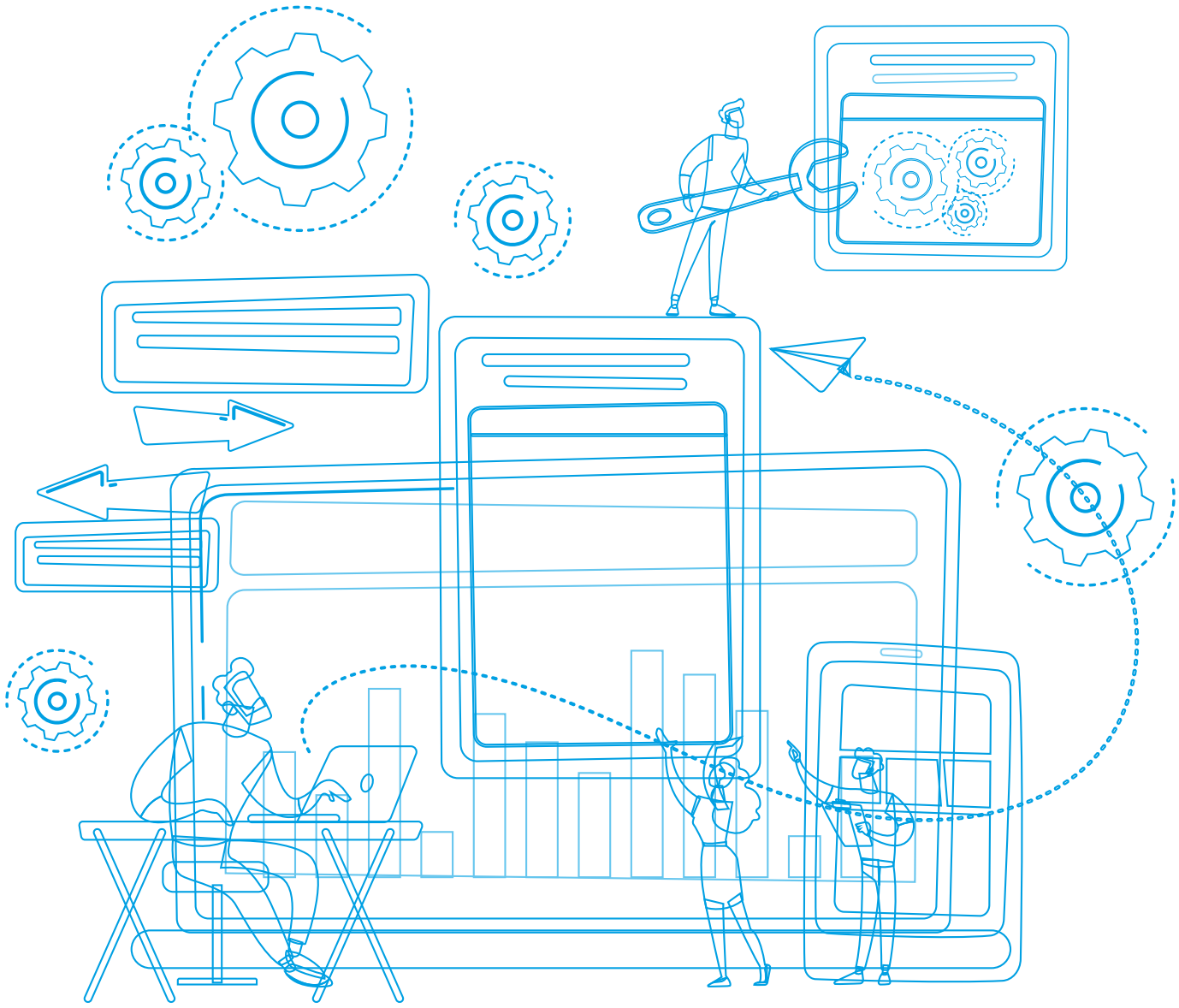
CODIFICATION

Unlocking
Test Automation



August 2022

Field Guide

Table of Contents

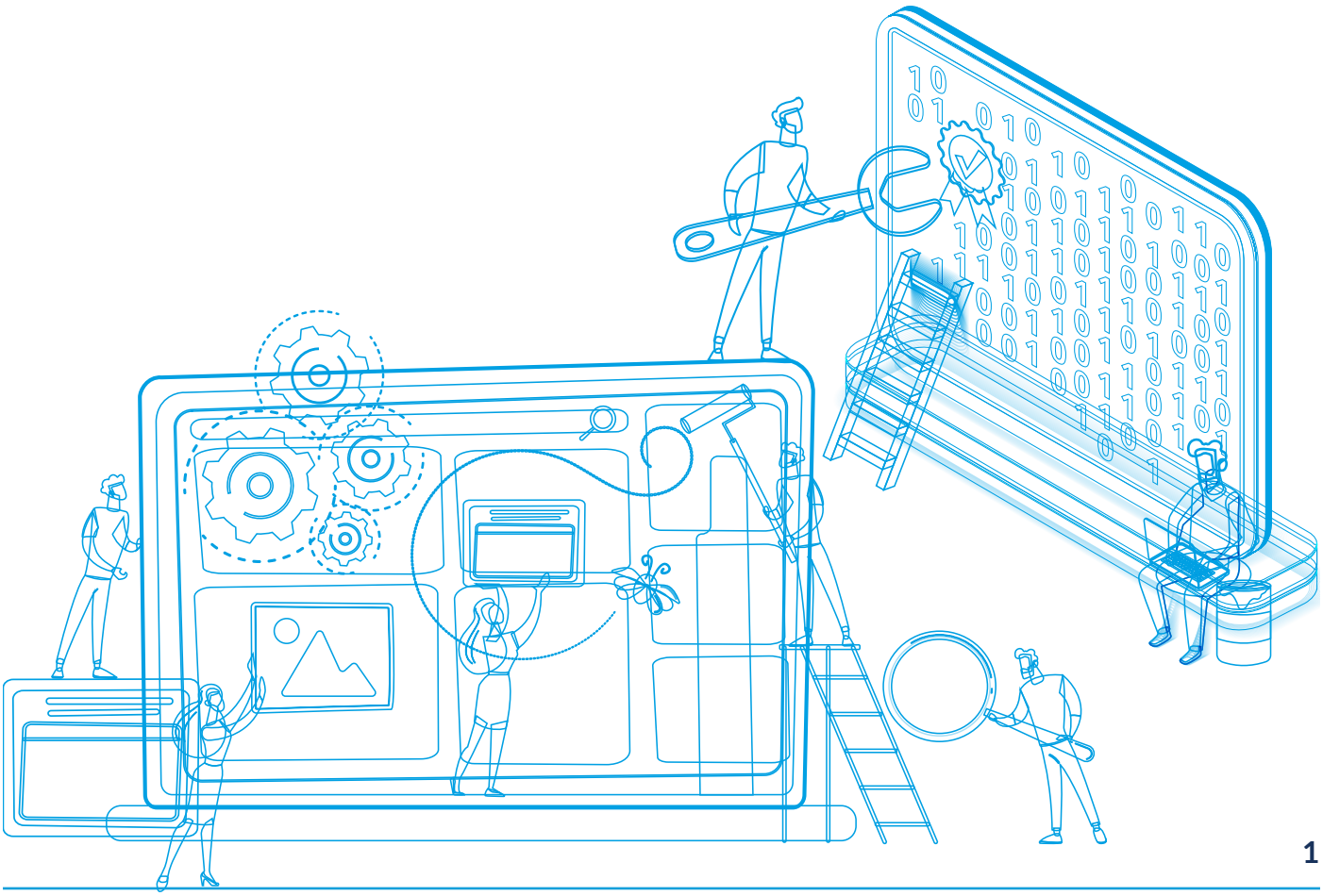
Introduction

Testing is a term that will be known to many in the technology industry. It entails designing and constructing a product or piece of software and then testing it repeatedly until it works.

It is essential to make sure your product is as fantastic as you want