*instinctools

Quality Assurance Audit For a Fintech Company

How a thorough QA audit and strategic fixes allowed a Swedish Fintech company to gain full visibility into their software at every stage of the SDLC, speed up new feature releases by **26%**, lower customer support expenditures by **33%**, and increase CSAT by **16%**.

Industry:
Fintech

QUALITY ASSURANCE
SAAS DEVELOPMENT

Business challenge

When underestimated from the start, QA issues bite back in the later stages. Our client learned this the hard way. A Swedish Fintech company, providing SaaS software for expense management, has prioritized feature development, putting QA on the back burner.

However, as their products gained traction, cracks began to reveal themselves:

Inadequate QA processes

allowed bugs to slip into their customers' software environments

Given how diverse their clients' IT ecosystems can be, each support request required a customized solution. The number of tickets was mounting, going beyond the customer support team capacity.

The lack of test automation

delayed releases

Sticking to a quarterly release plan was already off the table, and new features were rolled out once a year at best.

Although the products kept operating *somehow*, the quality assurance process was chaotic and unmanageable, putting the client's reputation at stake.

In the face of the QA turmoil, the client was at a crossroads: rush to hire a quick fix or pinpoint weak spots in their existing framework and rebuild it with expert guidance. While the former option would put out the immediate fires, it wouldn't align with the client's determination to have flawless software in the future. So they opted for the other route, seeking *instinctools' expertise to establish a robust and sustainable QA framework.

Solution

The client engaged our QA engineers to run an all-encompassing quality assurance audit of their expense management software, identify problem areas, and suggest improvements.

01 In-depth audit of the client's QA processes

Instinctools' QA specialists:

- Held a **pre-audit briefing** with the client's in-house team to see how the absence of QA expertise affected the software
- Defined the **audit scope and objectives** to keep all stakeholders in the loop
- Examined the client's **QA and project management tools**, such as Jira and Confluence, to assess existing workflows and testing methodologies
- Revised available **documentation** and performed **code review**
- Analyzed the **findings** to prepare spot-on, actionable recommendations

The audit revealed three fundamental hurdles to address:

01 **Disorganized software testing management.** The expense management software had its dedicated Jira space, but QA tasks were unstructured, and the overall testing lifecycle remained unclear, making it almost impossible to track test objects or related tasks.

02 **Inconsistent documentation.** Test cases were recorded arbitrarily, leading to an opaque and inefficient QA workflow.

03 **Reliance on outdated tools.** The client used legacy software without questioning its effectiveness.

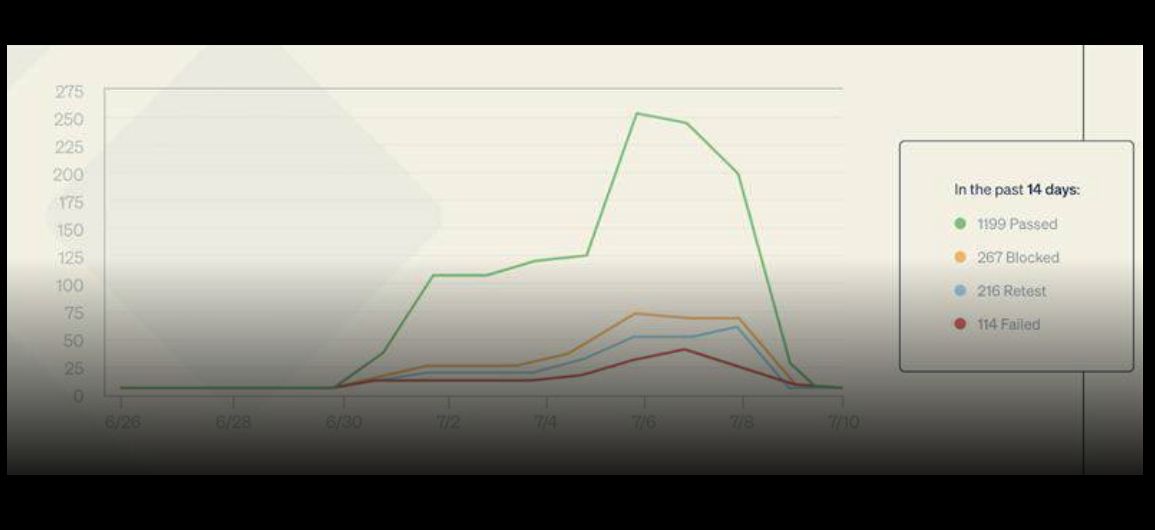
Besides identifying the problems, we mapped out solutions in three key areas:

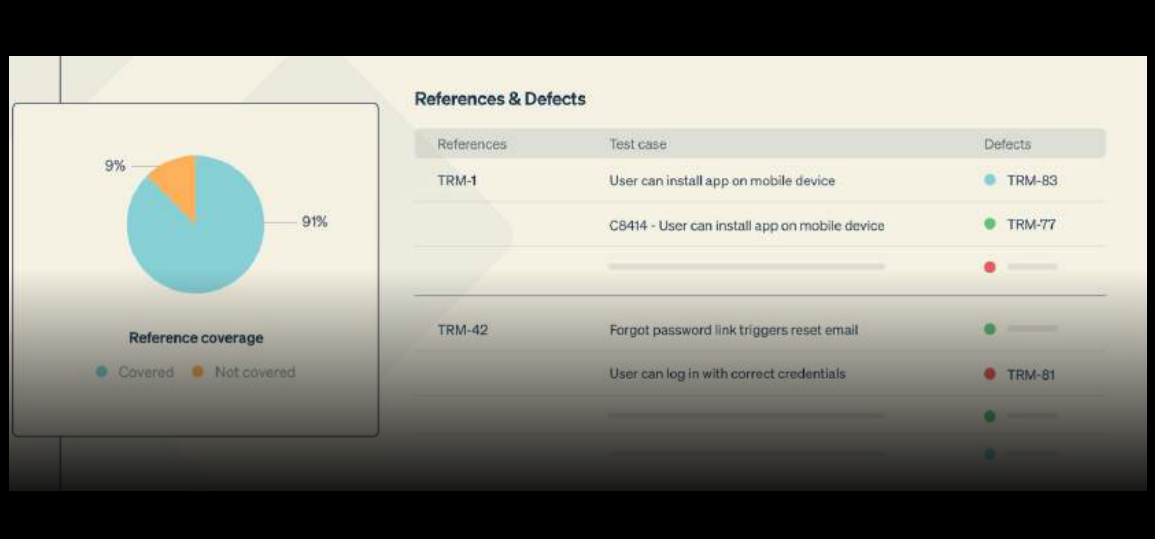
Quality assurance	Project management	Business analysis
Standardizing test management in a centralized QA-specific system and adding a post-testing check-up to verify the accuracy of test results.	Integrating the QA platform with Jira to ensure complete visibility into testing stages at all times.	Setting documentation standards for stress-free accreditation for any certification.

Satisfied with the audit's scope and depth, the client decided to move forward with *instinctools to *act on* these changes.

02 QA management platform implementation

To bring structure and visibility to QA operations, we explored multiple options and landed on **TestRail**, which offers the most budget-friendly pricing for small QA teams with up to five users.

