



Why AI is the future of
effective SAP releases





INTRODUCTION

Since the 1950s, the concept of artificial intelligence has captivated our imaginations — from science fiction and film to algorithms that can approximate conversation, to the predictive text programs that can produce new [Harry Potter fan fiction](#) today.

Generally, “artificial intelligence” refers to the idea of a machine that can make complex, human-like decisions and judgments. While movies like Terminator (1984) and Blade Runner (1982) use the concept of AI to question what it means to be human, these fictions are somewhat removed from the logistics, mechanics, and applications of AI as we use them today.

Consequently, business leaders and other non-technical stakeholders struggle to match their understanding (and imaginations) of AI to their expectations around business outcomes.



SO WHAT IS AI?

AI is a blend of three sciences that form a system that enables intelligent behavior.

- Psychology - how the mind's cognitive processes work
- Philosophy - how the mind interprets reality and classifies its observations
- Computer science - mathematics, logic, and statistics

Several disciplines have formed under the umbrella of AI, many of which may sound familiar:

- **Artificial neural networks** comprise layers of algorithms structured to mimic the neural networks in the human brain. More layers of neural networks allow for more complex decision-making.
- **Machine learning (ML)** applications use one to three neural networks to discern patterns in data and make predictions based on that data. ML programs get better at a task on their own, without human-written code for every possible situation or input.
- **Deep learning** refers to ML applications using more than three neural networks. Deep learning programs “learn” with less human intervention, and they can process unstructured data — pivotal factors for scaling the use of AI.

As more industries, business processes, and personal aspects of our daily lives have become digitized, almost every move we make produces a new data point. The challenge and opportunity of this data lie in its immense volume.

Cloud computing makes it possible to store more data than ever, but AI grants the ability to manipulate and wield data for business intelligence. AI is vital for using, scaling, and sustaining data strategies.



AI AND SAP

In 2020, SAP released its [Intelligence Enterprise Framework](#), which seeks to holistically align business processes and infrastructure to help users stay competitive and achieve transformation.

Intelligence tools form a significant pillar of this model, along with data management, analytics, and app development and integration. Some of the stack includes:

- SAP's Conversational AI: Lets users build and manage chatbots for both internal and customer-facing functions.
- SAP's Intelligent Robotic Process Automation: Lets users create bot workflows that automate manual tasks and processes across both SAP and non-SAP systems.
- S/4HANA's Situation Handling framework: Responds to exceptional situations and notifies the proper team members. For example, if a sales order is canceled and makes material available for further sales, the material planner is immediately notified without waiting for the next MRP to run.
- SAP's Data Intelligence: Uses built-in ML components to optimize processes and provide predictive analytics. It includes pre-configured scenarios for purchasing, accounting, logistics, sales, and many others.
- S/4HANA's IoT scenarios: Allow for live goods tracking, enabling sales staff to react quickly to complications in transport and trigger replacement deliveries.

These intelligent solutions that once seemed so futuristic and inaccessible have broken into the mainstream.

SAP and other packaged apps are built to be widely useable and therefore aren't as cutting-edge as they could be. The focus on AI tools and the push to S/4HANA signal a rise in the median level of innovation and iteration, another new normal to which enterprises must adapt.



AI IN SOFTWARE DEVELOPMENT

[Microsoft](#), [Google](#), [IBM](#), and other big names in tech are working to build powerful AI/ML developer tools. Each of their platforms is built to integrate and support open-source frameworks and collaboration. The evolution of AI, it seems, will be a group project.

Agile and DevOps comprise the human element of a modern, fast-paced approach. In the context of software development, AI-enabled processes empower teams to innovate at an unprecedented rate through:

➤ Increasing the velocity of development

AI and ML can drastically reduce the manual work in the SDLC. An ML platform can verify deployments and augment the coding process, as well as detect, diagnose, and repair defects automatically.

➤ Empowering developers for higher work

AI will never replace human developers, but developers will use AI-driven insights to create better code. AI can manage the most laborious parts of coding, giving dev teams more bandwidth for innovation. Not to mention, there is a growing need for developers who can code AI/ML tools.

➤ Validating opportunities and mitigating risk

AI can detect and process data previously invisible to businesses and strategists. When AI collects data on user behavior, it can more accurately assess the performance of new software and better detect potential points of failure. Business leaders can use this information to identify which initiatives will provide the highest value to the business strategy and developers can code solutions that better fulfill those initiatives.

But the biggest impediment to fast releases isn't development — it's testing. Testing is almost always underestimated as a strategic value-add in the development process, so it's treated as a chore. AI can drastically streamline development, but if that innovation isn't applied to testing, the bottleneck will remain.



AI IN SOFTWARE TESTING

[We polled organizations](#) about the top challenges they face when updating their SAP systems.

