opentext™

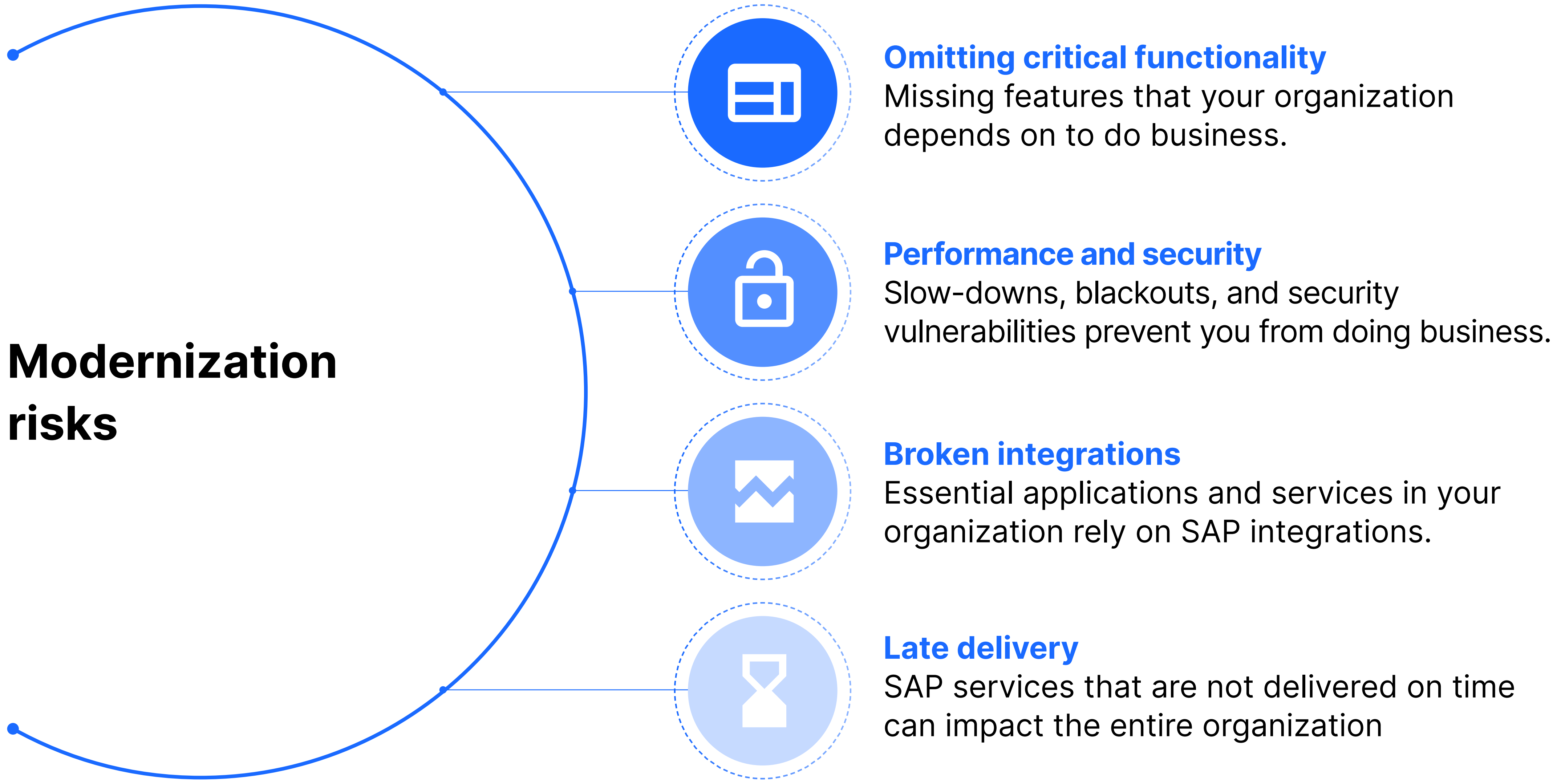
# ADM market insight:

Automate SAP testing with OpenText Functional Testing AI-based capabilities

# Contents

# Introduction

SAP S/4HANA®, the latest enterprise resource planning (ERP) system from SAP, introduces new functionality that supports seamless collaboration, better integration, and real-time analytics. However, fundamental changes in the SAP platform make migrating to S/4HANA complex.



