Webomates

Shift Left with Intelligent Test Automation

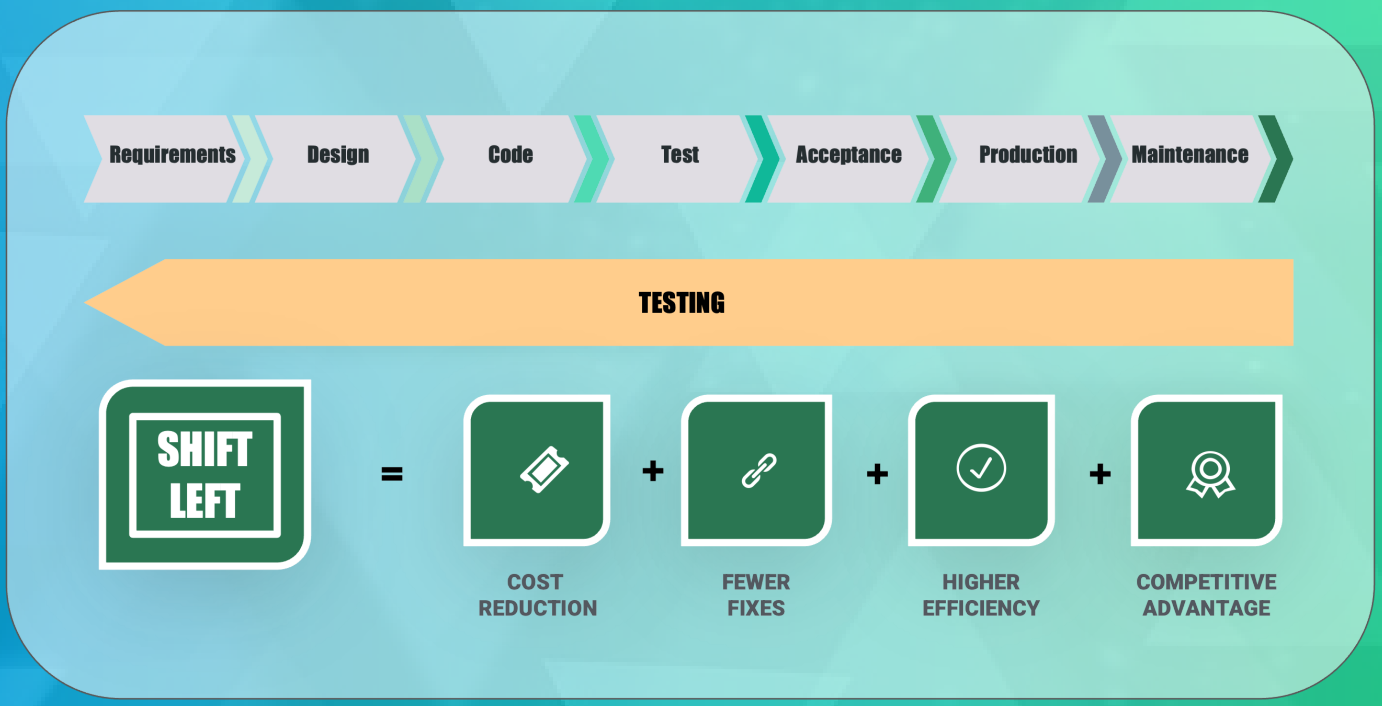
www.webomates.com | info@webomates.com

What is Shift-Left Testing?

A software project running out of time and budget is a situation we all go through. What could be the reason for this to happen? To one's surprise, the root cause may not be planning, more often than you think it is testing and bug fixes. The "Shift Left Testing" is a perfect solution to avoid delays and budget shortages.

We are all very aware of project deadlines, and in pressure to release the project on time or with the least delay, testing is often rushed, even at the risk of cutting corners. In the traditional waterfall approach or agile, software testing is performed at the end of the development cycle, which is very late in the project timeline. At the same time, the tester's input is received very late in the cycle.

Shift-left testing, we are testing early and much more frequently. The reason we are doing this? It is so that we can find defects earlier and fix them early before they go into the staging or integration or Beta. As we all know earlier, the bug is found lower the cost of the bug is. In short, with the shift-left, we are testing in the early stages, testing often to avoid finding defects in the later stages of the project.