it to be and is a crucial aspect of the product development cycle.

Testing is now more sophisticated and advanced than ever. To reduce the effort of the rest of the team and deliver precise and comprehensible findings, we have begun to use automation testing.

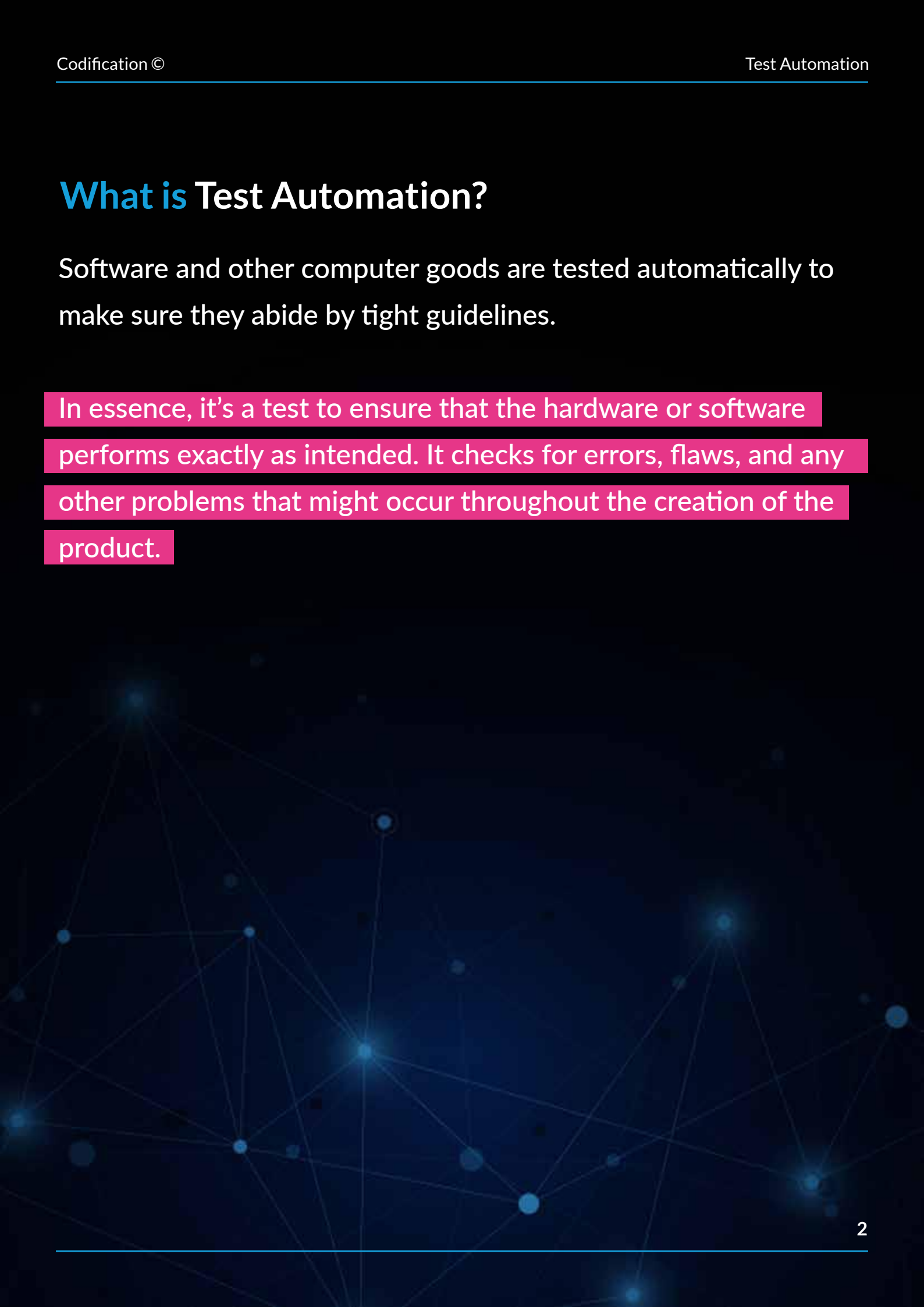
In this section, we'll go into more detail on automation testing, including what it is and how to start implementing it in your company.