**Modernization risks**

**Omitting critical functionality**
Missing features that your organization depends on to do business.

**Performance and security**
Slow-downs, blackouts, and security vulnerabilities prevent you from doing business.

**Broken integrations**
Essential applications and services in your organization rely on SAP integrations.

**Late delivery**
SAP services that are not delivered on time can impact the entire organization

A significant part of an SAP migration to S/4HANA entails evaluating your current system and deciding which parts to migrate—this migration will take more than a lift-andshift approach. Many legacy systems are old and have accrued functionality that is no longer used or is inefficient. SAP encourages customers to migrate to the new platform with its out-of-the-box integrations and keep customization to a minimum. However, there will always be a need for some customization to accommodate new fields and unique behavior. As a result, you'll need to perform extensive testing.

You can perform testing manually, but test automation can greatly reduce testing time and overhead, allowing for easy and repeatable test runs. Yet, setting up test automation poses new challenges all on its own.

This guide explains the testing challenges associated with migrating to SAP S/4HANA and shows you how to mitigate them using OpenText™ Functional Testing and its AI capabilities.

# Key takeaways

SAP customers must migrate from legacy SAP systems to the new SAP S/4HANA architecture by 2027. That may seem far away now, but the migration process can be arduous. While functional testing is crucial for a successful SAP migration, it can be challenging to ensure that the functionality works correctly.

Functional test automation can greatly reduce testing time and overhead and allows for easy and repeatable test runs. However, setting up test automation poses new challenges. OpenText Functional Testing and its AI capabilities help your team start SAP test automation.

# Testing challenges when migrating legacy apps

SAP S/4HANA supports the digital transformation of business processes and the adoption of Agile development methods, in addition to support for new platform features. However, frequent implementation changes lead to fragile test scripts breaking often.

During a migration, existing modules are often adjusted to accommodate the new fields that have been introduced to the business logic and transportation management system in S/4HANA. This is often a good chance to re-architect applications to take advantage of new features in the platform and new requirements from stakeholders.

Additionally, Agile methodologies are built around incremental develop-build-test cycles. While this has proven helpful for speeding development and lowering technical debt, it inherently contributes frequent changes to the codebase and testing regimen.

As a result, the success of the migration largely depends on the resilience of the test scripts to these frequent changes. These new features also imply that a lot of the existing test cases will either be modified or entirely discarded.

Writing test scripts for the new S/4HANA system with conventional standards and tools will be challenging. The pace of changes to the new system as it's being developed will keep breaking the automation. Considering that test scripts must be written while the new system is still being developed, being able to automate the technologies in the new system can put a lot of stress on the test team.

# Testing challenges when migrating legacy apps

The new S/4HANA user interface (UI), SAP Fiori®, also delivers an improved role-based user experience (UX) across all devices. The UX is designed to fit the role of each user in the business network. While this improvement is great for users, the proliferation of devices and user roles

SAP systems are generally complex structures that run on hybrid landscapes. A major difficulty lies in determining where your various assets are—on premises or in the cloud. You also need to determine which assets are custom-developed and which are out-of-the-box, SAP assets.

As part of the migration requirements, the integrations of the old system (using HTTP, JSON, XML, IDoc, RFCs, OData, and so on), which are mainly written in custom code, need to be re-engineered for the new system. Though several custom code customizations are now supported in SAP S/4HANA, some enterprises might prefer to stick to their trusted custom code.

The result is that thorough testing is required to identify duplicate **functionality, performance, redundancy, security vulnerabilities, and other loopholes** that may arise as you optimize the custom code for the new system. If these tests fail too frequently, it will hamper the progress of the build.

