

# Benefits of integrating CrowdTesting and Test Automation methodologies into the software delivery cycle

USE CASE OF ONE OF THE TOP 5 ITALIAN BANKS



# ABSTRACT

This Use Case reports the synthesis of AppQuality and Quence's experience in supporting **one of the main Italian banks** in achieving its **goals of efficiency, effectiveness and innovation of its life cycle and delivery of software** and digital services, both for mobile and web.

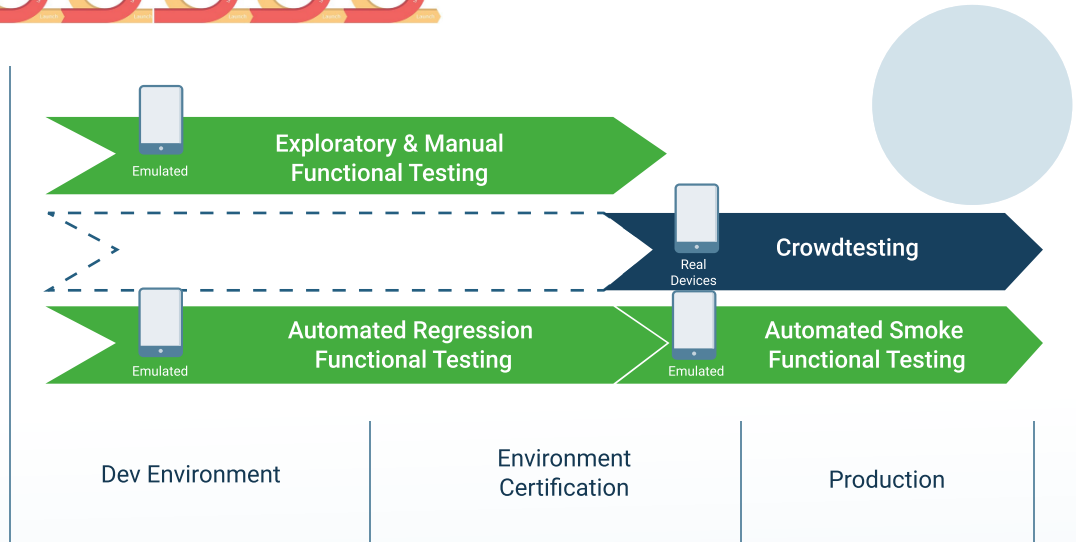
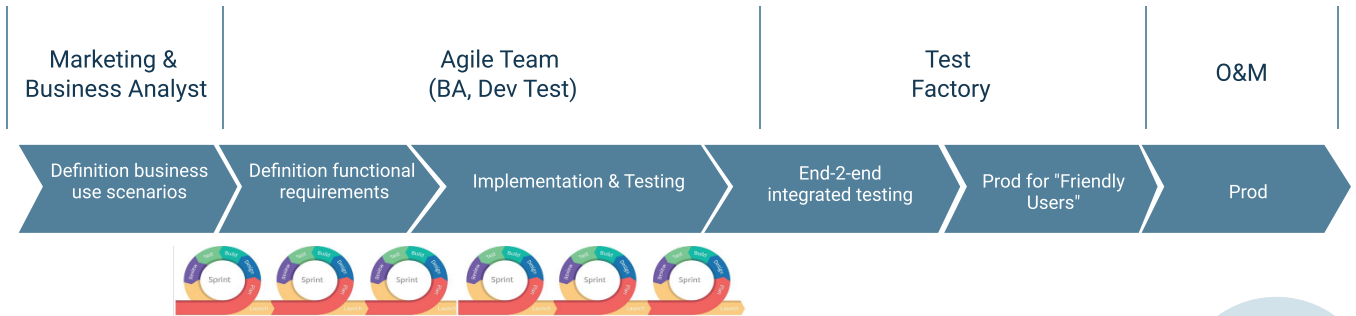
The introduction and integration of different methodologies, techniques and technologies proved to be an **added value both from a technological and a cultural point of view**. The approach did not only lead to the achievement of the client's technical and business objectives. This path has also resulted in a **greater propensity of testing teams towards process innovation** and the need for change and flexibility required by the market.

The model described is **permanently part of the bank's software life cycle** and is applied to all projects.

In particular, it has been successfully applied to different strategic projects, such as:

- **Digital Bank:** Internet Banking and Trading services provided both from PCs and mobile devices;
- **PSD2:** Open Banking platform and services as required by PSD2 regulations.

# Processes, infrastructures, customer delivery life cycle



## Exploratory & Manual Functional Testing

**During the cycles/sprints**, in order to provide quick feedback regarding the implemented functionalities, the **"Exploratory Testing" approach is used**, which is more suitable than the traditional approach based on defining Test Cases. This approach allows the tester:

- to **have greater speed in providing feedback on operation**;
- to provide **qualitative assessments** on aspects, such as usability, reliability, consistency, clarity, stability, performance, responsive-design characteristics of the tested object.

