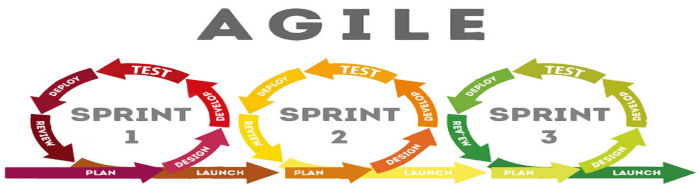
testsigma

Topics

REGRESSION TESTING

Iteration Regression Testing And Full Regression Testing in Agile

May 25, 2022 Pragya Yadav



Before we discuss the regression testing approaches in Agile: Iteration Regression Testing, and Full Regression Testing, let us explore Regression Testing a bit.

According to the **ISTQB** glossary, regression is “degradation in the quality of a component or system due to a change”.

And, Regression Testing is “a type of change-related testing to detect whether defects have been introduced or uncovered in unchanged areas of the software.”

Hence, whenever there is a change in the code, we need to ensure that the previously intact and working code continues to work and that the changed code has not impacted the working code in a negative way.

You may further check out a detailed guide that discusses **Regression Testing**.

Why do we need Regression Testing?

We need Regression Testing whenever there is:

1) Addition of new functionality in software
2) Code change due to changes in requirements
3) Any patch fix
4) Configuration changes of software
5) Code change for a Performance upgrade

testsigma

Regression testing approach in Waterfall methodology

When we are following the waterfall methodology, regression testing is run when the product is developed completely.

Since it is a waterfall approach, we wait until the development phase is complete before starting the new phase of testing. Typically we will have long testing release cycles while working in a waterfall approach.

The testing team will test the newly added functionalities in the code during these releases.

Next, regression testing is performed to ensure the old features are still working. This helps to ensure that the product is stable and is high in quality. There is a separate testing team who owns the testing activities. There is very little communication between the testing team and the developer's team.

In waterfall methodology, since the work is done in phases which are independent of each other, the release cycles are clearly longer and time-consuming.

Regression testing approach in Agile methodology

With the advent of DevOps and Agile approach, we aim to provide working software to the end-users as soon as possible, improving it continuously with customer feedback.

The result is smaller iterations or sprints where time is a constraint. Hence, a smaller regression pack is created which has the test cases related to the changes made in that particular sprint only.

This approach is called **Iteration Regression Testing** or **Partial Regression testing.**

Full Regression testing which has test cases covering the entire functionalities of the application is performed before a major release or delivery.

Continuous Integration (CI), frequent builds and code pushes by the developers call for an automated regression test run whenever there is a new build. In this case, the communication between testers and developers is imperative for continuous testing of the integrated software.

The collaboration and communication between testers and developers are needed for cross-training, faster feedback and quick turn-around time. The boundaries between the responsibilities of testers and developers have dimmed with the advent of **DevTestOps**.

Today, a developer can write automation cases for 'Continuous Testing', while tester works on parallel test-run perform some exploratory testing at the same time.

Read about **In-Sprint Test Automation at Agile and DevOps Speed Using Testsigma**.

Thus, bug-free software becomes the responsibility for everyone in the team and not just the testers' job.

Hence, in the Agile environment, there is a **two-level approach to Regression Testing**, which includes Iteration Regression Testing after every iteration or sprint and Full Regression testing before any major change is released

Implementation of Iteration and Full regression Testing

As we saw above, there could be two types of implementations of Regression Testing in Agile.

1) Iteration(Partial) Regression Testing

- Selection of the test cases is made based on the features developed or changed during that iteration. Test cases that could be impacted by the code-changes in that sprint are picked.
- To select the test cases, there are multiple approaches:

i. Risk-based prioritisation: Critical functionality with respect to users(most used and most visible) has the highest risk and highest priority in Iteration regression testing. Test cases are ranked and selected based on the priority of the functionality or area by the testing team.

The test cases can be assigned below priorities for selection in the Regression Suite, based on the **probability of failure** and **impact of failure**:

- ***High***: Business-critical functionalities of the application, most used and most visible functionalities, areas which are likely to fail.
- ***Medium***: Test cases which have identified defects in previous releases.
- ***Low***: Rest of the cases which will be added to the full regression test suite to fully cover the application testing.