# Optimizing test assets for multiple environments

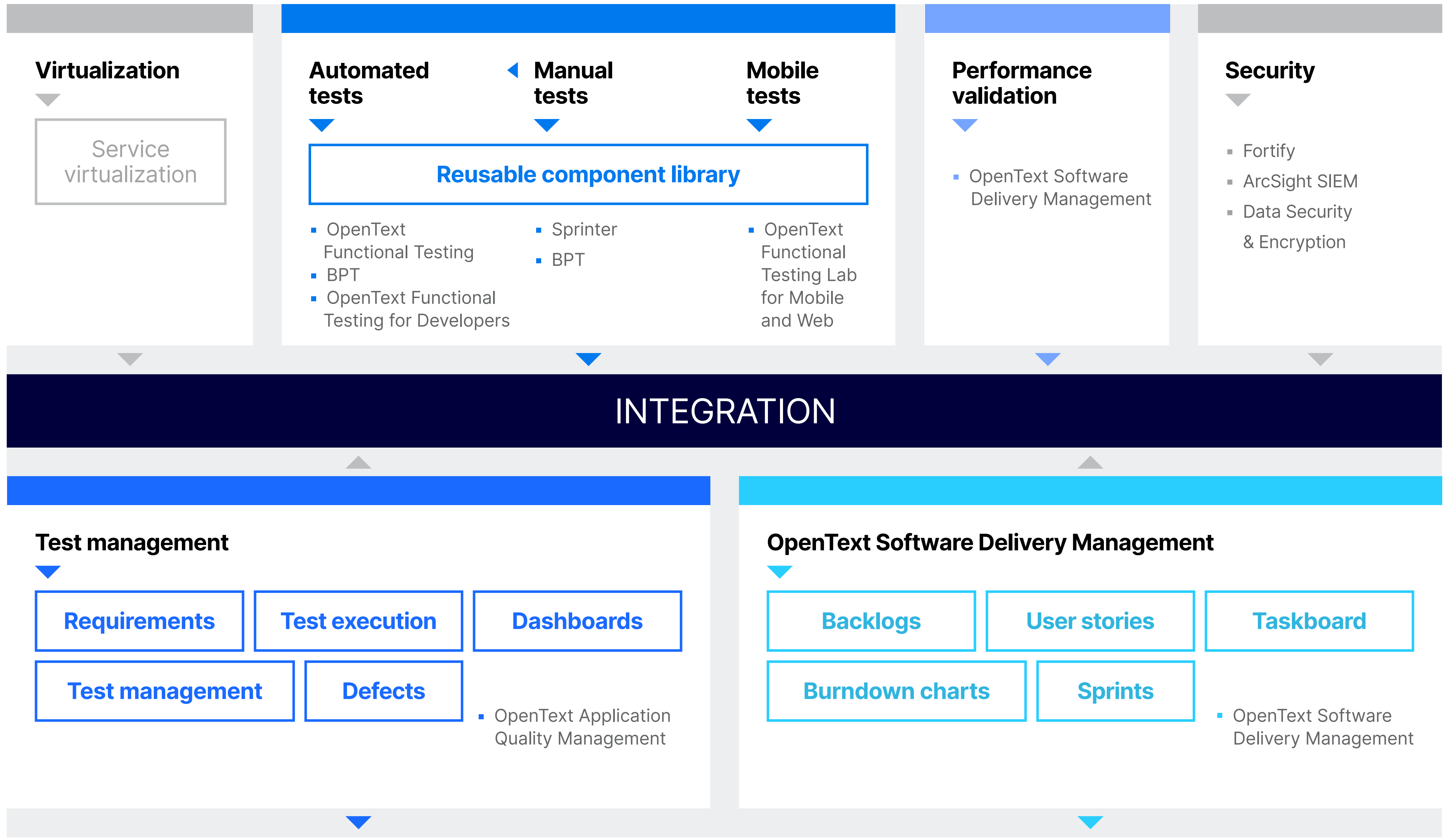
Running tests across different devices that the new system needs to support can be daunting. SAP S/4HANA supports smoother integrations that will help real-time applications run faster and better. It supports more devices and platforms, so you will need to run test cases across multiple environments to ensure that what works on a desktop also works on a smartphone.

This can be time-and cost-intensive using conventional practices. You will need separate test assets for each environment or distinct collections of test objects and the respective information of their properties to enable testing tools to recognize them. These collections of objects and their properties are called object repositories, and you must maintain and update them to align with object updates and their properties in the application to be tested.

