



# Software Test Automation for Radar and EW Systems: Designing the Battlefield of Tomorrow

eBook

 KEYSIGHT



## INTRODUCTION

Testing the hardware components of radar and electronic warfare (EW) systems is crucial to ensuring these mission-critical systems' overall reliability and security. However, beyond hardware, there is an intense focus on testing the software of these systems. This ebook will highlight key developments and best practices in software testing for radar and EW systems. It will also emphasize the role software test automation plays in accelerating releases and reducing errors that can cause delays.



# Software Test Automation and the Development Lifecycle

Software test automation is a critical component of DevOps. In recent years, IT systems began adopting the DevOps approach in private industry and national defense. DevOps refers to practices that automate processes between software development and operations. DevOps aims to build, test, and iteratively release more reliable software faster, often involving and complementing Agile methods.

The concept of DevOps also extends to DevSecOps. The benefits of adopting a DevSecOps approach include shorter release cycles, higher-quality releases incorporating continuous feedback from end-users, and the automation of testing and security.



The benefits of adopting a DevSecOps approach include shorter release cycles, higher-quality releases incorporating continuous feedback from end-users, and the automation of testing and security.





# Software Test Automation is a DoD Project Requirement

Traditionally, software testing involved the manual writing and execution of test scripts. Automated testing enables teams to automate repetitive, manual, and rule-based tasks to accelerate testing and widen test coverage.

The US Department of Defense (DoD) guidelines on software test automation in [DoD Instruction 5000.87](#), Operation of the Software Acquisition Pathway, establish the policy and procedures for “efficient and effective acquisition, development, integration, and timely delivery of secure software.”

The iterative software development approach suggests a constant cycle of planning, coding, building, and testing. More specifically, continuous testing is a requirement of the execution phase of software development. [DoDI 5000.87](#) indicates that whichever software development methodology you adopt must: “incorporate continuous testing and evaluation, resiliency, and cybersecurity, with maximum possible automation, as persistent requirements ... throughout the life cycle.”

# Choosing Appropriate Testing Infrastructure

There are several open-source and commercial tools available to automate software testing. However, because of the unique requirements of radar and EW systems, there are a few factors and capabilities you must consider when assessing software test automation solutions.

For example, testing should cover the user journeys and interactions between users and various devices in radar systems, not just individual software components in isolation. Testing should also consider that frequently, you will need to test a diverse set of interfaces from the command line to graphical user interfaces (GUIs) like touch screens. As such, it's essential to have the ability to test multiple endpoints built on entirely different technologies and programming languages.

With security a priority for mission-critical systems like radar and EW, it's important for testing to occur in a non-invasive manner without access to source code. Testing solutions should exist on-premise but also be available remotely. An application programming interface (API) call or request to the server also risks security, so ensure your testing solution does not require an API call "home."

Testing requirements are not limited to individual systems, components, or pieces of software themselves but rather the supersystems of connected devices and software covering multiple endpoints. As such, you can apply the principles and practices from software test automation in the context of radar systems and EW. The sheer diversity of devices and software means that any software test automation solution must be sufficiently adaptable. The approach goes well beyond simple bug testing. You must test interactions that real users perform, such as using a touch screen and transferring data across a network or a database.



# Creating a Testing Plan

Given the mission-critical nature of software testing, it's essential to build a coherent and robust software testing plan. Consider the broader strategic goals of your radar systems and how users will interact with these systems. In addition, identify the human and technological resources you will need to carry out your testing plan.

Consider the broader strategic goals of your radar systems and how users will interact with these systems.

Consider the various types of testing in your software testing plan:

| Types of testing | Considerations for your plan                                                                                                                                                                                |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Regression       | When you upgrade specific system components, will the previously developed and tested software still work in the desired manner?                                                                            |
| Usability        | Can operators of radar systems access and use the features they need to fully take advantage of the system's capabilities?                                                                                  |
| Performance      | Tests should replicate real-world conditions and assess peak conditions, operating capacity, application, and device integration, response time, and loading speed.                                         |
| Functionality    | These types of tests cover system functionality, user workflows, and accessibility considerations.                                                                                                          |
| Integration      | You must test the integrations to see if the software interacts with or draws information from other systems.                                                                                               |
| Security         | The software must meet the strict security standards and protocols defense agencies demand.                                                                                                                 |
| Exploratory      | The testing solution should be able to track down and identify bugs proactively. Augment software test automation by using artificially intelligent bug detection that will flag errors human testers miss. |



