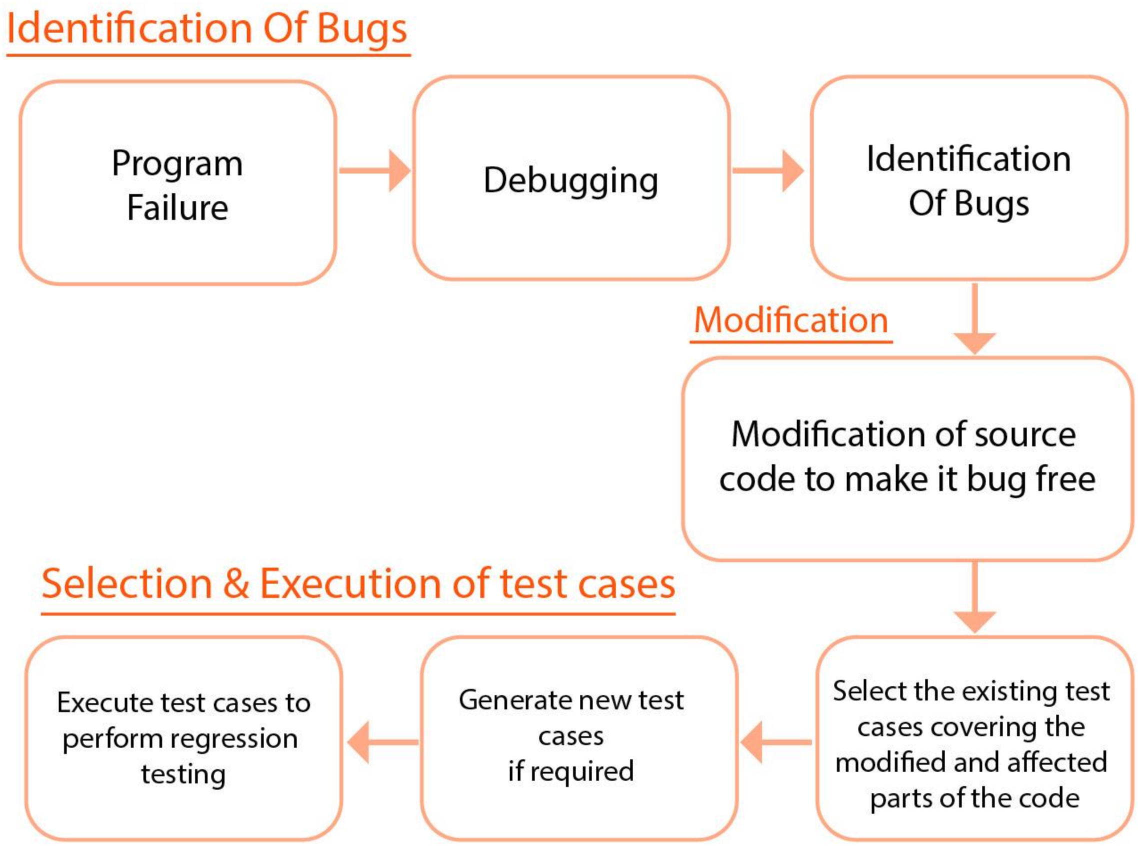
Regression testing verifies that no new mistakes have been added to the programme after the adjustments have been made by testing the modified portions of the code and the portions that may be impacted due to the alterations. Regression is the term for anything coming back, and in the context of software, it refers to a defect.

When should you do regression tests?