# OpenText DevOps Cloud solutions

OpenText enables enterprise IT to accelerate application delivery and build innovative software with less risk, to help businesses drive their digital transformation.

- **OpenText Functional Testing** is an AI-driven, end-to-end testing solution designed to build and maintain test assets that can adapt to today's fast-changing technologies and processes.
- **OpenText performance engineering solutions** enable testers to create and run performance tests on an unparalleled range of application technologies. These tests ensure that applications can withstand the demands placed on them in production. OpenText performance engineering solutions support SAP applications running on the web, as well as SAP GUI applications, and provide broad support for the many generic protocols that SAP applications employ.
- **OpenText™ Application Quality Management** empowers organizations to manage the core application lifecycle, from requirements through deployment, granting application teams the crucial visibility needed for predictable, repeatable, and adaptable delivery of applications. OpenText™ Software Delivery Management is an application lifecycle management platform that enables teams to manage the application delivery pipeline in an Agile or DevOps environment. Organizations often manage their slower-moving back-office systems with OpenText Application Quality Management, and as they modernize their fast-moving front office, they use OpenText Software Delivery Management and synchronize between the two products to provide true end-to-end traceability and governance of their SAP projects.



Virtualization

Service virtualization

Automated tests

Manual tests

Mobile tests

Reusable component library

- OpenText Functional Testing
- BPT
- OpenText Functional Testing for Developers

- Sprinter
- BPT

- OpenText Functional Testing Lab for Mobile and Web

Performance validation

- OpenText Software Delivery Management

Security

- Fortify
- ArcSight SIEM
- Data Security & Encryption

INTEGRATION

Test management

Requirements

Test execution

Dashboards

Test management

Defects

- OpenText Application Quality Management

OpenText Software Delivery Management

Backlogs

User stories

Taskboard

Burndown charts

Sprints

- OpenText Software Delivery Management

IDE, Source code management, Build system

# OpenText DevOps Cloud solutions

- **With Business Process Testing (BPT)**, an integrated, component-based test framework, SAP customers can accelerate functional test automation by creating and maintaining business process tests and flows easily. Bringing together subject matter experts who can build business flows from reusable components, and test automation engineers who can automate them, BPT helps get more people involved in testing. The component-based nature of the framework and the inclusion of subject matter experts in the process is especially relevant to SAP modernization teams using an Agile methodology.

Organizations are increasingly making SAP systems available on mobile platforms. OpenText Functional testing Lab for Mobile and Web provides a centralized, enterprise-level, end-to-end lab and management gateway of real devices and emulators that help teams develop, debug, test, monitor, and optimize their SAP mobile applications.

SAP landscapes contain many interdependencies among functional components, which leads to reliance on services and APIs that internal service providers and third-party vendors provide. When those services are unavailable for any reason, teams are unable to develop and test against them, leading to wait times and extra cycles.

Service Virtualization removes these dependencies by enabling teams to replace them with virtual services that are always available. By accurately simulating the behavior of the actual component, it enables developers and testers to perform functional or performance testing right away, in parallel, even when the real services are not available, when data access is restricted, when data is difficult to attain, or when the services are not suitable for the particular test.

| | OpenText DevOps Solution | SAP Technology |
|---|---|---|
| Functional testing | OpenText Functional Testing (GUI/BPT) | SAPGUI, SAPUI5, Fiori, WDA\J, NWBC, Portal, WebCUIF, ITS, Business networks |
| | OpenText Functional Testing (API Test) | IDoc, RFC |
| | OpenText Functional Testing for Developers | SAPUI5, Fiori SAPGUI |
| | OpenText Functional Testing Lab for Mobile and Web | Fiori (Hybrid App & Pure Web) |
| | Business Process testing | Business users, Packaged Apps Kit |
| | Sprinter | SAPGUI, SAP Web, Fiori Mobile |
| Performance and virtualization | OpenText Professional Performance Engineering | SAPGUI, SAPUI5 |
| | OpenText Enterprise Performance Engineering | SAPGUI, SAPUI5 |
| | OpenText Core Performance Engineering | SAPUI5, Fiori |
| | OpenText Service Virtualization | IDoc, RFC |
| Lifecycle | OpenText Application Quality Management<br>OpenText Software Delivery Management<br>OpenText Project and Portfolio Management | SAFe, Agile, Waterfall Process Documentation |