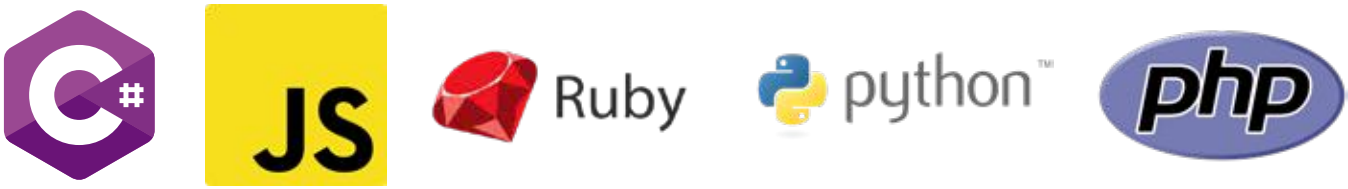
What is Test Automation?

Software and other computer goods are tested automatically to make sure they abide by tight guidelines.

In essence, it's a test to ensure that the hardware or software performs exactly as intended. It checks for errors, flaws, and any other problems that might occur throughout the creation of the product.



Although some testing methods, such as functional or regression testing, can be carried out manually, doing so has fewer advantages. Any time of day can be used to do automation testing. It looks at the software using scripted sequences. It then summarises what was discovered, and this data can be compared to results from earlier test runs. The most common programming languages used by automation developers are:



Many software companies will employ an automation tester for QA (quality assurance). In the first stages, they create and write the test scripts. To test the software and goods, the QA automation tester will collaborate with automation test engineers and product developers. They will develop a team, take charge of the test automation projects, and experiment with various test automation frameworks before selecting the most effective one.

The team must be knowledgeable about the attributes, runners, assertions, screenshots, test suites, and CI before beginning to work with unit testing frameworks (continuous integration). JUnit and Pytest are well-known user testing frameworks for Java and Python, respectively.

In the technological sector, observing testing protocols is crucial. Continuous delivery (CD) and continuous testing depend on it (CT). Both CD and CT will be used in the testing strategies of DevOps and agile software development teams.

Businesses can optimise their testing processes to provide the most return on investment by deciding to use automation testing (ROI). Why? Because automated testing may speed up the development process, remove the chance of human error, and automate tedious and time-consuming jobs.

Why is Test Automation Needed?

Some teams merely lack the time and resources necessary to test software manually. Automation can be useful here. Because it operates swiftly and effectively, testing products may be carried out in a fraction of the time. This gives developers and production managers back their time so they can focus on other project areas. As a result, productivity can increase significantly.

Utilising automation technologies also allows for more regular testing, which enhances functionality in general.

Cycles of software development necessitate recurrent testing, frequently the same test. This is made possible via automation testing, freeing up team members for other tasks.

Compared to manual testing alone, it can also produce results that are more precise and dependable. A further assurance that the product is prepared for the market or to advance to the following development stage. This affirmation encourages the team to keep improving.

The benefits of automation to product development are most significant. That's because the more effective design and production methods allow for continual development after a product has been released, whether it's software, an app, or something else. In essence, automation will enable the company to produce more software and goods with the same number of team members. This implies that they constantly develop new software and refine the final goods they provide.