According to the time allocated for testing, **High** and **Medium** priority test cases are selected for Iteration Regression Testing.

ii. **Collaborative selection:** The whole team – testers and developers, discuss and present their thoughts on areas where regression testing is needed.

With thorough discussion and understanding, a collaborative decision is made regarding the test case selection for the Iteration Regression test suite.
Read **how to prioritize test cases for regression testing**.

Features in a sprint are assigned in the order(rank) based on the testing priority using a card deck by the testing team members.

On the scrum board, team members provide testing order to the developed or changed features in the sprint as Priority 1, Priority 2 and so on.

Then, other team members suggest if the order seems fine, or if a change in the order is required. Ex. Testing team member 1 has mentioned the testing order based on priority as – feature 2, feature 3 and feature 1.

Testing team member 2 has also mentioned the same priority- feature 2, feature 3, feature 1.

But the developer 1 mentions the code change impacts feature 1 as well, and makes it more defect prone, hence the testing order should be – feature 1, feature 2 and feature 3.

This discussion helps in making a collaborative decision regarding the selection of the appropriate test cases.

As seen before, in **risk-based prioritisation approach**, only testing team makes the selection of test cases hence, a unified decision seems to be lacking in risk-based prioritisation approach, unlike the **collaborative selection approach**.

There are some automation tools that can benefit in the scenarios where iteration regression testing is needed. The tools help select test cases according to priority and areas and also can tell the test cases that could be impacted because of a change. One such tool is Testsigma.

2) Full Regression Testing

- The regression test suite fully covers all the areas or functionalities of the product.
- It is run before a major release or major change in code which will impact all the functionalities of the software.
- Since it contains test cases covering the whole software, it will usually have a huge size and will require time for completion.
- Selection of the test cases is made from the existing functional and non-functional test cases.

- Though it is time-consuming, it is still a reliable way to be sure that the application is tested and certified.
- Sometimes, even the customer demands a Full Regression test suite to be run to be assured that the application has gained the required stability.

Challenges faced while implementing regression testing in Agile

Adherence to the schedule is critical in the Agile approach with smaller sprints and frequent changes the testing team needs to work diligently to achieve the goals.

i. Regression suite size:

With each sprint, new features are added. This results in an increase in the size of the regression test suite every sprint. In the case of a big application, the number of test cases in the regression test suite can be huge, and the testing time required is directly proportional to the size of the test suite.

ii. Selection of test cases:

Selection of test cases during Iteration Regression Testing based on priority is a tedious task and requires expertise. Sometimes, the tester may not think from the perspective of the developer and may miss some testing which may pose a risk later.

iii. Frequent changes:

The word agile is synonymous with change and adaptability. Frequent and big changes in requirements are a part of Agile development. This leads to a change in test strategy and test cases. To accommodate everything in the sprint with the frequent changes becomes challenging for the testing team.

iv. Communication:

To conduct effective testing, the communication between testers, developers, product-owner, managers, business analysts and other stakeholders is vital. The new features, the changed features, obsolete features, everything needs to be discussed and then the test strategy should be followed.

v. Maintenance of test suite:

The maintenance of the regression test cases is a very important task. The changes should be updated, new test cases should be added and obsolete test cases should be removed periodically. During a sprint, management of the time required for the maintenance can be daunting for the testing team, but it is equally important for high-quality and effective testing.

What Functional test cases to choose as Regression test cases in Agile?

Usually, in the waterfall approach, a subset of Functional test cases is picked and the regression test suite is created. This approach is not effective in an Agile environment. The functions of the application are developed in iterations.

With each sprint, many changes would have taken place in a functional area until the final delivery. The functional test cases of a particular functional area in sprint 1, may require changes in sprint 2, because of adaptive and iterative nature of development.

testsigma

In each sprint, the overall stability of the application is checked via regression cases and hence the regression cases of a particular sprint should reflect the current sprint code and changes.