# Write smarter tests with AI-enabled testing

OpenText Functional Testing brings intelligent test automation to APIs, web, mobile, hybrid, and enterprise apps. Leveraging AI, automated test assets can identify and interact naturally with objects in an application and manipulate them by simulating real-life events and actions. Using OpenText™ Functional Testing for Developers, Agile and DevOps developers and dev-testers can create tests right inside their integrated development environments (IDEs), with the language and testing framework of their choice.

OpenText Functional Testing can significantly increase the reliability and resilience of automated tests, decrease the complexity of automating functional testing processes, and amplify test coverage, as well as reduce the time, effort, and cost of maintaining test assets. And it does all of the above through an intuitive visual user experience.

# Improve the resilience of your tests with OpenText Functional Testing

Migrating from a legacy SAP platform to S/4HANA is a test-intensive process. With many SAP instances supported by custom code, there is often a repository of test scripts that need to be regularly maintained throughout the migration process. Regardless of the migration approach that you choose—brownfield, greenfield, or bluefield—your custom code will need to be re-engineered and adapted to the new platform. This could have a far-reaching impact, owing to the complexity of the SAP infrastructure.

Test scripts that leverage OpenText Functional Testing's built-in AI capabilities are less susceptible to the impacts of ambiguous and frequently changing object properties during SAP migrations. With computer vision capabilities, the computer can identify objects under test by their visual characteristics instead of their code classes and layout positions. For instance, computer vision can recognize a search input field by its characteristic magnifying-glass icon. Using OCR, texts in objects can be verified and retrieved from the app.

The built-in Test Combinations Generator can extract data from the application-under-test and automatically prepare test configuration data for you, which is crucial to ensure sufficient test coverage. Testing often focuses on ensuring that the software performs as intended, and this is often referred to as "happy-path" testing or positive testing. But testing should also investigate unexpected scenarios to ensure that the software handles the situation gracefully, without failing. This is known as negative testing and is as critical as positive testing. By generating different combinations of test data, testers can easily create data-driven tests to perform extensive positive and negative testing of their applications.

**OpenText Functional Testing's machine-driven regression testing also detects errors that can arise when custom code and business processes are modified. It also exposes broken links and visual regressions.**

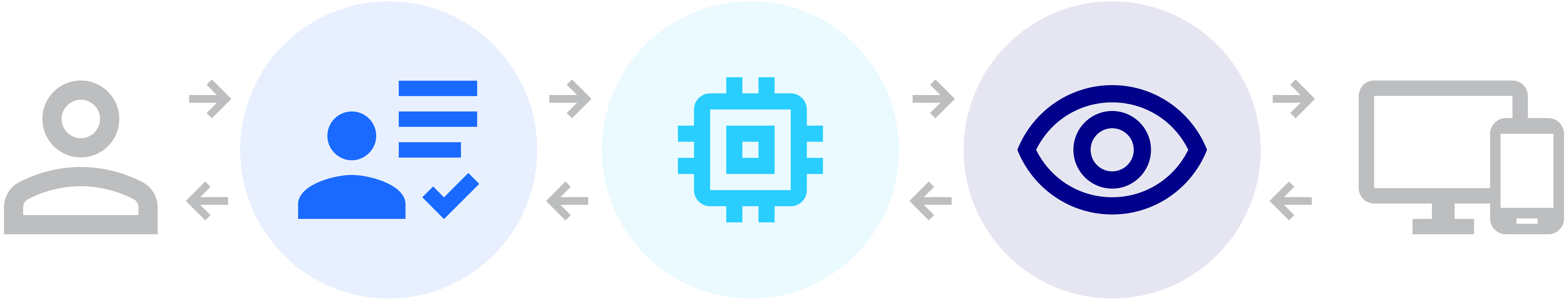
# Post-migration benefits of OpenText Functional Testing test resilience

Prior to S/4HANA, SAP would send updates and releases at most twice annually. With S/4HANA, that number has doubled for cloud releases, and there have been eight on-premises releases since its launch in February 2015. Fiori 2.0 was succeeded by Fiori 3.0 in September 2019, just three years into its lifespan.

