Robotic Process Automation and Quality Assurance – A Perspective

Manjunatha Gurulingaiah Kukkuru,
AVP – Principal Research Analyst, Infosys

Swati Sucharita,
Senior Project Manager, Infosys
Introduction

The emergence of new technology and processes over the past five years has spurred a transformation in quality assurance (QA). The demand for high-velocity deployments and shorter time-to-market makes it imperative for organizations to adopt an agile and automated QA and testing strategy that ensures rapid releases without compromising quality.

As automation becomes an integral part of digital transformation, organizations are increasingly adopting robotic process automation (RPA) since it is easy to implement and uses software bots that reduce operational cost while improving efficiency and accuracy. Thus, powered by artificial intelligence (AI) and machine learning (ML), RPA is driving a new era of enterprise automation – and quality assurance is a key automation candidate.

How organizations can test RPA implementation

Most organizations prefer to develop RPA software robots in multiple iterations using an Agile development methodology as this delivers faster value to clients. However, RPA implementations may include a combination of heterogeneous applications, components and technologies that run on multiple operating systems. Thus, even when it comes to enabling end-to-end testing, open source automation testing tools such as Selenium, Sikuli, AutoIT, Robot class etc. can be leveraged to streamline and accelerate testing.
The best-fit strategy to enable RPA in QA

End-to-end testing and QA offerings for organizations must include a holistic strategy for RPA implementation as shown below:

**Step 1:** Identify repetitive tasks performed by the QA team that can be optimized by implementing RPA

**Step 2:** Identify manual or automated testing requirements for each process

**Step 3:** Instead of testing the entire process, select only those applications / actions / output that are impacted or generated by the RPA bot

**Step 4:** Enable end-to-end automation scripting by combining various open source test automation tools

**Step 5:** Produce a single automation script from creating test data to bot execution and validation across impacted applications / systems

**Step 6:** Reuse the automation script in higher environments like system integration testing (SIT) and user acceptance testing (UAT)

---

**Precondition check / test data creation**

1. Automation script is triggered by the QA team
   i. Automation scripts are created by QA team
   ii. Test data is created or provisioned by the TDM team
2. Automation script retrieves the test data (test data creation is also automated wherever possible)

**Bot execution**

3. Automation script triggers the Bot through a job or from the control hub
4. RPA process is initiated (as seen in the sample workflows of App1, App2, App3 and App4) and actions are performed by following the sequence defined in the RPA process flow

**Validation**

5. Automation script validates App3 output (web page, desktop app or DB) if RPA is successful
6. Same as step 5, depending on the different actions or output by the bot
7. Automation script creates a test report for review

---

**Conclusion**

The demand for better quality, driven by the rapid adoption of disruptive technologies is forcing enterprises to re-evaluate their testing and quality assurance strategies. With technology agnostic and platform independent RPA, QA need to play vital role throughout the development life cycle. Proper automation testing strategy starting from test data creation to triggering the bot and finally validation of business process will improve the quality of bot.