Developing for Testability: Best Practices and the Opinion and Practice of OutSystems Professionals

Fernando Reinaldo Ribeiro, José Carlos Metrôlho*
R&D Unit in Digital Services, Applications and Content
Polytechnic Institute of Castelo Branco, Portugal

Joana Salgueiro
Polytechnic Institute of Castelo Branco, Portugal

* Presenter, metrolho@ipcb.pt

Instituto Politécnico de Castelo Branco

IARIA

About the presenter: José Carlos Metrôlho, PhD

José Carlos Metrôlho is Adjunct Professor at the Technical-Scientific Unit of Informatics of the School of Technology of the Polytechnic Institute of Castelo Branco, Portugal.

He is member of the R&D Unit in Digital Services, Applications and Content of the Polytechnic Institute of Castelo Branco, Portugal.

His research interests include: Programming Languages and Environments; Mobile App Development; Software Engineering; Agile Software Development.

Outline

- Objective
- Overview
- Test automation on low-code platforms
- Best practices for OutSystems development and its influence on test automation
- Opinion and practice of OutSystems platform practitioners about best practices in development in low-code software testing automation
- Conclusion and future steps

Objective

- Study the importance of the best practices in low-code development and analyze their impact on the test automation process
- Understand how professionals know and apply these best practices and how they value testing activities

Overview

- Low-Code Development Platforms (LCDP)
  - Growing popularity of LCDP and its growing adoption by IT companies
  - They may help fill the gap between business and IT through abstraction and automation and accelerate the software release time
- Some advantages
  - They allow to reduce the software delivery time and to update and deliver new features in shorter periods
  - They allow applications to be built for multiple platforms simultaneously
  - They integrate many of the same tools' functionalities that developers and teams use to design, code, deploy and manage their applications
  - Many of the data integration features have already been developed and can be easily customized
  - Developers may still need to do some coding for a specific task, but a significant part of the job can be done through the drag-and-drop interface
  - …

Overview

- Testing on Low-Code Development Platforms
  - LCDP are often associated with agile development which have implications for the way tests are managed (reduced use of documentation, more frequent interactions with end-users, etc.)
  - LCDP democratize application development to software practitioners with distinct backgrounds
  - In certain situations, the testing process may derail some of the benefits associated with the low-code development and agile methodologies
  - Low-code development is not synonymous with error-free development
- Why OutSystems LCDP
  - There are several LCDP e.g., : Appian, Mendix, OutSystems, Power Apps, Temenos Quantum, etc.
  - OutSystems is widely used by software development companies in Portugal
  - We have a collaboration with OutSystems company for several years, under which we have accessible software licenses
  - OutSystems is one of the leaders in the low-code market (according to “The Forrester Wave”, 2019)

Test automation on low-code platforms

- There are many situations where automated tests are more beneficial than manual testing approaches specially when it may be helpful to repeat tests already carried out, such as regression tests
- Manual testing is often complex, or impractical, or can be time-consuming and vulnerable to inaccurate results
- Test automation enables continuous quality assessment and may save significant time and effort.
- There are specific features of LCDP that raise a set of challenges in low-code testing

Test automation on low-code platforms

- Specific features of LCDP that raise a set of challenges in low-code testing:
  - Test cases are usually derived from the requirements, and it is common to involve partners with low-level technical knowledge in the testing activities, which poses some challenges
  - In software developed using LCDP, several situations should be continuously tested (e.g., many integrations to other services, and these integrations should be continuously tested)
  - LCDP are cloud-based and they support the development of cloud-based applications using cloud resources. Test automation must be adapted to this environment
- Despite the challenges it raises, test automation allows continuous quality assessment, and it is essential in agile and low-code development.

Test automation on OutSystems LCDP

- Several features that allow reducing the probability of errors occurring are available:
  - It performs continuous integrity validation that checks the impact of all changes in application layers (data model, business logic or presentation) to ensure that everything is integrated at the time of implementation.
  - It automatically updates all existing dependencies, when changes are made in the applications data models, API, and architecture
  - It performs an impact analysis for multiple applications when creating deployment plans, evaluating the impact of moving new versions of selected applications to the target environment before the deployment is performed
- As a result of this process, the number of bugs introduced is generally lower than traditional development technologies, leading to fewer test cycles and issue fixes, reducing the effort associated with development and delivery
- However, there is no guarantee that errors will not occur, and the need for testing remains