Evidently, the SAP platform is evolving faster than it ever has. With advances in data science and machine learning, businesses strive to stay ahead of the curve, which involves getting information in real time and integrating new functionalities faster.

Rapid advances result in equally fast-changing applications. Frequent updates imply the frequent modification of userinterface object location, appearance, and other properties, which invariably throws the software tester back into the maintenance battle.

With OpenText Functional Testing, you can be less anxious about the vulnerability of your test scripts to minor changes after an update, save time on test data configuration, and focus more on ensuring that your updated system is stable across all application environments.



**Natural language**
Writes tests in plain English

**Neural networks**
Interacts with and manipulates objects

**Computer vision**
Identifies objects the same way humans do

# OpenText Functional Testing codes test scripts the way you talk

We've talked about test resilience and the ability to write simple, flexible tests—now let's see what that means in action. The two code snippets below test a search function by telling the computer to search for a speaker. This requires entering a speaker in the input field.

Using Selenium, the test script looks like this:

```
/* Get the WebElement that corresponds to the Input Field for the search functionality (inputField) */
WebElement searchTextBox = driver.findElement(By.id("searchTextBox"));
searchTextField.sendKeys("speaker");
```

With OpenText Functional Testing AI capabilities, the test is much simpler:

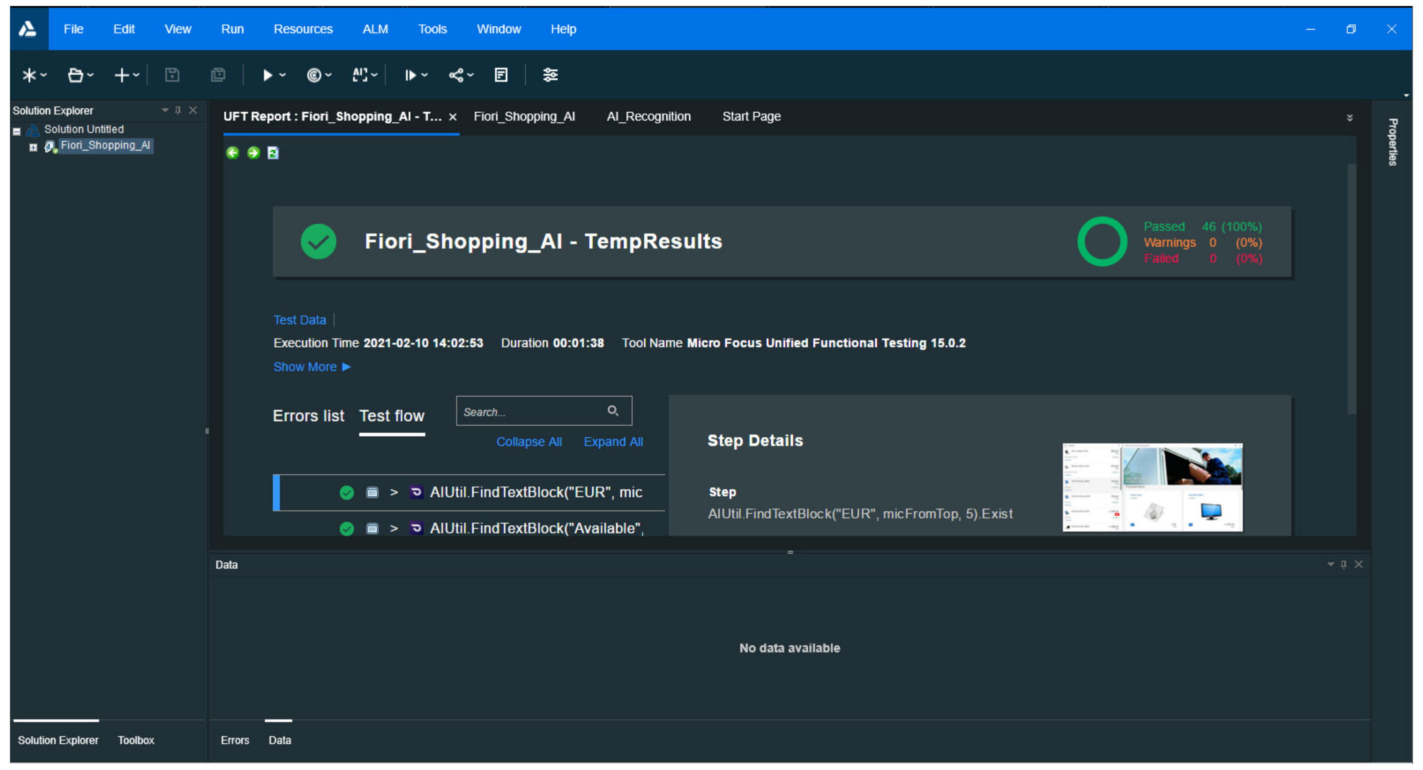
```
'Search for speaker
AIUtil("search").Search "speaker"
```