This approach is very similar to the one used later, as we shall see, in the Crowdfunding phases. The traditional approach of defining test scenarios, or test cases, based on the specification of test steps, input data and expected results, is then used to implement the **Risk-Based-Testing approach**. The latter serves to provide a method of assessing the business risk associated with a release against the progress of development and QA activities. Its implementation consists of:

- **defining the structure of functional requirements** in terms of requirements/sub-requirements, associating a "weight" with them that classifies their business "criticality";
- **defining the scenarios/test cases** that verify a given requirement and associate them with it;
- **proceeding to the execution of the test cases** by tracking the results in terms of passed/failed;
- in case of test failure, **associate the malfunction found**.

With the information on the progress of test executions it is possible to calculate the coverage level of the test against the requirements. In addition, by also associating the information of which malfunctions have been detected, the **necessary information can be obtained to assess the business risk associated with a release**.

## Automated Functional Regression Testing

Test automation is used for different purposes:

- **making test activities more efficient** in terms of speed of execution;
- **making the testing activity more reliable, as it is repeatable**, since it is then possible to evaluate the different outcomes of the tests performed in different time sessions and versions of the tested object;
- **implementing the concepts of Continuous Testing & Delivery**, as it can be included in the delivery chains set up in the context of DevOps implementation.

In relation to the previous purposes, usually the automation strategy that is defined is **using it for the execution of regression tests**, which have to be re-executed at each new functional release. This strategy **also allows for a return on investment** as the effort of implementing automation is offset by a reduction in effort and test execution time, with a consequent decrease in delivery time.

Automation maintenance **must be a continuous process** to ensure the efficiency of the entire implementation of DevOPS and Continuous Testing & Delivery concepts. The need for continuous maintenance is due to functional changes that normally occur in the life cycle of the tested application, even on functionality released in previous versions.

Automation is usually carried out with a set of test cases from those previously designed and executed manually.

## Crowdtesting

Crowdtesting is a testing methodology that allows software to be analysed at any stage of development (from prototype to post-release) to **identify bugs and quickly test the usability of the system**. This is done by leveraging a community of testers connected to a web-based, geographically distributed platform and managed by experienced Test Managers (or Customer Success Managers).

Crowdtesting was activated both by engaging customers of the bank in question with different product mixes and non-customers, performing different types of tests:

- **Exploratory Testing:** research of functional issues, both in pre-release and post-release versions, always in a production environment;
- **Customer Journey Analysis:** verification of the entire process, including beyond the digital touchpoints, for opening a current account, validating it from the user's perspective to identify the most critical stages and potential quitting times;
- **New OS Testing:** verification of the application's compatibility with Beta versions of the operating systems being released (Android and iOS) prior to market release.