Why Is It Called A Shift To The Left?

Why is this process called Shift Left testing? Because we are literally shifting the testing, wherein it is moving left and earlier in the software development life cycle. The shift-left testing does not mean just shifting the testing phase; it is performed as development progresses or in parallel to the development. Testing as the development team moves, so testing is performed every time a new change or feature is deployed in a build.

Shift left does not imply that there is no testing performed at the end of the project or the project's release. It is to be taken as that testing is performed at every step of the process flow and then at the end. It ensures that bugs are found in the early stages, and that final testing has minimal issues. It's about continuous testing and continuous feedback loop."

6 Phases of Software development Cycle

There are various benefits in moving to Shift Left testing. In this approach, developers can find defects early and often are given a chance to fix them earlier; doing this can reduce the time for the project to go live and help manage the project Budget. Based on the research by NIST, it is estimated that finding bugs in the production phase can cost over 30 times more than the original estimate, and if there are any security defects, it can go up by 60 times. But by shift-left testing, the final product is of higher quality as it has fewer patches and fixes, producing a stable product that is completed on time and within budget.

Advantages & Disadvantages of Shift Left Testing

Pros	Cons
Finding defects at the early stage	Increased infrastructure costs
Cutting down the cost by fixing bugs early	Requires human resources to accommodate test case healing
Top notch quality product as code contains very few codes fixes	Delivery of the software is delayed
The chances of the project going over the designated timeline is less	Improper testing – resulting in business losses
Higher Customer Satisfaction	
Higher test coverage	
Testing in parallel to development	

Pros:

Finding defects at the early stage: When shift left testing is done, bugs are found at the early stage of the application development, Which benefits in fixing the issues and bugs at an early stage of development rather than doing it at the end of the development of an application.

Cutting down the cost by fixing bugs early: Finding bugs at an early stage helps reduce the cost by avoiding retesting the whole feature after fixing the bug, which is found during traditional testing and delays the application delivery.

Advantages & Disadvantages of Shift Left Testing

Top-notch quality product as code contains very few code fixes: As the concept of shift left involves repeated testing of the application for all the builds, it's easier to identify any regression bugs that might be detected later in production. This, in turn, gives Higher quality products as we avoid code fixes and Patches.

The project's chances of going over the designated timeline are less: As traditional SDLC involves fixed timelines for each phase, detecting and fixing bugs in the testing phase can delay the product delivery. In shift-left testing, testing is done parallel with development bugs found fixed earlier, and the product is delivered on time.

Higher Customer Satisfaction: Since shift-left facilitates higher quality products with fewer patches and fixes, it provides higher customer satisfaction, and as the bugs are found and fixed in the development phase itself, it avoids the budget overflow.

Higher test coverage:

Testing in parallel to development: As mentioned previously in the above points, developing and testing in parallel helps in finding bugs and fixing them early, which in turn gives a higher quality product, customer satisfaction, delivery in time, and within budget.

Cons:

Increased infrastructure costs: Testing often involves increased use of resources. Testing earlier also requires parallel execution of the tests requiring more testing and cloud infrastructure.

Requires human resources to accommodate test case healing: Even after automating the test suites, shift-left involves the development and testing in parallel. Any newer changes would require updating test cases and test scripts - often called healing which needs resources to fix them. It also requires more resources to diagnose why automation scripts are not working. Scripts can fail due to automation system errors, script changes, and requirements change.