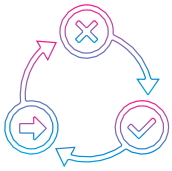
Benefits of Automation Testing

Automation technology is being used by many companies all over the world since software testing offers several advantages. Some of the main advantages of employing automation testing for software development are listed below:

- Better reporting
- Better bug detection
- Making the testing process more simple
- Making the testing process faster
- Cuts down on human intervention
- Cost and time savings

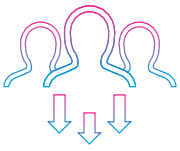
Better reporting

Well-written test cases for numerous scenarios are used during automation testing. These planned sequences can cover a lot of ground and produce in-depth reports that are simply impossible for a human to produce. not to mention delivering them in less time.



Making the testing process more simple

Most SaaS and tech organizations regularly test their products as part of daily operations. The key is to keep things as basic as you can. Automation has a lot of advantages. It is possible to reuse test scripts when automating test tools. While manual testing only requires one line of code to be created each time a test case needs to be performed.

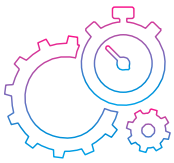


Cuts down on human intervention

Without a human in charge, tests can be carried out at any hour of the day or night. Additionally, when done automatically, this can lessen the possibility of human error.

Better bug detection

Finding bugs and other flaws in a product is one of the key purposes of testing it. This procedure can be made simpler with automation testing. Additionally, it can examine a greater test coverage than perhaps people can.



Making the testing process faster

Humans cannot keep up with automated technology and machines. This is why we employ them, along with increased accuracy. Your software development cycles are subsequently shortened by this.

Cost and time savings

Testing might take a lot of time. Automation might become more cost-effective over time even if it may need an upfront investment. In many circumstances, team members are no longer obliged to perform manual testing and instead spend their time in other ways. Their workflow is enhanced by this.

What Test Cases Should be Automated?

The entire testing process cannot be automated at once. As a result, you must choose which tests to automate first.

Let's examine the kind of tests that should be automated because they can benefit from it:

- ✓ Tests that run the risk of failing due to human error
- ✓ Tests that are repeated and monotonous
- ✓ Harder testing that requires several data sets
- ✓ Tests that are not possible to do manually
- ✓ Test that would take a long time to do manually
- ✓ Tests with high risks
- ✓ Tests that must be carried out on multiple hardware and/or software platforms

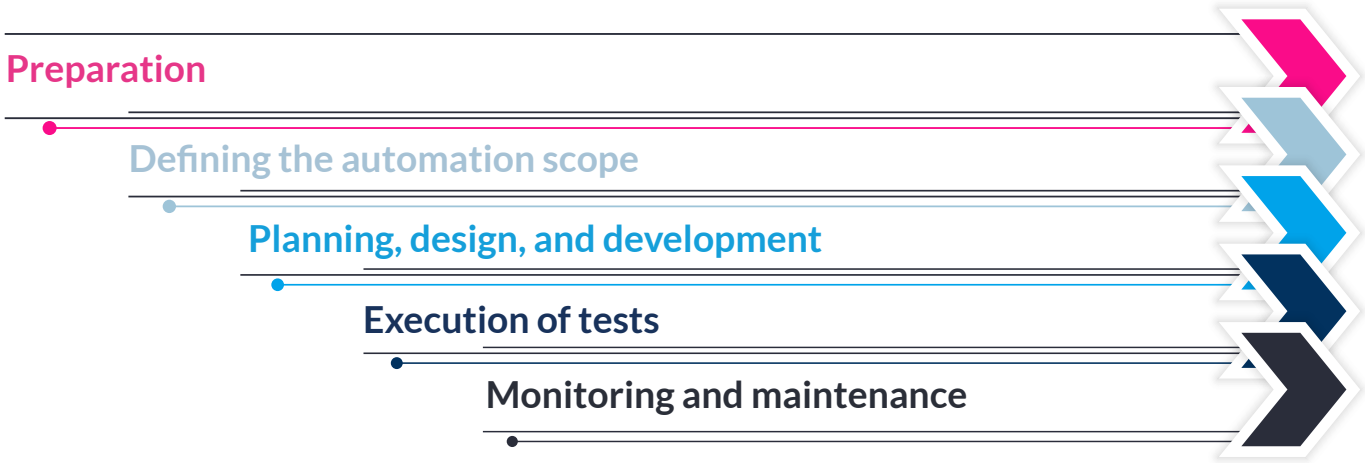
Test cases below are not suited for automation:

- ✗ Newly created test cases that haven't been manually run at least once
- ✗ Test cases when the requirements change regularly
- ✗ Ad-hoc test cases

Automated Testing Process

In an automation process, the following steps are carried out:

1. Preparation
2. Defining the automation scope
3. Planning, design, and development
4. Execution of tests
5. Monitoring and maintenance



1. Preparation

We must first arrange the environment, test data, and state where tests will take place. As we've seen, the majority of tests demand that the environment is in a specific condition before an action may occur. Usually, some preparation is needed for this. Either the data will need to be altered, the application will need to be placed in a particular condition, or both will need to be done.