### TEST OUTPUTS:

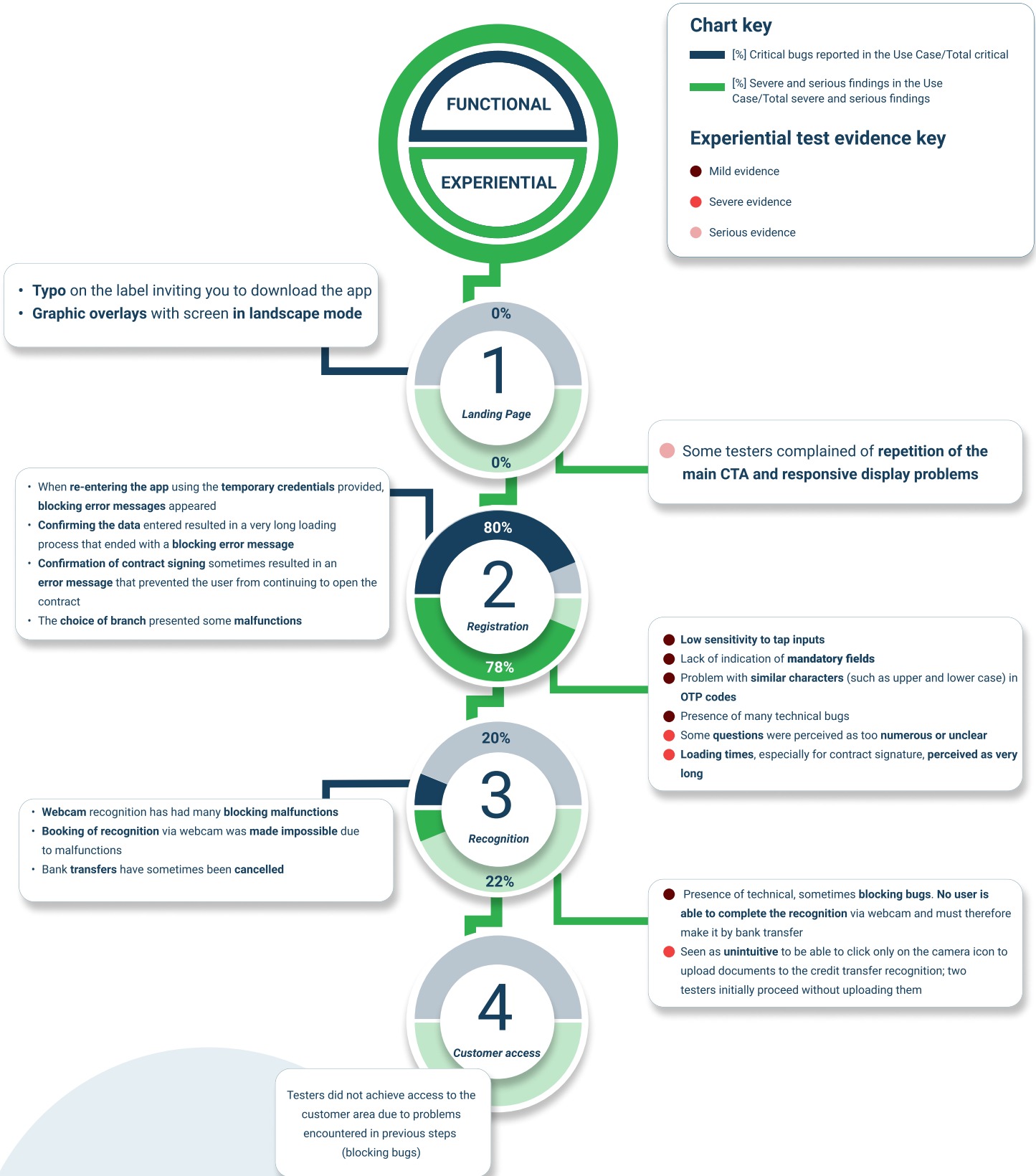
- more than **300 functional bugs** identified in different test cycles;
- more than **15 hrs of video analysed** for user experience optimisation.

### PRODUCT RESULTS:

- Significant **reduction of functional bugs** on versions released on the stores;
- **Compatibility of apps on new OS**, validated before market release;
- **Correct measurement of digital churn rate** in the account opening process and identification of improvement actions;
- **Improvement of the experience** by revising user support information to avoid moments of disorientation and/or discontinuation of account opening.

Crowdtesting  
Customer Journey

The graph represents the various paths experienced by the user to open the account through the new app



*N.B. The experiential test was conducted after the fixing activities on the main problems highlighted by the functional test*

## Automated Smoke Functional Testing

It was convenient to also define the **Smoke Testing sessions that involve the execution of the relevant automated test case suites**. These suites consist of a limited number of already automated regression test cases, so that the required execution time is limited.

The test cases that make up these suites are **tests that verify the most critical functionalities**, the operation of which is fundamental to assessing the state of availability of an environment or provision of a service.

Smoke Testing" sessions are performed in different phases for different purposes:

- during planned test sessions for a new release, they are performed daily **in the test environments** to check availability and operation before starting planned testing activities;
- are run daily **in the production environment** to check their availability and operation;
- these are **carried out after installations of new releases** in production to evaluate and decide on their "Go/"NoGo" in the face of test failure.



## Complementarity of Crowdtest's integrated approach and Test Automation

Features	Exploratory Testing	Test Automation	Crowd Testing
Calculation of test coverage vs requirements, Risk-Based-Testing	No	Yes	No
High test repeatability	No	Yes	Yes
Fast feedback during development phases	Yes	Yes, during the regression phase	No
Fast feedback during service delivery	No	No	Yes
Statistical qualitative evaluations on usability, reliability, consistency, clarity, stability, performance, responsive-design characteristics	Yes, partial	No	Yes, complete
Statistical qualitative assessments of the use of services by users	No	No	Yes
Can be used during the development cycle	Yes	Yes	Yes
Effective during service delivery	No	Not in use	Yes
Allows testing on very large sets of real devices (screen size, OS version), browsers and their different versions	No	No	Yes, very effective
Testing using device emulators	Yes	Yes	No



AppQuality  
Real Crowd, Digital Quality

### ***About AppQuality***

*AppQuality is the first Italian pure player Crowdttesting platform created to test Apps, websites, eCommerce, Chatbots and any other digital product, by leveraging a geographically distributed community of testers that are professionally managed and interconnected by the proprietary remote collaboration platform.*

**CONTACT US**