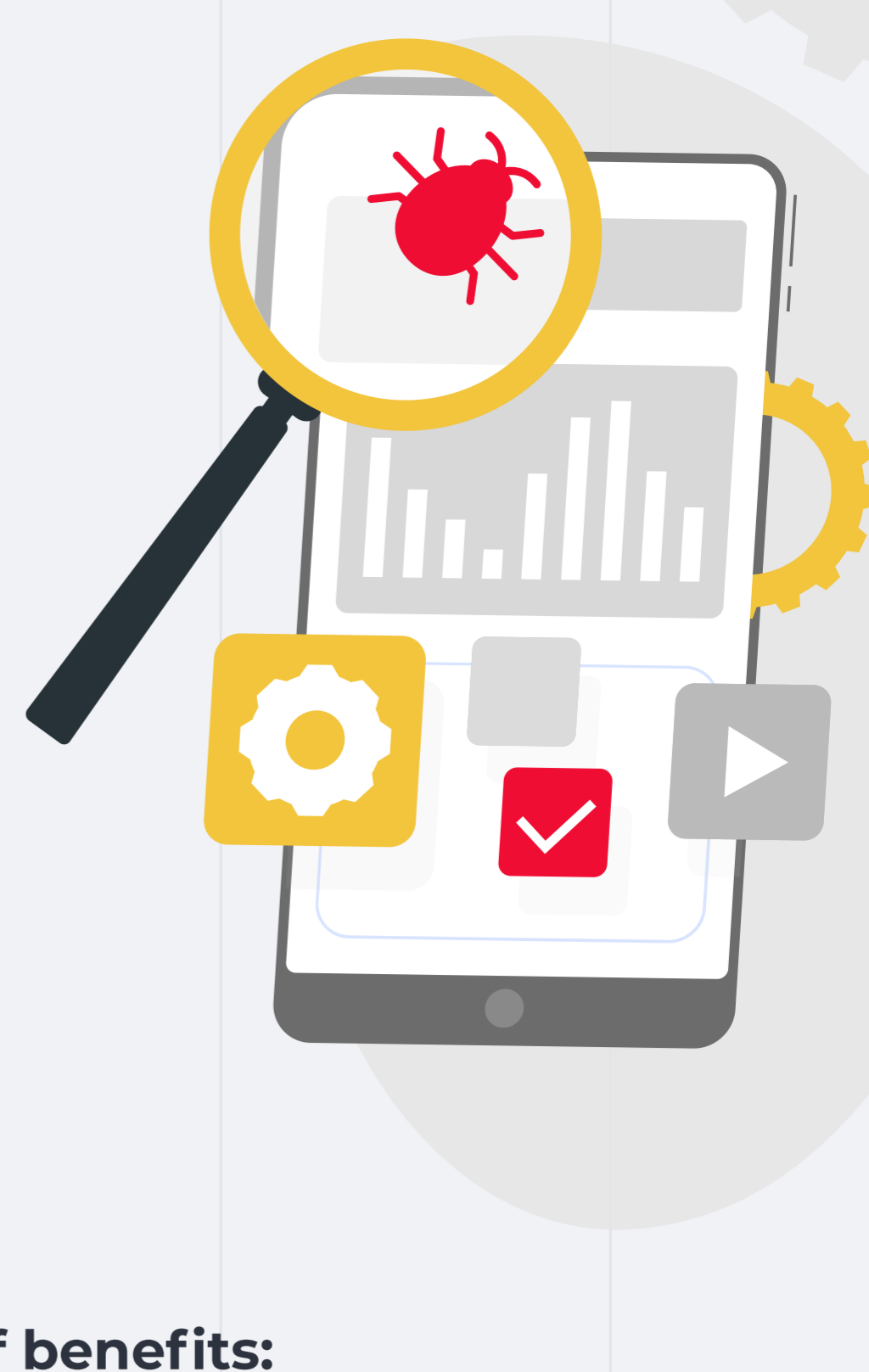


Software Automated Testing vs Manual Testing

Software testing is the process of verifying that software products not only fulfill the purposes they have been designed for but are also error-free. Software testing includes the analysis, observation, examination, and evaluation of multiple aspects of a product before it is launched to the market.