2. Defining the automation

The area of your application under test that will be automated is known as the scope of automation. The following details aid in determining the scope:

- The characteristics that are crucial to the business
- Scenarios with a lot of information
- Cross-application capabilities that are common
- Technology readiness
- How much of a business's components are recycled
- The difficulty of the test cases
- Cross-browser testing capability to use the same test cases

3. Planning, design, and development

You develop an automation strategy and plan at this phase, which includes the following information:

- Chosen automation tools
- The characteristics and design of a framework
- Items of automation that are both in and out of scope
- Preparing a testbed for automation
- Schedule and execution timeline for the script
- Results of automated testing

4. Executing of tests

This stage involves the execution of scripts. Before the scripts are scheduled to run, test data must be entered. After being used, they offer thorough test findings.

Either the automation tool itself or the test management tool, which will call the automation tool, might be used for execution.

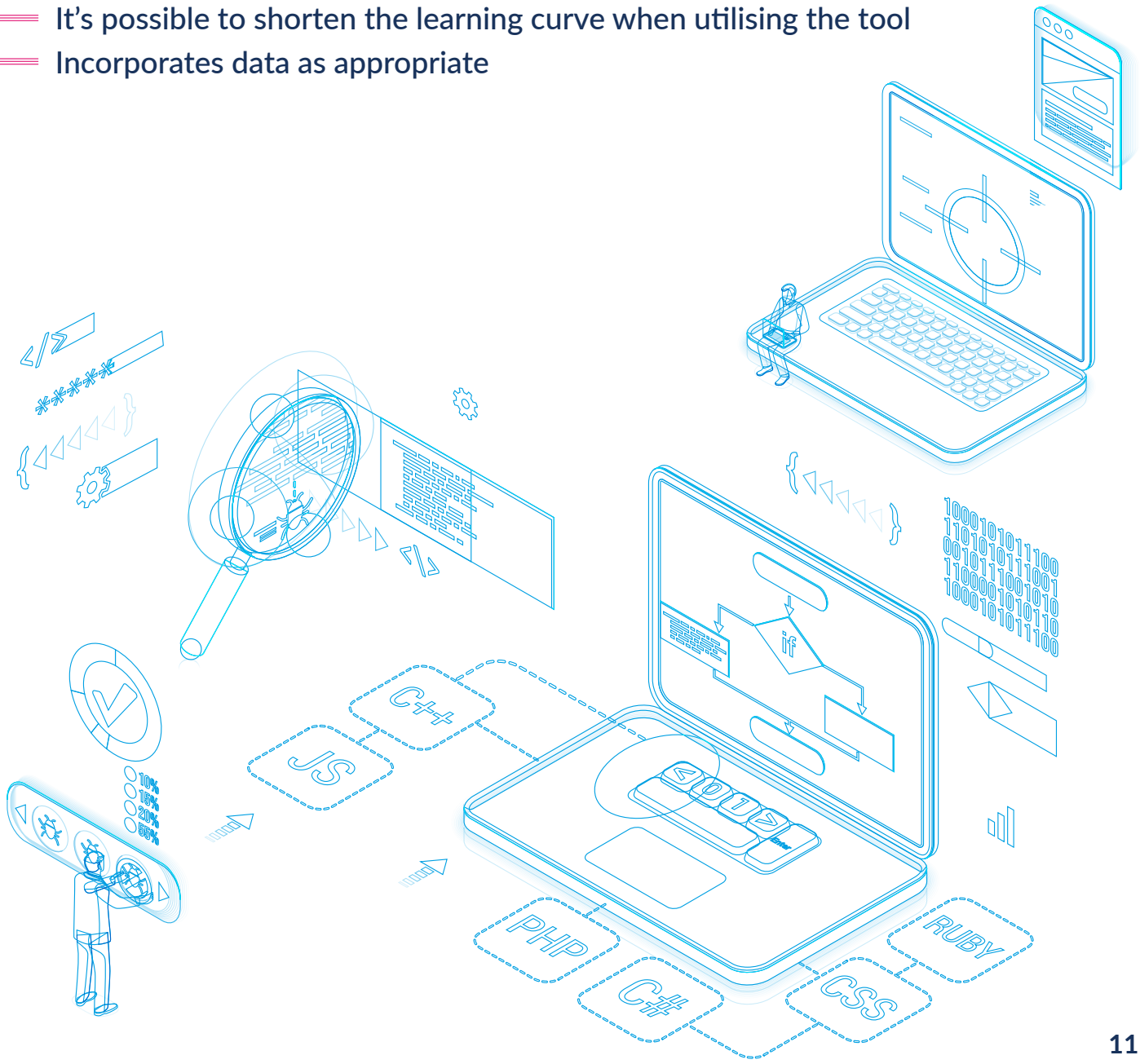
5. Monitoring and maintenance

Monitoring and maintenance are used to determine whether newly introduced software functionalities are functioning properly or not. When new automation scripts are implemented, they need to be reviewed and maintained to increase their effectiveness with each subsequent release cycle. This is done during maintenance in automation testing.