Test automation on OutSystems LCDP

- The life cycle of an OutSystems application includes several stages when testing activities must be performed. The four levels of testing, provided in the International Software Testing Qualifications Board (ISTQB) classification are included:
  - Component Tests are used to verify the behaviour of code units. In some cases, code units are not easily accessible to be tested. The developers deliver these tests as part of the activities developed in the sprint performed in the development environment and the continuous integration environment. Usually, they are automated tests performed using the BDD Framework
  - Integration Tests are tests to verify the integration with external systems. These tests are critical since it is widespread that LCDP make use of external API. These tests must be performed in the development environment by the developers or Quality Assurance. These tests can be automated
  - System Tests are usually run through a web or mobile interface. They are performed considering the perspective of the end-user or the system (End-to-End tests). The quality team can automate this type of test if they are UI tests. Usually, they are performed in a Quality Assurance environment
  - The clients perform Acceptance Tests. Usually, they are performed manually in the Quality Assurance environment

Best practices for OutSystems development and its influence on test automation

- Regardless of the development platform or programming language used, applications must be developed to facilitate testing activities to facilitate tests that validate its correctness
- A set of good development practices, architectural and design decisions, can significantly facilitate test automation at various levels:
  - Facilitate the automation of Integration Tests
    - Isolate the API consumption in a specific module that exposes the API methods through public actions. Other modules, which need access to the API, will have to do it through this specific module, avoiding implement and run tests on every module that is consuming this specific API
  - Facilitate the automation of System Tests
    - Test automation usually involves simulating and recording a user's interactions in a browser to complete the functionality under test. To be less hard work, test automation tools, which are being used, should correctly identify the web elements found on the web page. To make it possible, it is necessary that the web elements identifiers (names and ID) are easily found and identified by the test tool. It often implies the use of personalized identifiers in place of the identifiers assigned by the development platforms. In applications developed in OutSystems, those elements should be appropriately identified in Service Studio by the developer. The developer must customize the elements' identifiers to ensure that all elements have an identifier that would be uniquely identified during the test automation process

Best practices for OutSystems development and its influence on test automation

- To know how best development practices are applied and the opinion and practices of professionals regarding their use is important
- This is particularly important when referring to LCDP professionals since the allocation of time to facilitate or develop the tests, and the adoption of certain development practices can undermine some of the benefits associated with the use of low-code platforms.

- A survey addressed to IT professionals, with experience in OutSystems development, was performed
- The goal is to analyze their perception of the importance of software testing in low-code development and the influence of the best development practices in test automation

Opinion and practice of OutSystems platform practitioners about best practices in development in low-code software testing automation

- The survey
  - Part one
    - 7 questions
    - Characterize the respondents regarding their experience in the IT area and their experience with LCDP and in the area of software testing and quality
  - Part two (only for participants who had some experience in testing activities)
    - 9 questions
    - Understand how they recognize the influence that best development practices have on the testing automation process and how they apply these best practices in their daily activities
- The participants
  - 27 participants complete part one
  - 25 participants complete part two

Opinion and practice of OutSystems platform practitioners about best practices in development in low-code software testing automation

- Characterization of the respondents (27 respondents)
  - Most of the respondents (48.1%) are between 26 and 30 years old, and 25.9% are between 36 and 40 years old
  - 88.8% of the respondents have between 3 and 15 years of experience, 40.7% have between 3 and 5 years, 25.9% have between 6 and 10 years, and 22.2% have between 11 and 15 years
  - Most of the respondents answered that they use, or have used, HTML/CSS, JavaScript, C#, Java, and PHP
  - All the participants answered that they use, or have used, OutSystems. Two of them pointed out that they have also used two other LCDP
  - 59.3% of respondents are developers and 22.2% of respondents are team leaders or managers
  - In terms of target platforms (web, Android, iOS or Multiplatform), we obtained 26 responses, of which 73.1 % of respondents indicated multi-platform and 26.9% for the web
  - Regarding the development methodology that is most common in the participants' projects, only 1 of the participants answered "Lean", while the remaining 26 participants answered Scrum

Opinion and practice of OutSystems platform practitioners about best practices in development in low-code software testing automation

