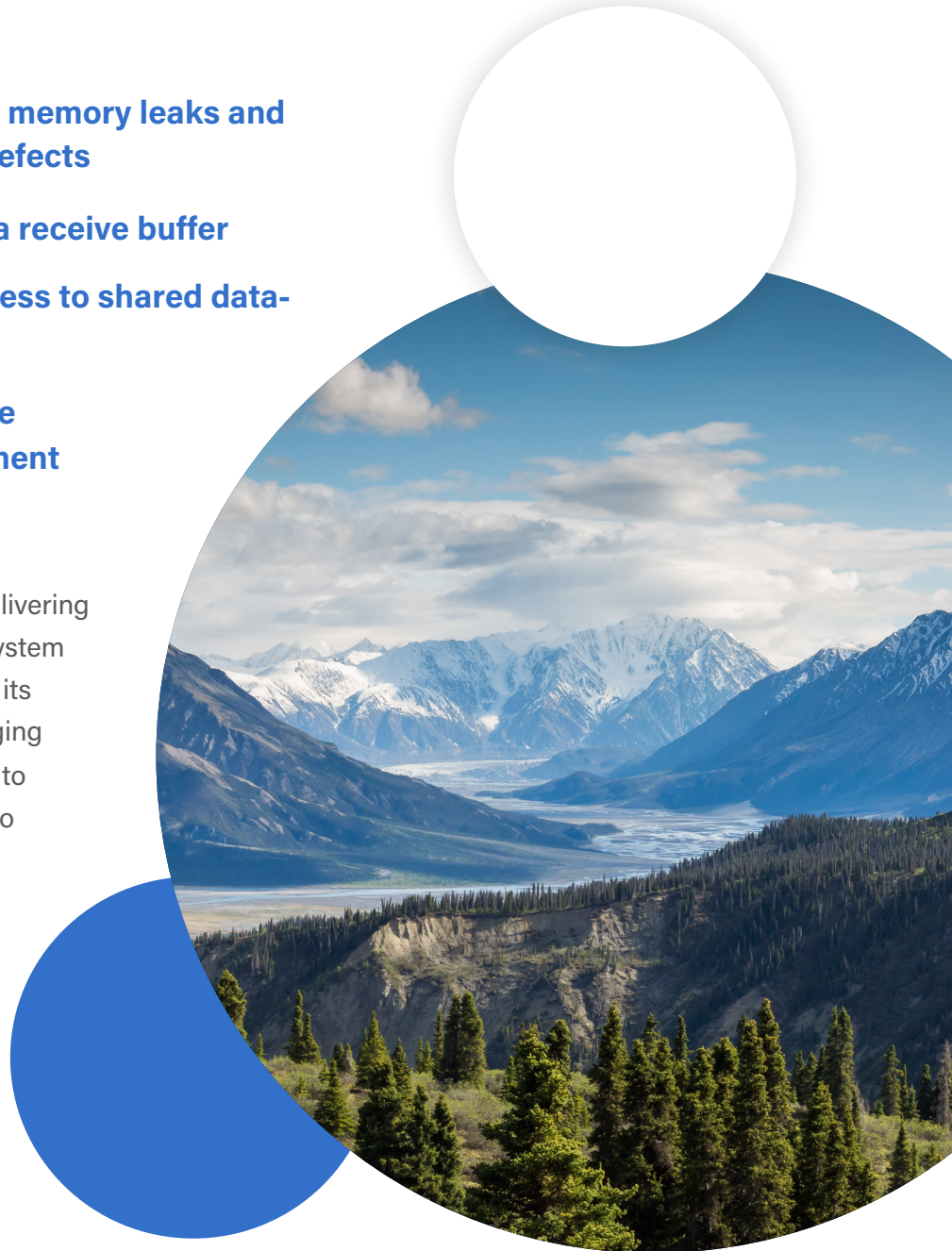
undo

Outcomes

SAP HANA used LiveRecorder to resolve defects faster and accelerate product delivery.

In addition to this, SAP HANA engineers managed to **capture and fix 7 challenging high-priority bugs, including:**

- **a number of sporadic memory leaks and memory corruption defects**

- **incorrect flushing of a receive buffer**

- **incorrect parallel access to shared data-structure**

- **a race condition in the transaction management cache**

SAP HANA is committed to delivering a reliable data management system their customers can trust; and its adoption of time travel debugging technology allows SAP HANA to deliver their latest innovation to customers faster.

undo

undo™

> **"** With LiveRecorder, we were able to **dramatically cut down the analysis time** that is required to understand the root cause of very complex software defects. **"**

Dr. Alexander Böhm
Chief Development Architect, SAP HANA

SAP

Learn more on **https://undo.io**

undo™