Advantages & Disadvantages of Shift Left Testing

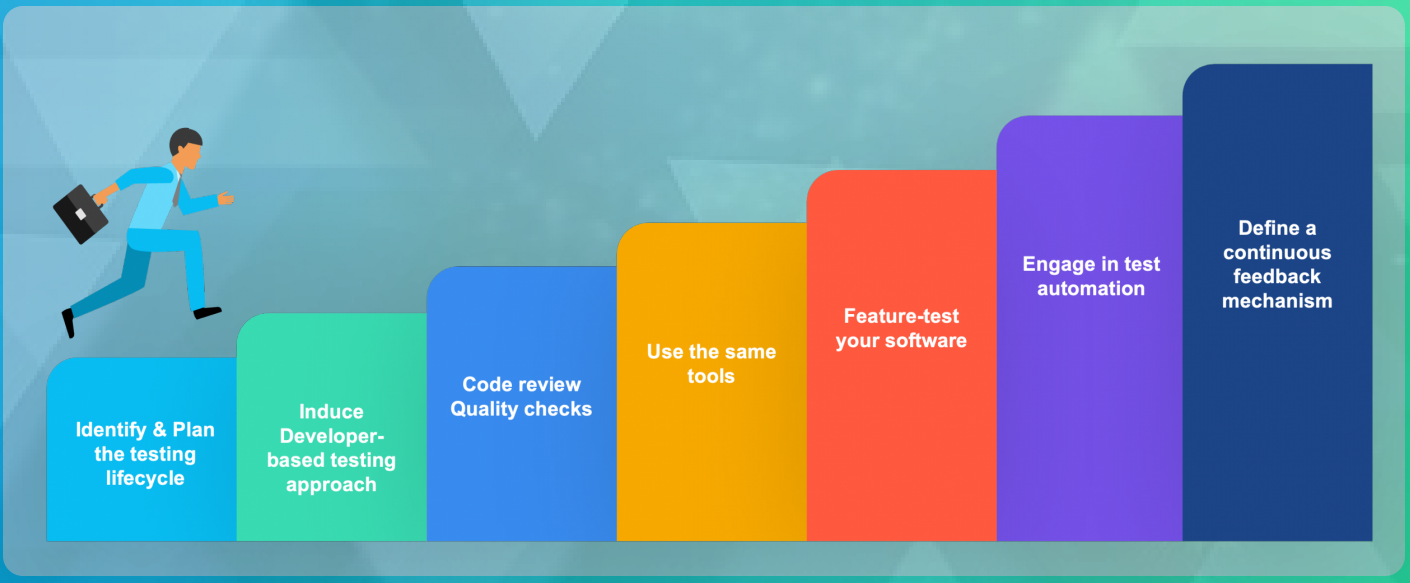
Delivery of the software is delayed: As testing and fixing are done in parallel, the development timeline increases, affecting the overall delivery timeline.

Improper testing – resulting in business losses: Since frequent testing is a tedious task, there might be ignorance from the tester, which can transmit the bugs to production, which in turn affects customer satisfaction which results in potential business loss.