# AI Accelerates Software Test Automation

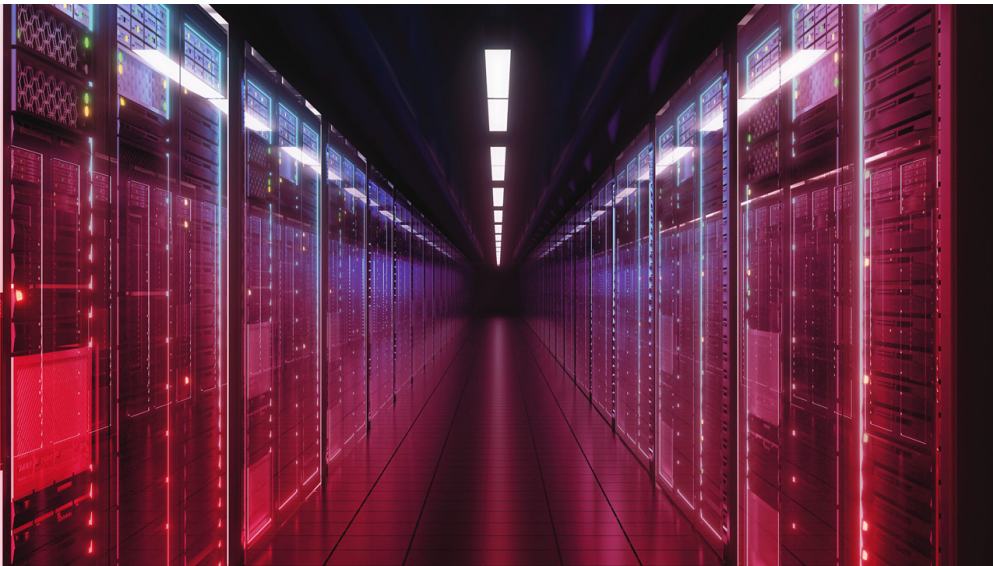
Advances in artificial intelligence (AI) have direct consequences for testing the software of radar and EW systems. AI can augment many areas of software testing — regression, usability, performance, functional, integration, security, and exploratory. In short, AI can interpret and intelligently interact with software interfaces without human input.

AI can also explore the various permutations of actions possible in the interface to generate new user journeys autonomously. The process may unearth otherwise unknown bugs and defects. Exploratory testing is a discipline that traditionally relies heavily on manual human testers. AI-augmented exploratory testing is an effective method to “identify system capabilities, limitations, and deficiencies,” according to the DoDI 5000.89 requirement.



# The Benefits of AI

Software test automation benefits from AI by first interpreting the text and visual elements on the screen via optical character recognition and computer vision, respectively. The process requires the AI to recognize and interpret various screen elements, whether they are input controls (buttons or text fields), navigational components (breadcrumbs or icons), informational components (notifications or message boxes), containers, or other objects. Once you build a model of the application or system under test, intelligent agents, sometimes called bots, will navigate the application and perform various actions.



# How it Works

The training set that enables the AI to navigate a software interface intelligently comes from two sources. The first is exploratory testing models that come from millions of user journeys across all software and devices. The second involves observations of user behavior from interactions with the specific application for testing. You can also instrument the application and collect event-level data from actual usage, which is then used to train the AI.

AI is also helpful for the visual validation of software. Code or object-level testing tools can provide a certain level of confidence that data and objects load correctly. However, you can only achieve certainty if you use visual validation tools to ensure that the elements onscreen reflect the intended outcomes. In other words, you can use computer vision to verify that the user interface behaves as expected. When visual validation fails, it signals to development teams that a behavioral quirk in the software requires a resolution.

Intelligent agents can also stress test systems for performance and load testing. Virtual users can flood an application with visual validation tools to determine the impact on the user experience at peak loads.



## CONCLUSION

The bottom line? AI has a great deal to offer software teams in the radar and EW space. Developers can better understand their software's actual performance, behavior, and utility in real-world scenarios by augmenting existing testing methods through AI and automation. Ultimately, this reduces the amount of uncertainty for the modern warfighter. [Learn more](#) about how Eggplant can level up your radar and EW systems. Better yet, try Eggplant for free with our [product tour, free trial, or live demo](#).





Keysight enables innovators to push the boundaries of engineering by quickly solving design, emulation, and test challenges to create the best product experiences. Start your innovation journey at [www.keysight.com](http://www.keysight.com).

This information is subject to change without notice. © Keysight Technologies, 2023, Published in USA, February 14, 2023, 7123-1010.EN