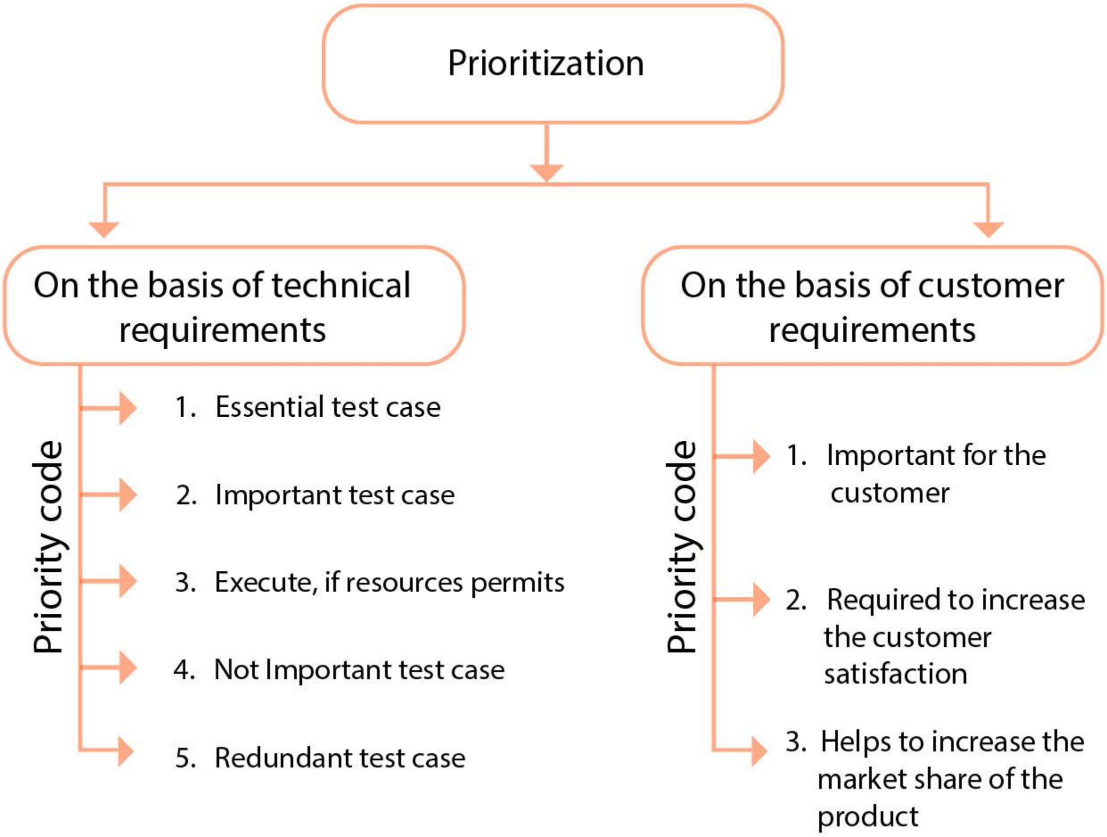
1. When a new feature is added to the system and the code is changed to accommodate and incorporate that feature with the current code.
2. When a software flaw has been found and is being fixed by debugging the code.
3. When a code change is made to improve performance.

Regression testing procedure:

Initially, anytime we make changes to the source code for whatever reason—such as adding new functionality or optimising existing code—our software fails in the previously created test suite for apparent reasons when it is run. After the failure, the source code is debugged to find the program's faults. The necessary changes are done after finding the problems in the source code. Then suitable test cases are chosen from the test suite that already exists and covers all the updated and impacted portions of the source code. If more test cases are needed, we can add them. Regression testing is ultimately carried out utilising the chosen test cases.


Identification Of Bugs
Program Failure
Debugging
Identification Of Bugs
Modification
Modification of source code to make it bug free
Selection & Execution of test cases
Select the existing test cases covering the modified and affected parts of the code
Generate new test cases if required
Execute test cases to perform regression testing

- All test cases are chosen with in this manner from the test suite that is already in place. Although it is the simplest and safest method, it is not very effective.
- Randomly choose test cases: This strategy involves choosing test cases at random from the test suite already in place; however, it is only effective when every test case has an equal capacity to identify faults, which is extremely uncommon. Because of this, it is rarely used.
- Choose test cases that cover and test the updated portions of the source code and the parts that are affected by these modifications are chosen in this approach.
- **Pick higher priority test cases:** In this method, each test case in the test suite is given a priority code based on its capacity to identify bugs, client requirements, etc. The test cases with the greatest priorities are chosen for regression testing after giving the priority codes. The test case with the highest priority is ranked first. A test case with priority code 2 is less significant than one with priority code 1, for instance.