- 51% said SAP updates put their business processes at risk.
- 46% said SAP updates take longer than desired.
- 41% said the cost of updating their SAP systems is a concern.
- 26% reported frequent issues and bugs after updating their SAP systems.
- [90% of SAP enterprise users](#) plan for a period of hypercare after a release goes live.

That last bullet speaks volumes — organizations plan for failure in every release. Resolving a defect post-release is the most expensive, laborious, and complicated time to do so. Not to mention the cost downtime in terms of productivity: \$10,000 per minute, according to a study by [the Ponemon Institute](#).

QA teams simply can't run the tests necessary to ensure a smooth, quality release. The reasons for this are vast and AI speaks to almost every one of them.

➤ Understanding the impact of your updates

When SAP releases a new update, it can be a guessing game to determine what functions and applications it will affect. Are you over-testing? Under-testing? Can you even know?

AI can tell your teams exactly what to test and how to prioritize those tests.

When a new change comes in, an AI-powered impact analysis tool identifies all environments impacted by the change and notes which tests are necessary to test that change. The AI automatically runs the necessary tests that are already automated and delivers those results.

As for the necessary tests that aren't automated, the AI adds those tests to your team's backlog, so they'll know what to build for the next sprint.

Enterprise QA teams who can conduct change impact analysis typically [reduce test scopes by more than 80%](#). Knowing what to test is key to streamlining your migration, which comes with the challenge of maintaining both the legacy tool and the new one.

➤ Enable risk-based testing

The risk to business processes and the costs of downtime and hypercare will only grow more untenable. Yet most organizations don't even calculate risk coverage.

[Our research](#) has revealed these insights: Most organizations begin their automation journeys at around 50% risk coverage and on average, 80% of risk can be covered by 20% of test cases.

Several factors can cloud your picture of release readiness, especially under the pressure of an ambitious release cycle, like:

- Developers introducing ABAP code into delivery pipelines
- Untested code and configuration changes slipping into production
- Missing dependencies in custom code transports
- The inability to detect out-of-sync objects

When deployed in an SAP environment, an AI tool analyzes your SAP systems and usage to learn everything about your SAP landscape, including business processes, integrations, custom code, and security.

AI gathers data from your production systems to predict the risks of a new update or release. When

integrated with a [test automation](#) platform, this tool can automatically identify and run the proper regression tests. When the AI detects an at-risk object that lacks a test case, it can generate requirements for tests needed to fill that gap.

[See how an apparel company used risk-based testing to go live with an eight-year backlog of SAP support packs with zero defects.](#)

➤ Democratize quality

AI brings the benefits of low/no-code to the realm of QA, increasing speed and minimizing workload.

AI uses object recognition functionality to scan an application's UI, a mockup, or even a mere UI description to create a business-readable model that can be tested before the first line of code is written.

Because this type of AI recognizes visual UI elements, rather than creating a model based on underlying code, it's just as easy to develop models based on legacy apps and customizations — 40% faster than coding them.

Producing a business-readable model makes test creation accessible to non-technical members of your team. It gives business users the power to create their own tests, closing the loop on user-generated customizations.

➤ Improve automated test case design

Combinatorial test case generators can automate the creation of new test cases, but left unchecked, these tools can bloat your test library with meaningless test cases.

Typically, a business user helps guide the development of the initial test set, ensuring the accommodation of countless dependencies between attributes and instances. Then the business user moves on without thoroughly documenting the purpose of each test case. When the test case needs to be adjusted for a change in the app or business logic, developers lack the insight to do so effectively.

AI enables you to define rules between attributes and instances, and these rules factor into the automated test generation process. You can define rules that forbid, constrain, and confine certain combinations of attributes and instances.

More importantly, you can define rules that account for scenarios that wouldn't have been created by combinatorial methods, enabling you to fill gaps in risk coverage.

➤ Streamline test maintenance

Automation is key to optimizing test processes and setting the stage for Agile and DevOps workflows. However, in [a study conducted in partnership with ASUG](#), Tricentis found that SAP customers tended toward low automation rates of SAP customers:

- 44% had an automation rate of less than 25%. Only 11% had an automation rate higher than 51%.
- 14% didn't know how much of their testing was automated.
- 18% weren't practicing automation at all.

Once automation rates reach 40%, teams spend so much time maintaining test scripts that they don't have time to automate more manual tests or pursue higher innovation projects. As the testing burden continues to swell under the release cycles of a complex systems landscape, automation rates sink back down to under 20%.

So why do test cases fail in the first place?

When changes are made to the application, the underlying code destabilizes the identifiers used to locate and automate an object under test.

Self-healing AI tracks historic executions and monitors stable identifiers. When a failure occurs, the AI automatically generates a new, stable identifier, which “heals” the broken test suite.