Testing software comes with a series of benefits:

<ul style="list-style-type: none"> Ensures quality and performance 	<ul style="list-style-type: none"> Reduces development costs
<ul style="list-style-type: none"> Reduces the risk of failure 	<ul style="list-style-type: none"> Prevent bugs
<ul style="list-style-type: none"> Prevents vulnerability 	<ul style="list-style-type: none"> Facilitates the addition of new features
<ul style="list-style-type: none"> Enhances customer satisfaction 	<ul style="list-style-type: none"> Improves user experience
<ul style="list-style-type: none"> Identifies errors and/or missing requirements at an early stage 	<ul style="list-style-type: none"> Detects compatibility with various platforms and devices
<ul style="list-style-type: none"> May prevent catastrophic incidents (e.g. system blackouts, communication breakdowns, etc.) 	

Software testing can be categorized according to multiple criteria. Here are the main types of software testing:

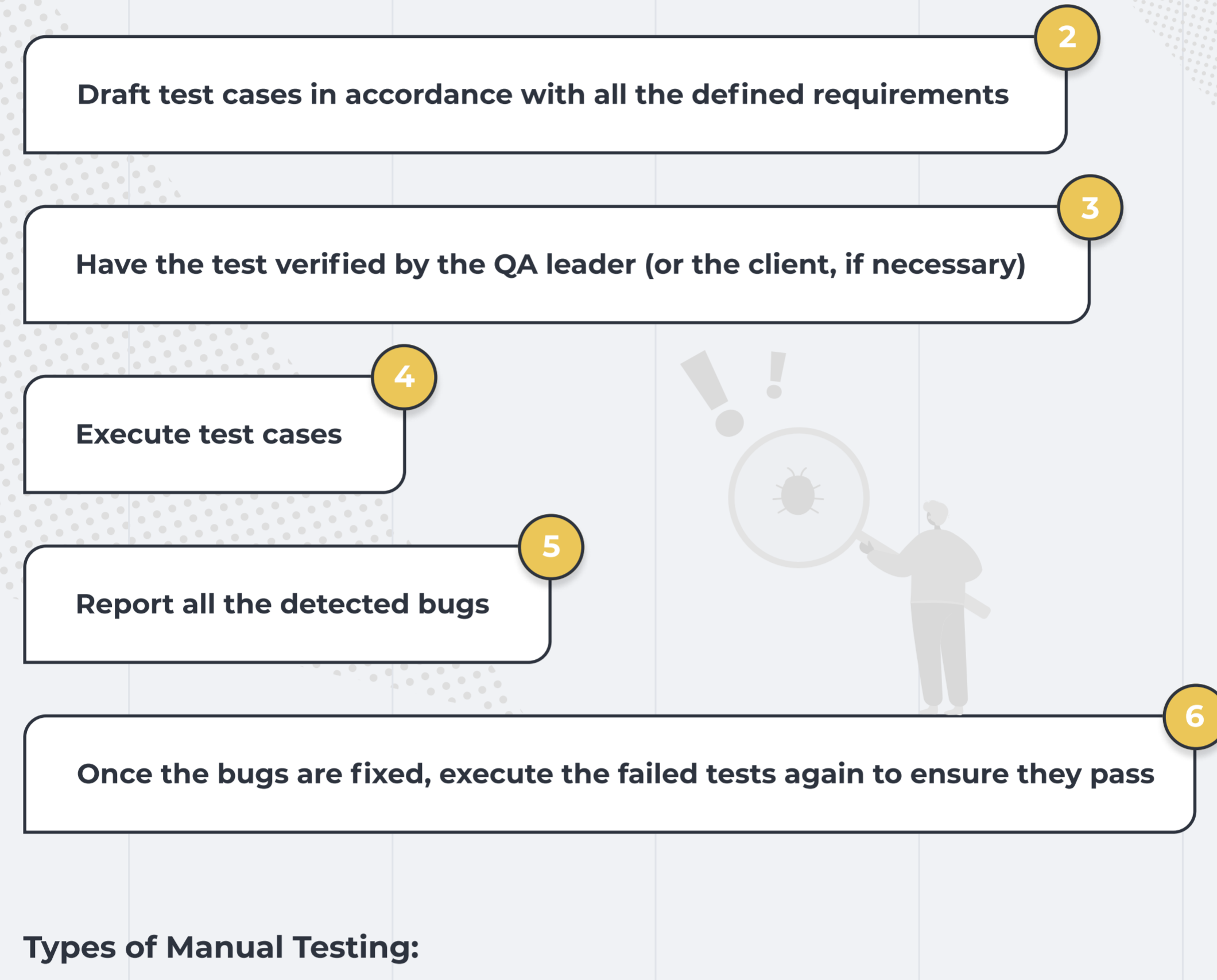
- Type:** manual, automated
- Functional testing** (includes system testing, beta testing, unit testing, integration testing, etc.)
- Non-functional testing** (includes reliability testing, performance testing, security testing, usability testing, etc.)