Prioritization
On the basis of technical requirements
On the basis of customer requirements
Priority code
1. Essential test case
2. Important test case
3. Execute, if resources permits
4. Not Important test case
5. Redundant test case
Priority code
1. Important for the customer
2. Required to increase the customer satisfaction
3. Helps to increase the market share of the product

Tools for regression testing: In regression testing, we often choose test cases from the current test suite itself, ; so, we don't need to compute their expected outcome and it can thus be readily automated. Automating the regression testing process will be extremely effective and time saving.

The following are the most often used regression testing tools:

- Selenium
- WATIR (Web Application Testing In Ruby)
- QTP (Quick Test Professional)
- RFT (Rational Functional Tester)
- Winrunner

WATIR (Web Application Testing In Ruby)

- A Ruby programming language-based open-source package, Watir stands for Web Application Testing in Ruby. It allows for the creation ofto create and maintain simple-to-read and maintain tests on a light and adaptable user interface.

- Watir enables a range of user interaction features for website testing, including the ability to click links, complete forms and validate texts across a number of browsers.
- Features of the Tool:
- Extremely portable and user-friendly gadget
- Excellent browser interaction features are included with this programme.
- Designed to test web applications.
  Enables the creation of understandable, maintainable, and easy automated tests.
- Support for cross-platform technologies
  Numerous large corporations, like SAP, Oracle, Facebook, etc., use it
- Technology independent

QTP (Quick Test Professional)

It is an automated functional testing tool that consumes high resources and requires a license. QTP provides customer support in the form of selenium community forums. For parameterization in QTP, built-in tools are available. QTP supports only Windows and VB Script. There is support for testing on both web and desktop-based applications. There is built-in test report generation within the tool QTP, and there is a built-in object repository in QTP. QTP is user-friendly, and there is a build-in recovery scenario in QTP. Browsers supported in QTP are specific versions of Google Chrome, Mozilla Firefox, and Internet Explorer.

RFT (Rational Functional Tester)

An object-oriented automated functional testing tool called Rational Functional Tester can run automated functional, regression, GUI, and data-driven tests. HTML, Java,.NET, Windows, Eclipse, SAP, Siebel, Flex, Silverlight, Visual Basic, Dojo, GET, and PowerBuilder programmes are just a few of the many applications and protocols that RFT supports.

Benefits

The following are the primary advantages of Rational Functional Tester:

- Reusability: Tests may be immediately performed on several iterations of an application, cutting down on the time required for regression testing.
- Consistency: The exact same actions will be taken each time an RFT script-based test is executed.
- Productivity: Automated testing is quick and flexible, requiring no additional resources.

- Rational Team Concert and Rational Clear Case are two source control management technologies that RFT interfaces with. Users may manage their RFT functional test assets using either Rational Clear Case or Rational Team Concert by integrating RFT with these source control management programs.
- RFT and Rational Quality Manager have excellent integration. Users may run test scripts from within Rational Quality Manager after integrating RFT with RQM using the adaptor.
- Features
- Broad skills match: The RFT tool is designed for users with a range of technical skills to ensure that your quality assurance team is not limited to basic testing and that other subject matter experts in your company can participate in and comprehend the test flow using a visual storyboard format.
- New software versions can employ user interface features that technology has learned, reducing time spent writing new test scripts.
- Automated scripts – Rational Functional Tester gives your development teams the ability to write scripts with keyword associations that are simple to reuse, increasing productivity.
- Using the Eclipse Java Developer Toolkit editor, your team may quickly and easily develop Java test scripts. It includes sophisticated debugging features and automates code completion