A Framework for Test Automation

A framework is a set of automation standards that assist in:

- Consistency of testing being upheld
- Enhances test structuring
- Minimum amount of coding being used
- Less code maintenance
- Increasing reusability
- Non-technical testers may work with code
- It's possible to shorten the learning curve when utilising the tool
- Incorporates data as appropriate



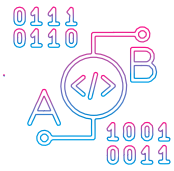
Types of Automation Tests

We really couldn't fit all the many test types—many of which can be automated—into one post. But these are sufficient to serve as a solid foundation.

1. Analysing the code
2. Unit tests
3. Integration tests
4. Automated acceptance tests
5. Regression tests
6. Performance tests
7. Smoke tests

Analysing the Code

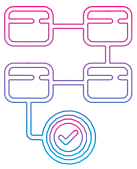
Code analysis tools come in a wide variety of forms, including static and dynamic analysis. These tests range from style and form checks to security vulnerability detection. When a developer checks in new code, these tests are executed. There isn't much test authoring required for these automated tests other than setting up the criteria and keeping the tools current.



Integration Tests

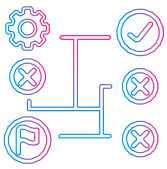
The necessity to interface with external dependencies makes integration tests, also known as end-to-end tests, more difficult to set up. Often, especially when dealing with resources that are out of your control, it is best to construct fictitious external resources.

If the vendor's web service is unavailable, for instance, and your logistics app depends on it, your test may unexpectedly fail. Does this imply that your app is unusable? Although you ought to have sufficient control over the complete testing environment to directly build each scenario, it might. Never allow an outside factor to influence the result of your test case



Unit Tests

A unit test suite can also be automated. Unit tests are made to test a single process, or function, separately. Usually, they function on a build server. Databases, third-party APIs, or file storage are not required for these tests. They should test the code solely, not the external dependencies, and they must be quick.



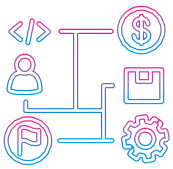
Automated Acceptance Tests

Automated acceptance testing (AAT) is used in a variety of ways today, but they all essentially accomplish the same goal. Automated acceptance test-driven development (AATDD) and behaviour-driven development (BDD) are related concepts. They both adhere to the same procedure, which is to write the acceptance test before working on the feature.

In the end, the automated acceptance test is run to see if the feature fulfils the agreed-upon requirements. To write these tests effectively, developers, the business, and QA must collaborate. They guarantee that the feature performs as planned and acts as regression tests in the future.

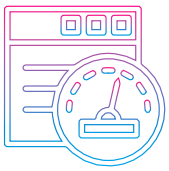
Regression Tests

You must create regression tests after the fact if AATs are not in place. Although they are both types of functional tests, the way they are written, when they are written, and who writes them all vary greatly. They can be controlled by an API by code or a UI, just like AATs. These tests can be written using tools that include a GUI.



Smoket Tests

A smoke test is a simple test that's typically carried out following a maintenance or deployment schedule. It is designed to confirm that all of the services and dependencies are operational. It should not be considered a functional test, and it can be started manually or as part of an automated deployment.



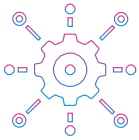
Performance Tests

There are many different types of performance tests, but they all focus on a specific component of an application's performance. Will it withstand high pressure? Are we checking for high heat in the system? Is it just response time under load that we are looking for? Consider scalability.

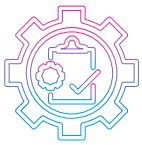
These tests occasionally call for simulating a large number of users. In this situation, it's crucial to have a setting that can pull off such a feat. This type of testing can be assisted by cloud resources, but on-premises resources can also be used.

Best Practices in Test Automation

Keep the following in mind to maximise your ROI from automation:



Before the project begins, the **Automation scope** must be precisely specified. This correctly sets the bar for Automation.



Choose the appropriate **automation tool:** A tool must suit the automation criteria, not only be popular.

Pick an acceptable **structure.**

Scripting Guidelines:
When writing scripts for automation, guidelines must be observed. A few of them include:

- Create consistent scripts, comments, and code indentation.
- Adequate exception handling refers to how errors are handled when a system fails or an application behaves unexpectedly.
- For error logging, user-defined messages must be codified or standardised so that testers can comprehend them.

KPIs to track:
The effectiveness of automation cannot be assessed by contrasting manual and automated efforts alone, but also by monitoring the metrics listed below.

- Percentage of faults discovered
- The amount of time needed for automation testing during each release cycle
- The release takes the shortest possible time
- Customer satisfaction score
- An increase in productivity

If followed, the aforementioned recommendations can substantially aid in the effectiveness of your automation.

Choosing an Automation Tool

Finding the ideal tool might be challenging. The criteria listed below will assist you in choosing the right tool for your needs:

- Environment assistance
- User-friendliness
- Database testing ability
- Identifying objects
- Testing images
- Testing error recoveries
- Mapping objects
- Using scripting languages
- Ability to support various types of testing
- Ability to support multiple testing frameworks
- Ease of use of debugging automation software scripts
- Can it recognise objects across any environment?
- Comprehensive testing reports
- Low cost of training for tools selected

Before implementing automation, one of the main hurdles to be overcome is tool selection. Prior to moving forward with a proof of concept, identify the needs, investigate potential tools and their capabilities, and establish your expectations for the tool.