There are two approaches to self-healing AI:

- **Technical self-healing** analyzes the DOM structure to identify changes in the code. The AI suggests different identifiers that locate the object. Testers can usually adjust the level of AI accuracy to regulate precision.
- **Visual self-healing** uses neural networks to detect changes in visual elements, like minor UI changes or technical changes to the control, then automatically updates test cases accordingly.

Self-healing AI makes your tests much more resilient, saves your team hours of refactoring work, and enables Agile and DevOps deployment.

[See how an AI-forward medical company used an AI-powered testing solution to achieve 3x faster test cycles, reduce testing costs by 35%, and save 93 testing hours per month.](#)

➤ De-clutter test case libraries

Bloated test case libraries are another instance where a lack of insight leads to overwork and underperformance. With more to test and shortened release cycles, you can't run every test in your test portfolio.

The average level of [test redundancy in enterprise test portfolios is 67%](#). Think of those implications. How much time are you spending creating, migrating, and maintaining unnecessary tests?

This happens for a few different reasons:

- Teams create custom code and tests for non-standard functionality. When a new SAP release includes a native version of that functionality, the custom code and tests aren't erased.
- In the test maintenance rat race, it becomes easier to create a new test than update an old one, leading to a glut of obsolete test cases.
- Siloed teams often lack a central repository for test assets and data. Consequently, they create tests that have already been created by another team.

AI can identify redundant, stale, low-value tests for a given SAP transaction. Furthermore, AI can also prioritize tests that best support the technical requirements of the update to ensure business value and reduce test scope.

➤ Get more out of exploratory testing

When testers have more bandwidth, they can conduct exploratory testing, which lets them explore the product in a natural, unstructured manner.

This lets testers view the application from the user's perspective to evaluate its usability beyond its functionality. In Agile workflows, exploratory testing lets teams test new features quickly without revising the broader test plan.

The informal nature of exploratory testing is both its pro and its con. When a new defect is found, testers must create a new test on the spot. Traceability or documentation of exploratory testing pose a challenge. In a recorded exploratory testing session, an AI tool can automatically capture and record interactions, taking screenshots of each issue and logging them into your Agile ALM. This creates an automatic audit trail that enables developers to take action faster.

AI can use its intelligent capturing engine to turn exploratory testing sessions into automated test scripts.



THE BIG PICTURE

As you consider the implications of AI in your business, development, and testing processes, your mind may leap at the promise of being able to do more. That promise is certainly real, but the true value of AI is the ability to do what matters.

Three things to remember. AI can:

1. Expand automation to improve test coverage and facilitate Agile and DevOps workflows.
2. Enable high-value tests like end-to-end testing to catch defects before they become a nightmare to fix.
3. Drastically reduce test scopes by prioritizing tests that provide the most value and cover the most risk.

AI-driven solutions aren't free, but the cost of inefficient processes probably drains your budget more than you realize. [According to a Forrester study](#), SAP users who adopted AI tools for risk-based testing and impact change analysis achieved a 78% reduction in production errors and reduced costs associated with both testing and hypercare. The same study showed that AI-powered testing solutions helped enterprises reduce the cost of testing by 25%, achieve 334% ROI within three years, and pay for themselves in under six months.

An old adage says you can't have your cake and eat it too. In the software space, this means you can have a solution that can be fast, affordable, or high-quality — but never all three. The emergence of AI has the potential to break that paradigm and eliminate the testing bottleneck that stymies the success of SAP releases.

DISCLAIMER: Note, the information provided in this statement should not be considered as legal advice. Readers are cautioned not to place undue reliance on these statements, and they should not be relied upon in making purchasing decisions or for achieving compliance to legal regulations.



ABOUT TRICENTIS

Tricentis is a global leader in enterprise continuous testing, widely credited for reinventing software testing for DevOps, cloud, and enterprise applications. The Tricentis AI-based, continuous testing platform provides a new and fundamentally different way to perform software testing. An approach that's totally automated, fully codeless, and intelligently driven by AI. It addresses both agile development and complex enterprise apps, enabling enterprises to accelerate their digital transformation by dramatically increasing software release speed, reducing costs, and improving software quality. Tricentis has been widely recognized as the leader by all major industry analysts, including being named the leader in Gartner's Magic Quadrant five years in a row. Tricentis has more than 2,100 customers, including the largest brands in the world, such as McKesson, Accenture, Nationwide Insurance, Allianz, Telstra, Dolby, and Vodafone.

To learn more, visit www.tricentis.com or follow us on [LinkedIn](#), [Twitter](#), and [Facebook](#).

Tricentis has locations across the globe. To find an office near you, visit www.tricentis.com/locations