deviQA

10+	1M+	100+	300+
YEARS OF EXPERIENCE	HOURS WORKED	ENGINEERS	HAPPY CLIENTS

Case Study: Compass

Web Testing, Mobile Testing, Automated Testing, Dedicated QA Team

Project Overview

What Compass had in the testing processes when they came to us and what they got after they started working with us.

Before improvement

- ☒ Didn't have a formal QA process
- ☒ Testing was carried out by the devs
- ☒ Negative quality of tests
- ☒ Didn't make automated testing
- ✓ Project had a well-detailed specification

After improvement

- ✓ Set up a formal QA process
- ✓ Developed BDD scenarios based on a detailed specification
- ✓ Full coverage of the application by test cases
- ✓ UI automated tests for web application
- ✓ Automation smoke test run before each deployment to check main functionality
- ✓ Automated API tests
- ✓ Detailed Automatically Reports Generation which were sent to the customer
- ✓ UI automated tests for iOS and Android
- ✓ Parallel autotests run on all supported browsers

6000+	>40%	1500+	12000+	6000+
TEST CASES CREATED	TESTS CASES AUTOMATED	PASSED TEST RUNS	BUGS REPORTED	TICKETS WITH MAJOR, CRITICAL AND BLOCKER STATUSES

QA Team:
2 Automation and 1 Full-stack Engineers

Project length:
4+ years

TECHNOLOGIES & TOOLS

- ✓ Android
- ✓ iOS
- ✓ Ruby
- ✓ Cucumber
- ✓ Gherkin
- ✓ BDD
- ✓ Capybara
- ✓ Selenium
- ✓ Jira
- ✓ TravisCI
- ✓ Git
- ✓ Bash Script
- ✓ TestRail
- ✓ Android Studio
- ✓ XCode

The Challenge

Contactually is a real-estate tool that turns business relationships into results. It is a web-based customer relationship management tool that allows companies to oversee and manage communication activities using an easy-to-use and well-built interface.

Before we joined the project, the team did not have a formal Quality Assurance process. Testing was carried out by the developers, which negatively affected the quality. Moreover, the team didn't make automated testing either for Web or for Mobile. Although the project was at the very initial stage, we have already clearly seen problems that could become critical if they were not eliminated in the early stages.

The good news was that the project had a well-detailed specification. We worked closely with the developers to help them to avoid potential bugs, set up the formal testing process, and do it all in the shortest possible terms because the customer's budget and time frames for the beta was pretty limited.

Due to the fact that DeviQA engineers have experience in similar projects, we were able to do even more than expected.

Achievements

In the early stages of the project, DeviQA team consisted of 1 one full-stack (manual/automation) and one automated QA engineer. The customer allowed us to make any changes in the processes to ensure the quality of the product. So, from now on, the responsibility for the quality is on our shoulders.

As a first thing, we started covering the entire application with test cases using BDD scenarios based on a detailed specification. We made it for several important reasons, namely:

1. Because we used BDD scenarios, developers quickly read them before writing code and paid attention to both positive and negative scenarios, which subsequently reduced the number of potential bugs.
2. Since the scenarios were easily readable, we could use them as test cases in the early stages of the project when we did not have enough autotest coverage, which allowed us to reduce the time for writing additional test documentation.
3. Full coverage of the application with the test scripts at times allowed us to increase the speed of writing autotests since half of the work has already been done.
4. This allowed any non-tech team member to add the desired test cases written in a language understandable by the whole team.

Based on our experience, we chose TestRail as a test case management system. It was done to provide detailed reports to the customer and to have detailed analytics on the number of working and non-working scenarios. The customer was pleased with the results and added one more automation QA from our side.

In the shortest possible time, we made well detailed BDD test cases for smoke, covered them with automated tests, which allowed us to test the basic functionality for ~ 6 mins.

We set up the automatic reports in the TestRail and made autotest support for all main browsers using Browserstack. We run them in parallel(multiple browser instances to speed up the testing process). All automated tests were integrated into the Pipeline process on TravisCI. After each test runs, there was an automatically generated report which was sent to the customer.

Having enough resources, we continued to work on creating additional BDD scenarios for regression. DeviQA created ~ 6k of the well-structured BDD scenarios divided by the 148 sections for regression testing. Because of the right grouping, we always had our scenarios up to date. About 40% of the scenarios for regression testing were Automated.

We run autotests for a smoke before each deploy, and because some of them were built via API, it took ~ 6 mins to run all tests.

Besides, during the manual testing, we have noticed that a good part of the bugs was actual on mobile devices only. We've adapted all main UI automated tests for mobile and run them on the real devices using the Device Farm.

Services Provided

Quick processing of all fixed tickets and close cooperation with the automation team allowed us to avoid major/critical issues on production. Passed hundreds of test runs during regression & smoke testing.

Web Testing →

DeviQA team provided mobile testing in parallel with web testing as a good amount of the issues were found on mobile devices. QA checked the app with a range of old and new devices with different versions of OS. Each crash was reported with the logs. Each bug had detailed steps and attachments for quick replication, so additional questions about issues from devs are infrequent events.

Mobile Testing →

2 Automation QA and 1 Full-stack covered the app with autotests and updated them frequently. Autotest results always helped to find bugs before/after deploys, so new issues were found and reported very quickly. Good test coverage of the main functionality eliminates unexpected issues in production. Parallel autotests run on all supported browsers allowed to exclude bugs related to the specific web environment.

Automated Testing →

The dedicated team is the best way to keep your app well tested, maintain communication with the development team fast, and be calm about provided QA services. Three dedicated senior QA engineers were able to build smooth and well-working processes.

Dedicated QA Team →

Let us know your details
so we can get back to you for discussion!

Telephone:	Skype:	Email:
+1 805 491 9331 +44 1922 214429	deviqa_co	info@deviqa.com

https://www.deviqa.com/

1. **Free proof of concept**
We will conduct a free proof of concept and prove that you can trust our quality assurance services
2. **12 hours to start**
Within 12 hours, our team is ready to start your project
3. **Quick team resize**
The size of the QA team on your project could be resized in no time
4. **Daily progress reports**
We send an email report with detailed statistics and progress on daily basis