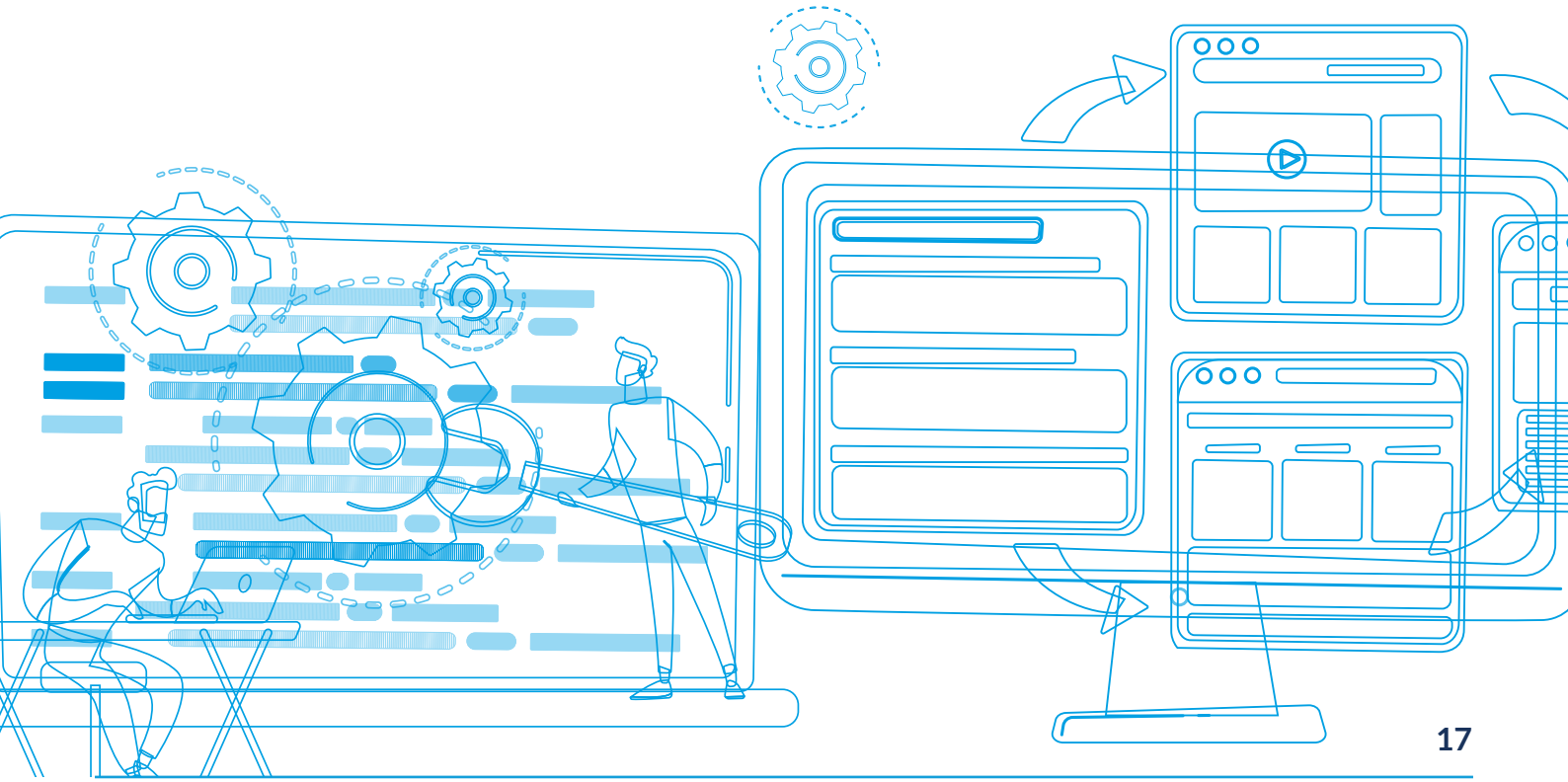
How Organisations Can Leverage Test Automation?

Automation should be used by businesses, especially those in the technology sector, to enhance their operating systems and business processes. Businesses can take advantage of the useful tools that automation offers, whether they want to speed up product delivery or fulfil higher security standards.

Once you've decided on the test, you need to define objectives as a benchmark to gauge how well it does. It will be challenging for you to fully utilise the test result if you don't create goals. Keep your attention on this single goal, and don't be hesitant to do additional tests if necessary. Think about your goals and how this test can assist you in achieving them.

Create logically smaller tests from your larger tests. Larger, more intricate tests are more challenging to run. To make better use of their time, team members who aren't producing test code can be transferred to other stages of the product development cycle.

Making testing simpler and enhancing corporate procedures are the main goals of automation.