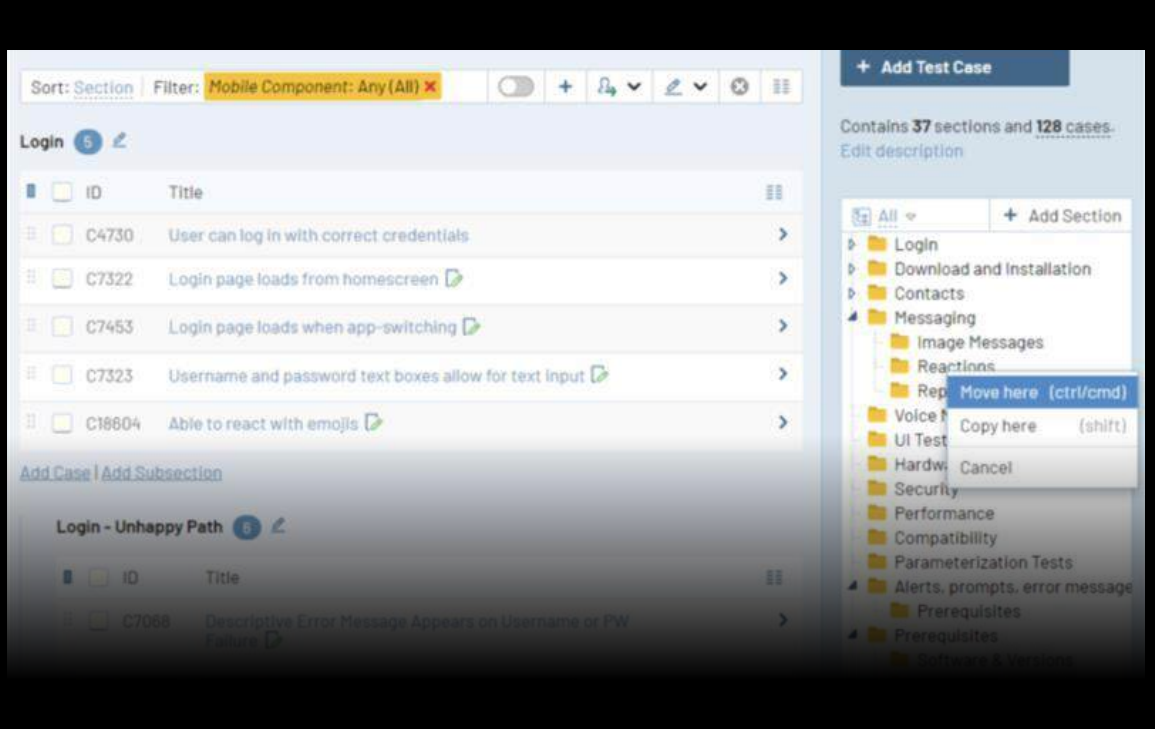




References	Test case	Defects
TRM-1	User can install app on mobile device	TRM-83
	C8414 - User can install app on mobile device	TRM-77
TRM-42	Forgot password link triggers reset email	
	User can log in with correct credentials	TRM-81

With **intuitive test plan management and deep analytics,** TestRail gave engineers real-time insight into patches and releases, helping them ship high-quality code faster.

We also integrated TestRail with the client's Jira. This way, users can link test cases to tasks, stories, and epics to clearly indicate what is being tested. Other product team members, like project managers or business analysts, can access test plans and test cases from Jira tickets with one click.



ID	Title
C4730	User can log in with correct credentials
C7322	Login page loads from homescreen
C7453	Login page loads when app-switching
C7323	Username and password text boxes allow for text input
C18604	Able to react with emojis

03 Stress testing upgrade

The client's stress-testing tool was outdated and tricky to upkeep, making its ongoing use impractical. We suggested switching to a **Hyper and Apache JMeter** combo. The first is an advanced terminal emulator used as an interface to run JMeter. However, the terminal's functionality extends beyond stress testing. Thanks to automation and additional plugins, it streamlines multiple everyday QA tasks, such as log parsing, cleaning up environments, and debugging.

04 Performance testing with 99,99999% reliability of test results

Smooth performance testing with load, stress, spike, and soak tests has always been at the top of the client's agenda, and they wanted us to put a premium on it.

First, we ran regular tests, measuring solutions' responses to unusual, sudden, and extreme loads over different periods. However, as the client's software implies corporate card management, invoice processing, and other financial operations, they need to be more than 99% certain of the reliability of the test results.

That's why we added a *check-up step* when the QA team analyzes performance test data and confirms that the **outcomes are 99,99999% reliable.**

05 Robust workflows for other mission-critical tests

We proceeded with writing missing templates for test plans, cases, scripts, and reports in TestRail for usability, regression, and new feature testing. All activities were carried out in parallel. Our QA engineers performed:

- **New feature testing** within release management
- **Regression testing** after every hotfix, patch, or release
- **Usability testing** alongside every regression and new feature test

06 API and UI test automation

Following the classic pyramid of testing, we moved on to API and UI tests.

Our QA engineers worked in three directions:

- Revising old tests created by the client's previous vendor and revamping them if possible
- Writing missing auto tests for old APIs
- Designing tests for new APIs

During periods when no new APIs are being written, test coverage reaches 100%.

Speaking of **automating UI tests**, we've already covered 55% and keep working on it.

07 Expanding our testing coverage

With every new iteration of the change management plan, we are increasing our efforts in other types of testing to ensure even better software quality. For instance, along with further UI test automation, **accessibility, durability, and penetration testing are in the project's backlog.**

Before

- Chaotic, opaque, and unpredictable QA process
- Bugs constantly slipping into the customers' environments
- Annual releases at best
- Unverified performance tests' results
- Up to 90% of customer support requests required input from senior-level specialists

After

- Centralized, transparent, structured, and easy-to-manage QA process
- Bugs are uncovered and fixed in the early development stages
- Quarterly releases
- 99,99999% accuracy in performance testing
- Junior engineers resolve up to 75% of customer support queries

Business value

- **Fully transparent** and traceable software testing pipeline
- **20% faster** test case creation
- **x1.5 faster** release cycles
- **26% reduction** in time-to-market for new features
- **33% drop** in customer support costs
- **86%** CSAT

Client's testimonial

Here's how the client's Product Owner describes *instinctools' QA experts' contribution to their product's uplift:

> *Instinctools' input was truly transformative for our product. We were impressed by their meticulous audit and proactive, holistic, and incremental approach to dealing with our QA bottlenecks.*

Multiplier effect

It's good if your developers follow the basic playbook by writing clean, modular code, conducting regular code reviews, or even building simple API and unit tests themselves. Good, but not enough. The lack of well-honed QA processes is a ticking time bomb — the cracks might not show early on, but they will emerge when you decide to scale your solution.

The good news is that even if QA-related issues snowball into a costly threat, there's still room for change as long as you have a stalwart tech partner by your side. Opt for an all-encompassing QA audit to forge the path toward transparent, structured, and easy-to-manage QA processes.

Do you have a **similar project idea**?

instinctools.com
contact@instinctools.com