Winrunner

- For functional testing, Win Runner is a popular automated software testing tool. Mercury Interactive created it. C and web technologies including VB, VC++, D2K, Java, HTML, Power Builder, Delphe, and Cibell (ERP) are supported. WinRunner makes it simple to create tests by capturing your application development process. You may navigate your application’s GUI (Graphical User Interface) elements by pointing and clicking.
- A test script written in the C-like Test Script Language is generated by WinRunner. With hand programming, we can improve our test scripts even further. The Function Generator, which is part of WinRunner, makes it simple and quick to add functions to our recorded tests.
- The crucial features of WinRunner include:
- We are able to do functional and regression testing on a range of application software created in languages, including PowerBuilder, Visual Basic, C/C++, and Java. Additionally, we are able to test ERP/CRM software programs.
- Runs tests on many browser settings, including Internet Explorer and Netscape Navigator, as well as all versions of the Windows operating system.
- In the “record” mode, we may record GUI operations. A test script is automatically generated by WinRunner.
  To compare actual and anticipated outcomes, we can add checkpoints. Bitmap

checkpoints, GUI checkpoints, and web links are some examples of the checkpoints.

- It offers a tool for test case synchronization.
  A recorded test may be transformed into a data-driven test using Data Driver Wizard. So, within a test script, we may swap out data for variables.
- During automated testing, database checkpoints are utilised to validate data in a database. We will maintain database integrity and transaction correctness by highlighting the records that are added, removed, changed, or updated.
- WinRunner is taught to detect, record, and replay custom objects using the Virtual Object Wizard.
- The reporting tools offer the ability to create test results automatically and examine flaws.
- Many testing-related tasks may be automated by integrating WinRunner with the testing management programme, TestDirector.

Silktest

Silk Test is a solution for corporate application regression and function testing. It was first created by Segue Software, which Borland purchased in 2006. Micro Focus International purchased Borland in 2009. QA Partner was the original name of the product from 1993 to 1996.

Silk Test provides a range of clients:

- Silk Test Workbench supports visual automated testing (much like the previous TestPartner) and uses the VB.Net scripting language.
- For automated scripting, Silk Test Classic makes use of the domain-specific 4Test language. Similar to C++, it is an object-oriented language. Classes, objects, and inheritance are all used.
- UFT Developer, formerly known as Silk4J, enables automation in Eclipse using Java as the scripting language.
- UFT Developer, formerly Silk4Net, enables the same functionality in Visual Studio using VB or C#.