Conventional test automation tools require an understanding of programming logic and platform-specific scripting practices. In contrast, OpenText Functional Testing's AI Codeless Testing* lets you write scripts that have a near-natural language flow.

**This solution has two advantages:**

1. With OpenText Functional Testing, less-experienced QA personnel, business users, and non-technical team members who understand the business process can understand test scripts better and be more involved in the testing process.
2. Your test scripts are not bound to specific app platforms. Consequently, there is no need to write separate test scripts for all applicable platforms. You can run the same test script

In the end, you get a test report and confidence-level rating on the actions executed. OpenText Functional Testing's AI tells you how sure it is about the decisions it made, and a customizable output file tells you where and when a test step failed.



Fiori_Shopping_AI - TempResults

Passed 46 (100%)
Warnings 0 (0%)
Failed 0 (0%)

Execution Time 2021-02-10 14:02:53 Duration 00:01:38 Tool Name Micro Focus Unified Functional Testing 15.0.2

AIUtil.FindTextBlock("EUR", mic
AIUtil.FindTextBlock("Available",

Step Details

Step
AIUtil.FindTextBlock("EUR", micFromTop, 5).Exist

No data available

# OpenText Functional Testing saves time and resources

Prior to OpenText Functional Testing, you would need to write or adapt four different scripts for the individual environments if you had to test your new SAP S/4HANA deployment across four different environments or app platforms. An alternative would be to have four object repositories where you mapped logical objects to their respective object properties in one script.

Regardless of the approach you chose, you'd still have to maintain four different test assets to test one functionality.

The multi-platform, omnichannel compatibility of OpenText Functional Testing test scripts eliminates the time and resources spent writing and maintaining four test scripts. OpenText Functional Testing's advanced computer vision and OCR can recognize a "sign in" button on an Android device, even if the same button is labelled "log in" on an iOS device.

This compatibility can give you an impressive 75 percent savings in time and resources, which can be redirected towards improving test quality.