Best practices for Regression testing in Agile

i. Automated regression testing:

Due to the repetitive nature of the regression cases, they are ideal for automation. A parallel automated run of regression testing can help save time and improve quality.

You may check this article that discusses how to **select the right automated regression testing tool**.

ii. Scope of testing:

The scope of testing during iteration regression testing can be different from the scope of testing during full-regression testing. In iteration regression testing, a smaller scope can be under consideration for testing, whereas during full-regression just before the major release, the full scope covering all the functionalities will be considered for regression.

E.g. In sprint 1, if the application is released for Windows platform only, the scope of testing in strategy document is only for Windows and test cases are picked accordingly.

After many iterations, suppose a major release is planned where applications will be released on Windows, Mac, android platforms also. Here, the scope of testing for full-regression testing will include all the three platforms for testing.

Hence, we should clearly understand the scope of testing -whether it is for iteration regression testing or full regression testing otherwise we may end up losing time or end up testing features which were not meant to be for that sprint.

iii. Efficient regression test strategy:

Defining an efficient test strategy is very important for regression testing in Agile. The regression test strategy should contain a proper balance of below key points for effectiveness.

iv. Manual vs Automated:

Proper mix of manual and automated test cases should be selected. The test cases which are difficult to automate should be kept out of the regression pack because time is crucial in agile.

v. Iteration vs Full regression:

Based on the requirements and changes in the sprint, the iteration and full regression testing type should be selected.

vi. Maintenance of regression test suite:

Whenever there are changes in the functional requirements. Deletion of obsolete test cases is as important as the addition of new cases based on the development tasks taken up in the sprint.

vii. Continuous testing:

In Agile age, continuous testing is the backbone of **Quality Assurance**. **Fail fast and fail often** through continuous testing, so that defects are uncovered fast and fixed before delivery with minimum impact on customers.

testsigma

Return on investment is important and should be measured in Agile regression testing also. The measured ROI is compared with the ROI of the previous sprint(e.g. ROI is measured in terms of the time saved during test case execution after the test cases that were being executed manually were automated), to optimize the regression testing in the next iteration.

Conclusion

The Agile approach towards regression testing makes continuous testing imperative. Continuous Integration, Continuous Delivery and Continuous Deployment are all part of this game.

This enhances the dependency on automation for all these activities including the build generation after the code integration. The nightly build cycles are triggered by CI server and automated continuous regression testing is run.

Regression testing ensures the business continuity with minimum impact on customers and therefore cannot be skipped.

In each sprint, the whole team focuses on the functionality being developed in that particular sprint, and the regression tests take care of the stability of the overall application, including the code which was developed in previous sprints.

Hence, the importance of the **regression testing in an agile** environment is immense and requires due attention not only from the testing team but also from developers, product owners, business analysts and other stakeholders.

Suggested Reading:
Test Automation by Functional Experts and SMEs for Continuous Testing in Agile and DevOps

RELATED BLOGS

Top 20 SAP Testing Interview Questions and Answers

YAMINI PRIYA

REGRESSION TESTING

SAP Regression Testing | What it is & Why it Matters?

YAMINI PRIYA

AUTOMATION TESTING REGRESSION TESTING

How to Build an Effective Regression Test Suite

AARON THOMAS

testsigma

Start automating your tests now

Try Testsigma Get a Demo

Status
Careers
Terms & Conditions
Privacy Policy
Contact Us

Sales & Support

Product

What's New
Why Testsigma
Features
Integrations
Pricing

Community

Github
Discussions
Community Home
Discord
Add-ons Marketplace

Guides

Regression Testing
Codeless Testing
API Testing
Data Driven Testing
Automated Testing
Mobile Testing
Cross Browser Testing
Continuous Testing
Parallel Testing

Resources

Case Studies
Docs
Tutorials
Blogs
Webinars
Affiliate Program
Write For Us
Free Tools
Sitemap

Alternatives

TestProject Alternatives

testsigma

Appium Alternatives

Playwright Alternatives

Selenium Alternatives

Lambdatest Alternatives

Browersling Alternatives

Zephyr Alternatives

Top Features

Top Software Testing Tools

Platform Emulators