Manual Testing vs. Automated Testing

Manual testing

Manual testing is a process in which testers manually perform different test cases without using automated tools. The key purpose of manual testing is to make sure that the product meets the specified functional requirements and is error-free. Apart from an extensive skills set, performing manual testing also requires substantial patience and creativity.

Steps to perform Manual Testing:



Types of Manual Testing:

White Box

White Box: the QA is familiar with the structure or internal code of the app. White Box is used mainly for unit testing and other specific techniques such as path testing, control flow testing, flow testing, decision coverage, etc.

Black Box

Black Box: the QA is not familiar with the structure or internal code of the app and interacts with it from an end-user perspective to test its functional and non-functional features.

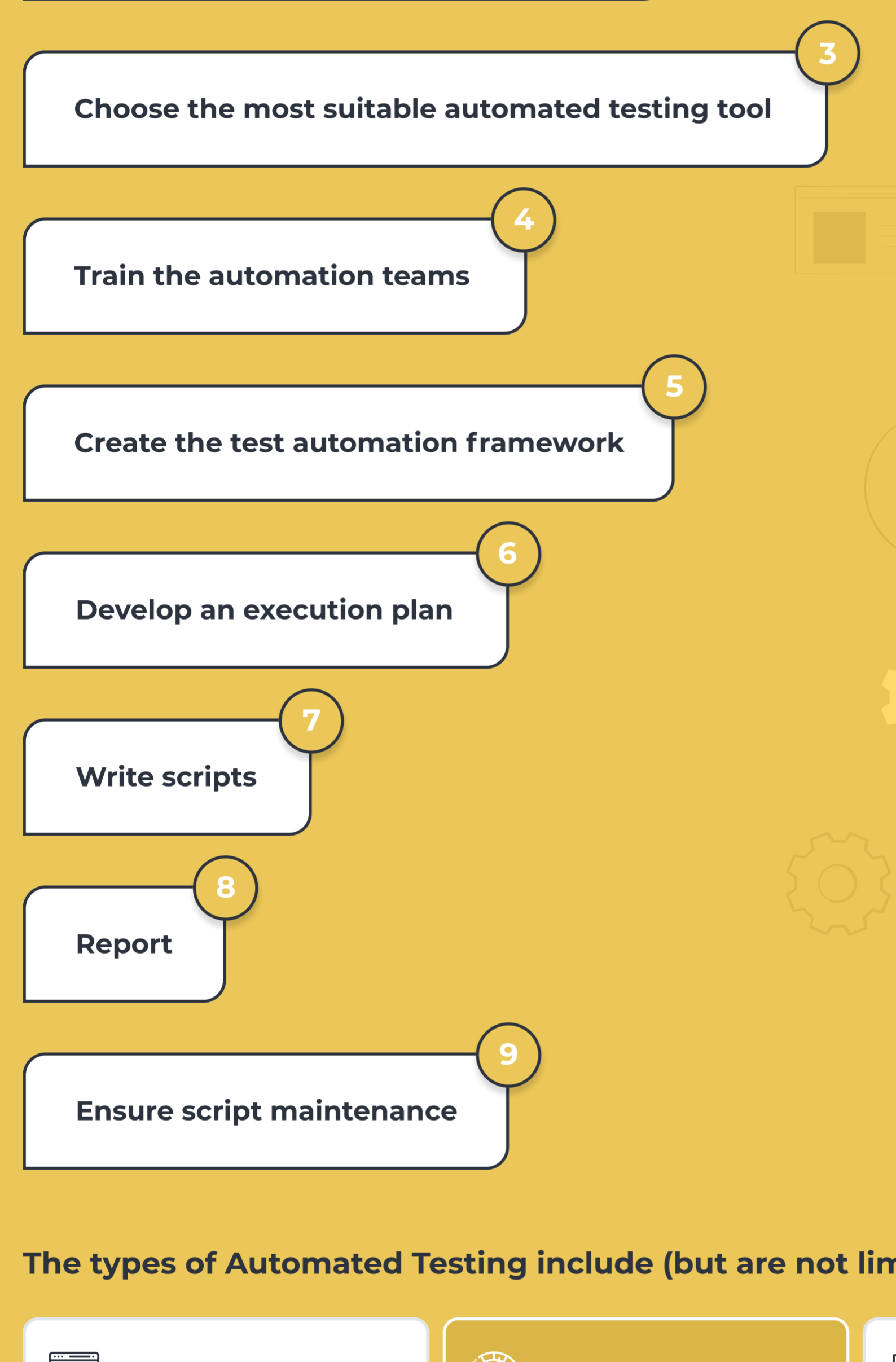
Grey Box

Grey Box: this approach combines white box and black box techniques to identify bugs caused by structural flaws or inappropriate usage.

Automated Testing

Automated testing is a software testing technique used to test and contrast the expected outcome with the actual one. Testers write test scripts and use test automation tools to execute them, thus making testing faster and reducing the risk of human errors.

Steps to perform Automated Testing:



The types of Automated Testing include (but are not limited to):

- UI Tests
- Performance Tests
- Security Tests
- Integration Tests
- API Testing
- Unit Testing

Tools for Manual Testing

Even though automated testing is slowly becoming indispensable, manual testing is still required to ensure that new builds can be further tested and are suitable for automation.

Depending on the covered area, there are various tools that can improve manual testing:

- Issues trackers** (e.g. Basecamp, Asana, Trello, etc.)
- Testing Case Managers** (e.g. Zephyr, TestLodge, etc.)
- API Testers** (e.g. REST Assured, SoapUI, etc.)
- Cross-Browser Testers** (e.g. Browser Shots, Sauce Labs, etc.)
- Email Testers** (e.g. Mailtrap, Litmus, etc.)
- Screen Capturers** (e.g. Skitch, CloudApp, Loom, etc.)

Tools for Automated Testing

While automated testing does not completely discard manual testing, it does streamline the testing process.

Examples of automated testing tools include:

- Test Automation Frameworks** (e.g. Selenium, Cypress, Robot, Appium, Citrus, Carina, etc.)
- AI-powered automation** (e.g. Eggplant, Kobiton, Mabl, TestCraft, ZeuZ, etc.)
- Cross Browser Testers** (e.g. LambdaTest, Browsera, Ranorex Studio, TestComplete, etc.)
- API testers** (e.g. Assertible, API Fortress, Testim, Katalon, etc.)
- GUI testers** (e.g. Rapise, Autot, Maveryx, RIATest, etc.)
- Defect Trackers** (e.g. UFT, TestinWhiz, Tricentis Tosca, etc.)

Manual Testing Benefits

- Ensures a more accurate UI testing
- Certain features are more suitable for manual testing (e.g. usability, app crashes, user experience scenarios, etc.)
- Cost-effective and fast for small projects
- Easy to adapt to UI changes
- Minor issues can be analysed immediately without the need to write a code and execute it
- No environment limitations
- Allows for human observation (i.e., testers may identify defects and provide development suggestions)
- No need for programming

Manual Testing Pitfalls

- Tasks are completed entirely manually
- Time-consuming
- Boring, repetitive, and prone to human error
- Difficult to ensure complete test coverage
- Requires skilled human resources
- High costs for large projects
- May not detect all the defects
- It is not reusable
- Not feasible for Performance Testing

Automated Testing Benefits

- Ensures enhanced test coverage
- More efficient and less time consuming than manual testing
- Eliminates the risk of human errors
- Scalable for large projects
- Ideal for repeated and large-volume testing
- Runs multiple tests simultaneously
- Accurate and reliable
- Reusable
- Provides quick feedback
- Programmable
- Frees up testers' time

Automated Testing Pitfalls

- Costly and slow for small projects
- Lack of flexibility and creativity
- Unable to test user experience scenarios
- More difficult to adapt to UI changes
- Requires maintenance (effort and time)
- May produce false negatives
- Not suitable for all testing types (e.g. visual testing, usability, etc.)
- Does not allow random testing
- Programming knowledge is mandatory
- Debugging test scripts is a major problem

Wrap up

Manual and Automated Testing are not only interdependent but they both come with a series of pros and cons. However, to highlight the obvious, thorough software testing before it goes live is the only way to ensure performance, dependability, security, customer satisfaction, cost-effectiveness, and – ultimately – an increased ROI.