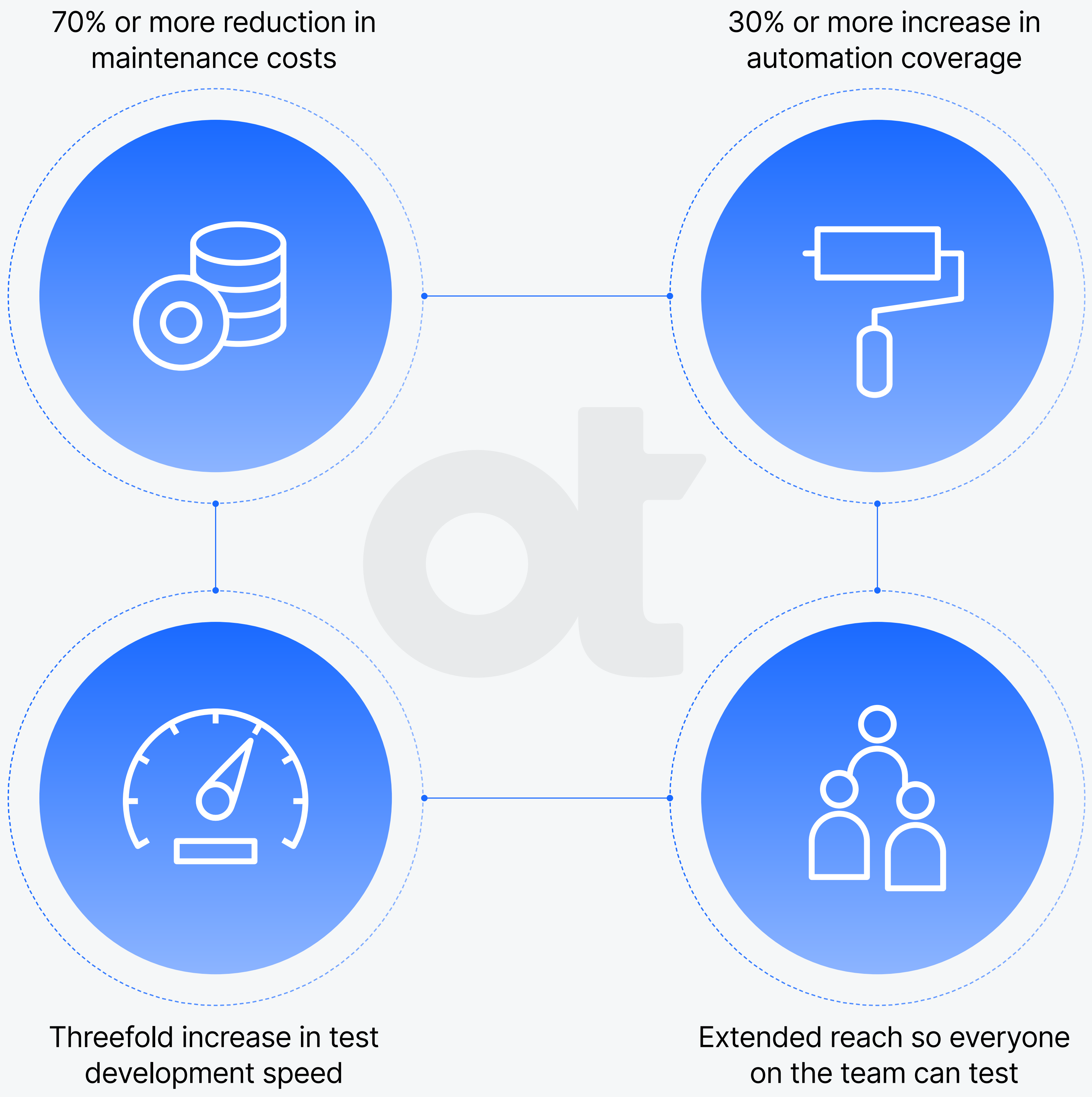
# OpenText Functional Testing helps you truly shift left

Shifting left in an SAP migration context means commencing test activities earlier in the migration process. This test-driven development approach does more than save time and money. It ensures quality too!

The challenge of shifting left during an SAP migration is creating usable test scripts during process configuration and implementing them while executing trial runs. Considering that application mockups and proofs of concept are susceptible to reviews, the system is expected to go through several modifications during the development phases of migration.

Instead of waiting for the migration to be well underway before creating your automation scripts, you can have test scripts ready to go, even before you actually start migrating to S/4HANA. OpenText Functional Testing's AI-based mockup identification lets you inspect mockups and identify objects that you can use in your test. All you need to do is tweak your script accordingly when significant modifications are made to the app—like changes to the business process or module removal or addition.

## Key benefits of OpenText Functional Testing's AI capabilities



70% or more reduction in maintenance costs

30% or more increase in automation coverage

Threefold increase in test development speed

Extended reach so everyone on the team can test

Try out OpenText Functional Testing to see how it can make your SAP migration process more error-proof while saving you time and resources. Sign up for a free 30-day trial ›

## About OpenText

OpenText, The Information Company, enables organizations to gain insight through market leading information management solutions, on premises or in the cloud. For more information about OpenText (NASDAQ: OTEX, TSX: OTEX) visit opentext.com.

opentext.com | X (formerly Twitter) | LinkedIn | CEO Blog

opentext™