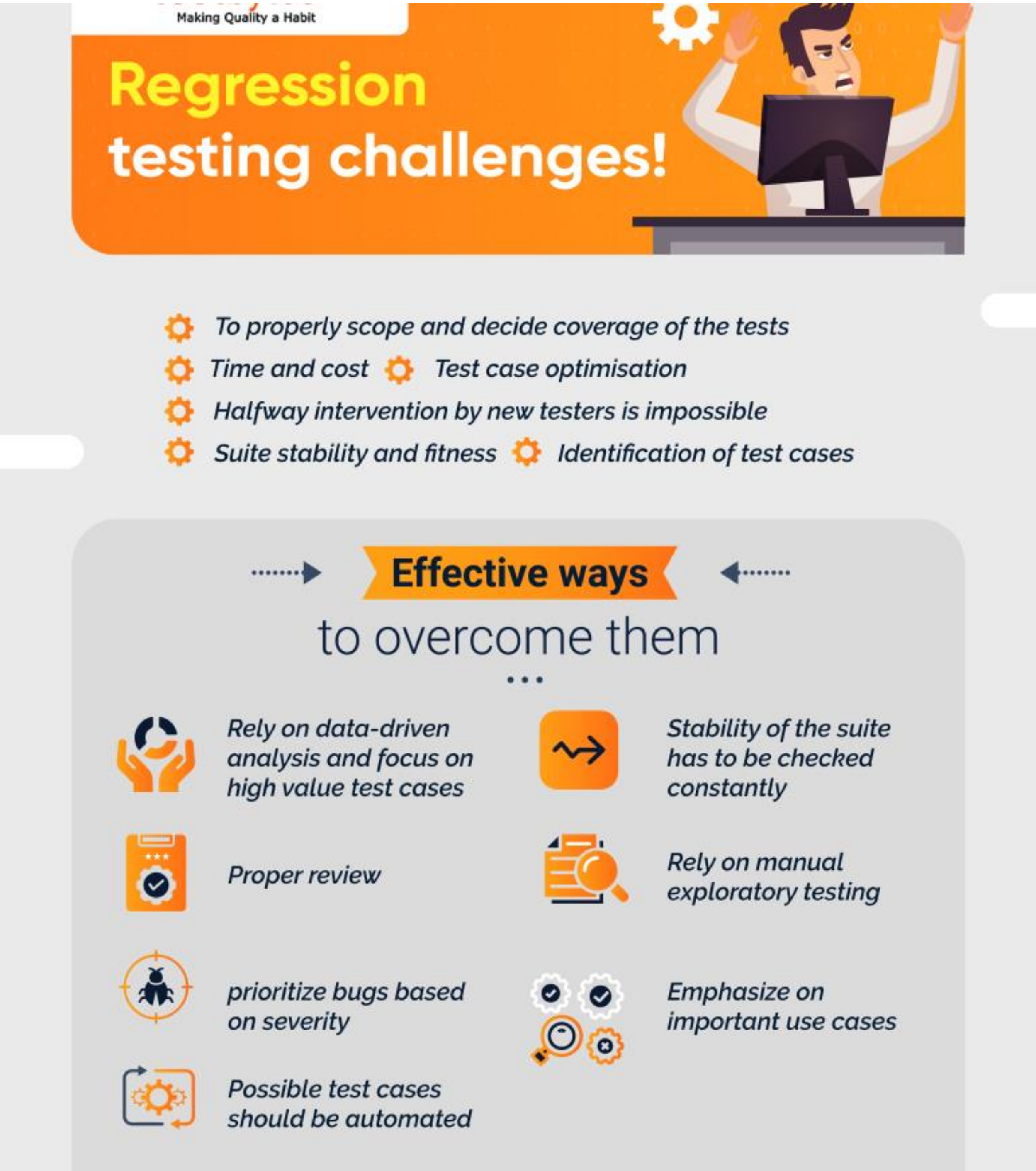
Benefits of Automation using QTP

- It allows for recording and replay.
- It allows testers refer to the screen object attributes when recording scripts on an active screen.
- It has a great system or procedure for identifying objects.
- It supports a variety of add-ins, including those from PeopleSoft, Oracle, Java, SAP, NET, and others.
- Through an active screen, you may improve the current tests even without the AUT.

- It supports well-known automation frameworks, including data-driven testing, modular testing, and keyword testing.
  It has an internal IDE.
- It can be connected with test management programmes like Winrunner, Quality Center, and Test Director.
- It is simple to manage several suite kinds, including Smoke, Regression, and Sanity.
- It works with XML.
- Through QTP, test reporting is possible for analytical purposes.
  Easily maintained.

Regression testing provides the following benefits:

- It makes sure that no new defects have been created after the system has received new functionality.
- Since the majority of the test cases chosen for regression testing are already part of the test suite, their anticipated results are known. As a result, automated tools may readily automate it.
- It aids in preserving the source code’s quality.
- The drawbacks of regression testing include:
- If automated tools are not employed, it may take a lot of time and resources.
- Even after relatively minor modifications to the code, it is necessary.



Making Quality a Habit

Regression testing challenges!

- To properly scope and decide coverage of the tests
- Time and cost
- Test case optimisation
- Halfway intervention by new testers is impossible
- Suite stability and fitness
- Identification of test cases

Effective ways to overcome them

- Rely on data-driven analysis and focus on high value test cases
- Stability of the suite has to be checked constantly
- Proper review
- Rely on manual exploratory testing
- prioritize bugs based on severity
- Emphasize on important use cases
- Possible test cases should be automated

Conclusion

An essential component of software development processes is test automation. Similar to this, automated regression testing is regarded as a crucial component.

Product teams may obtain more detailed feedback and react immediately with a speedy regression testing procedure. Regression testing finds new vulnerabilities early in the deployment cycle, saving firms the expense and work of fixing the accumulated flaws.

Apparently, a little change can occasionally have a cascading effect on the product's essential capabilities.

Because of this, developers and testers must not let any modification—no matter how small—that falls outside of their sphere of influence.
Functional tests don't take into account how new features and capabilities interact with the old ones; they merely look at how they behave. Therefore, determining the main cause and the product's architecture is more challenging and time-consuming without regression testing.