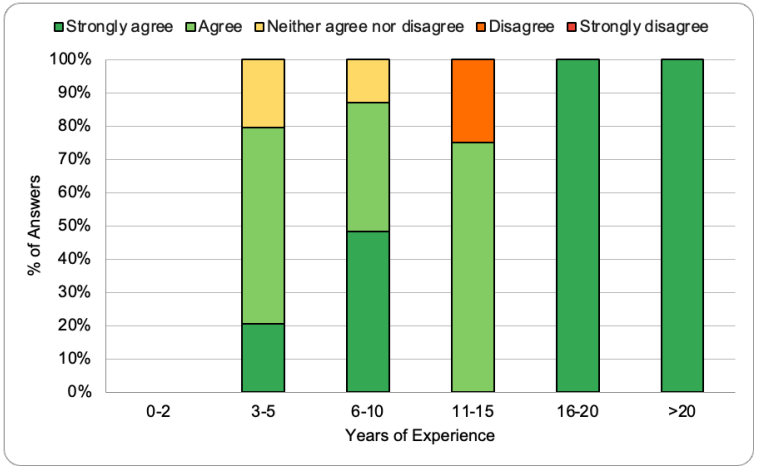
- 100% of the participants considered that "Testing is important and should be performed regardless of the development methodology used"
- 73.9% of the participants answered that they perform unit/component tests, 17.4% answered that they perform system tests and, finally, 8.7% answered that they perform integration tests. These answers seem to be in line with the fact that a significant number of the respondents are currently developers, and therefore unit/component testing is more common.
- BDD Framework is the test tool most used by professionals whose activity includes implementation and execution of tests

Opinion and practice of OutSystems platform practitioners about best practices in development in low-code software testing automation

- 52% of the respondents feel these difficulties sometimes and still 16%, 4 participants, feel difficulties in evaluating what should be tested and how
- 52% of the respondents answered that they strongly agree, and 36% answered that they agree that the way functionalities are described (use cases, user stories, etc.) contribute to facilitating the test design
- 33.3% of the participants answered that they use common sense to write the test cases
- 45.8% answered that they use recognized design techniques, such as BDD (Given - When - Then) and user stories acceptance criteria

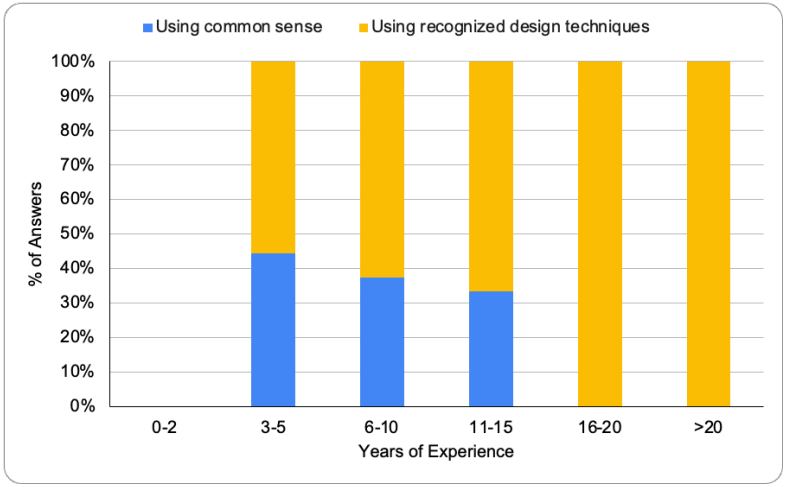
Opinion and practice of OutSystems platform practitioners about best practices in development in low-code software testing automation

- The way how the code is developed can facilitate the implementation of tests
  - Only 25% of the participants, who have between 11 and 15 years of experience, disagree that the way code is developed can facilitate the implementation of tests
  - All professionals with more than 16 years of experience (despite the low number of respondents) strongly agree that the way code is developed to facilitate the implementation of tests
  - Results seem to suggest that professionals with more experience are more aware of this issue



Opinion and practice of OutSystems platform practitioners about best practices in development in low-code software testing automation

- The way they plan and write test cases
  - Some professionals still use only common sense as a way of writing tests (mainly professionals with less experience)
  - Professionals with more experience strongly use recognized test design techniques
  - Results seem to suggest that professionals with more experience are more aware of the importance of using formal techniques



Conclusion

- Developing for software testability is recognized as very important also in the case of LCDP
- Professionals who participated in the survey:
  - Recognize the importance of testing, regardless of the type of application to be developed
  - More than 50% recognize that they often have some difficulty assessing what should be tested and how
  - Recognize that the way functionality is described and how software is implemented influences the process of testing activity
- The code abstraction allowed by these platforms does not exclude the need to follow best practices during the development cycle
- It is also important that professionals have knowledge of adequate testing techniques and tools that allow more support for testing activities