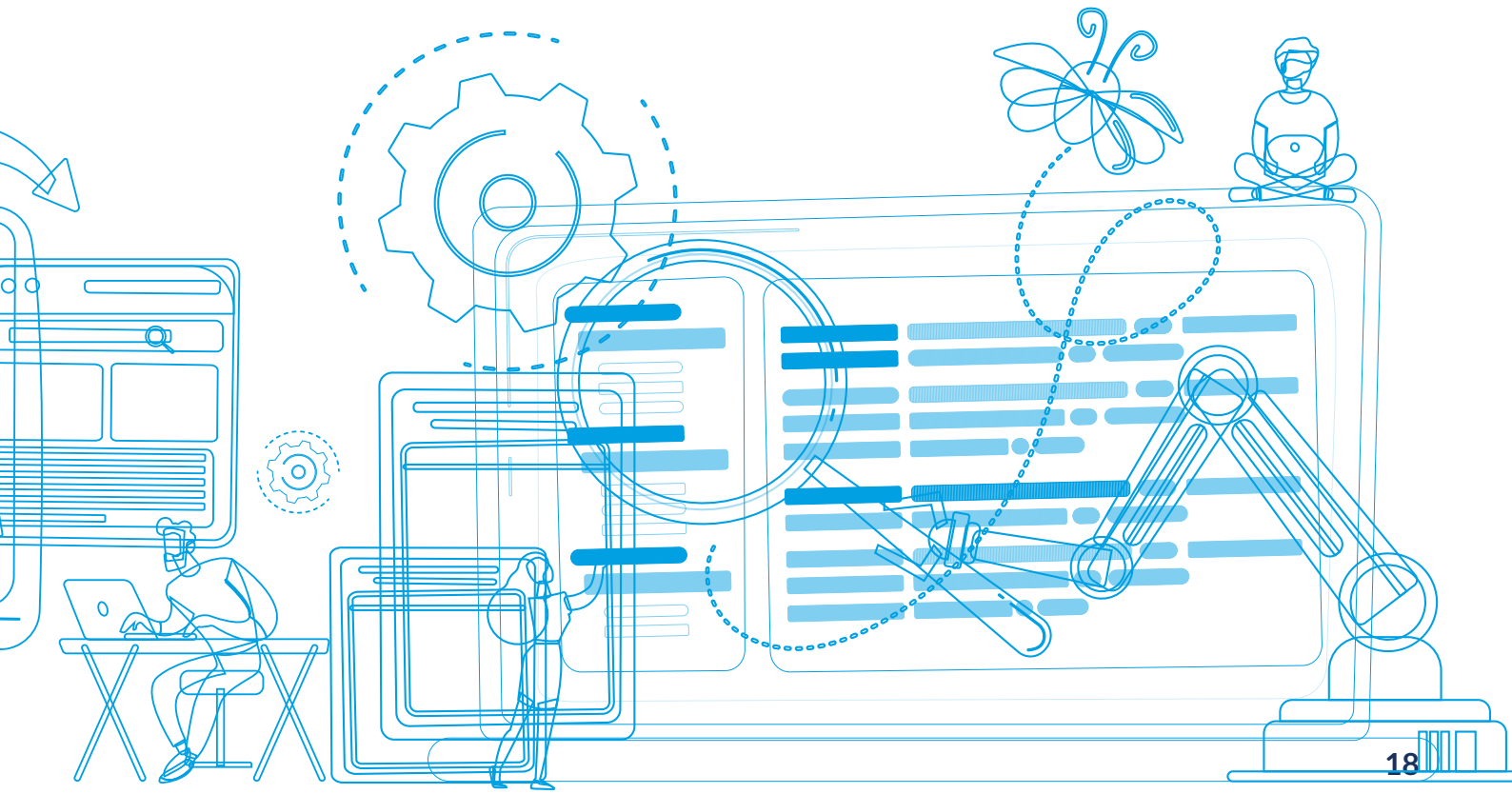
Conclusion

The way you run your company might be significantly altered by automation. The time has come to adopt new technology and develop approaches that will improve the effectiveness and efficiency of the workday. Don't be scared to experiment with a few different tools to determine which one suits you the best. Because every firm is unique, so are the goods you are developing or the software you use.

The following steps are key:

- Choosing the test tool
- Determining the automation scope
- Strategy and preparation
- Execution of test automation
- Ongoing maintenance

The greatest approach to make sure your company stays on top of debugging, flaws, and issues that can easily sour the production process if not straightened out as soon as possible, is to use automated testing.



Get in touch with Codification to start your journey toward Test Automation

Visit our website to learn more: www.codification.io/services

About Codification

Codification is a Cloud Native transformation consultancy, with a team of over 100 engineers, consultants and business professionals distributed across the world. We were founded in 2019 in the United Kingdom. We have grown since then to have a presence in Europe, the Middle East and Asia, serving leading multinational corporations, government institutions, global banks, and industry giants with our consultancy and expertise.

Through our experience, we have noticed that visionary leaders want to transform their organisations into technology companies to thrive in the new digital-first economy. Here, businesses want to release software faster, improve quality and build a continuous improvement culture where the best ideas win. At Codification, we establish the direction of a company's technological transformation journey and help implement new technologies and processes, resulting in a modernised digital-ready organisation.

CODIFICATION

Codification United Kingdom
The Core
Bath Lane
Newcastle upon Tyne
NE4 5TF

Phone: +44 01670 223994

Web: www.codification.io/