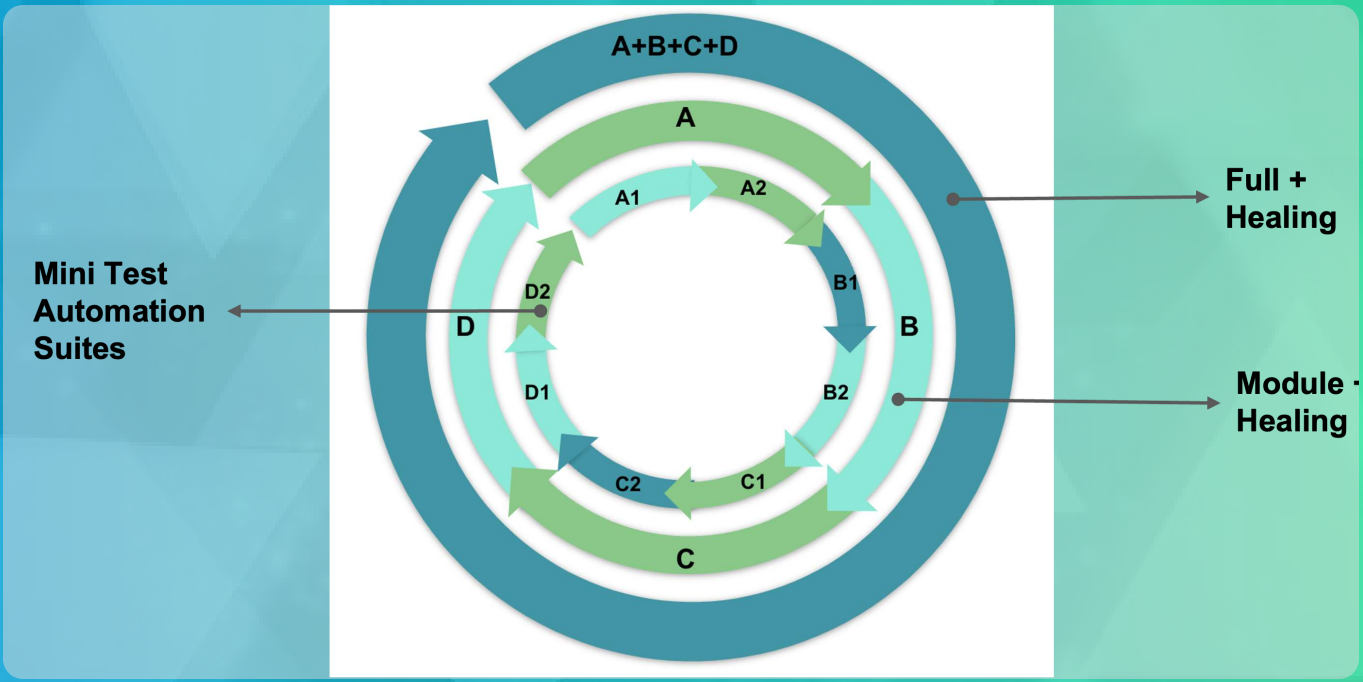
EASIER SAID THAN DONE

Implementing Shift Left is a cultural reset, and for it to be successful, organisations need a lot of team effort. There needs to be a shift in timing, responsibilities, and mindset for this to happen. So how to implement shift-left within Teams?

What does a team need to implement Shift Left?



Healing through Module & Full feature tests



- Full + Modular Overnights include healing which updates test cases/scripts
- Use Full for Staging, Modular Overnights for integration, and mini-suites for Dev
- Move defect discovery left

1. 60% defects in Development
2. 30% defects in Integration
3. 9.9% defects in Staging

Shift Left Testing with AI Automation

Shift left is a repetitive task that can be tedious task to the testers, so there may be errors that can be avoided by automating the test suites.

The development team should adopt test automation technologies because shift-left testing necessitates regular testing. In addition to automating the deployment of new builds, each code increment's testing should be automated. This will relieve stress on the testing team and provide immediate feedback on the code's reliability.

In general, test automation will minimize the time to market by speeding up the development life cycle. Furthermore, it ensures fewer errors are discovered later in the software development life cycle.

For software developers, AI can help them shift left in two areas. It can increase the pace and quality of the code by automatically developing a suite of tests that is updated directly by the AI, allowing the developer to focus on the tasks that only they can complete. DevOps teams increase quality and catch more regressions (catching errors early will help them with the shift left). It will also assist you in identifying more issues earlier in the process when they are easier and less expensive to resolve. This will allow you to increase the frequency of deployments and decrease the time it takes to recover.

And With Shift Left AI-Automation, we are also reducing the cost of testing infrastructure by reducing the constant changing and updating of the test script.

Traditional automation benefits are available with AI-based automation but without the cost and labor of manual maintenance and human error.

The addition of cognitive skills (algorithms) to vast volumes of structured and unstructured data across different systems of record is a significant benefit of AI. Human-driven automation, when combined with AI and machine learning, helps IT businesses to make faster, more accurate decisions about when, when, and how services are delivered.

In terms of the shift left, AI systems' knowledge and insights can be correlated and communicated to IT support employees, end-users, and systems in real-time. AI helps IT, employees, to go beyond lower-level tasks (such as account unlocking and password resets) to those with greater business significance.

When it comes to self-service (level 0), AI is progressing to the point where autonomous, virtual agents can hold increasingly interactive conversations and perform sentiment analysis to understand how a user is feeling (i.e., dissatisfied, frustrated, and happy) and make recommendations, execute processes, and/or open/route tickets in response. This minimizes the need for IT personnel to perform routine activities.

What role does AI Automation play in speeding up the shift-left?

Employees today regard IT services to be far more difficult to access and utilize than consumer services. Long wait times and red tape for users are a result of convoluted SLAs and procedures created with IT. A rigid ITSM approach will not support the diversity in when, where, and how employees want to work, especially among the next generation of workers who grew up with smartphones, tablets, and social media as the norm.

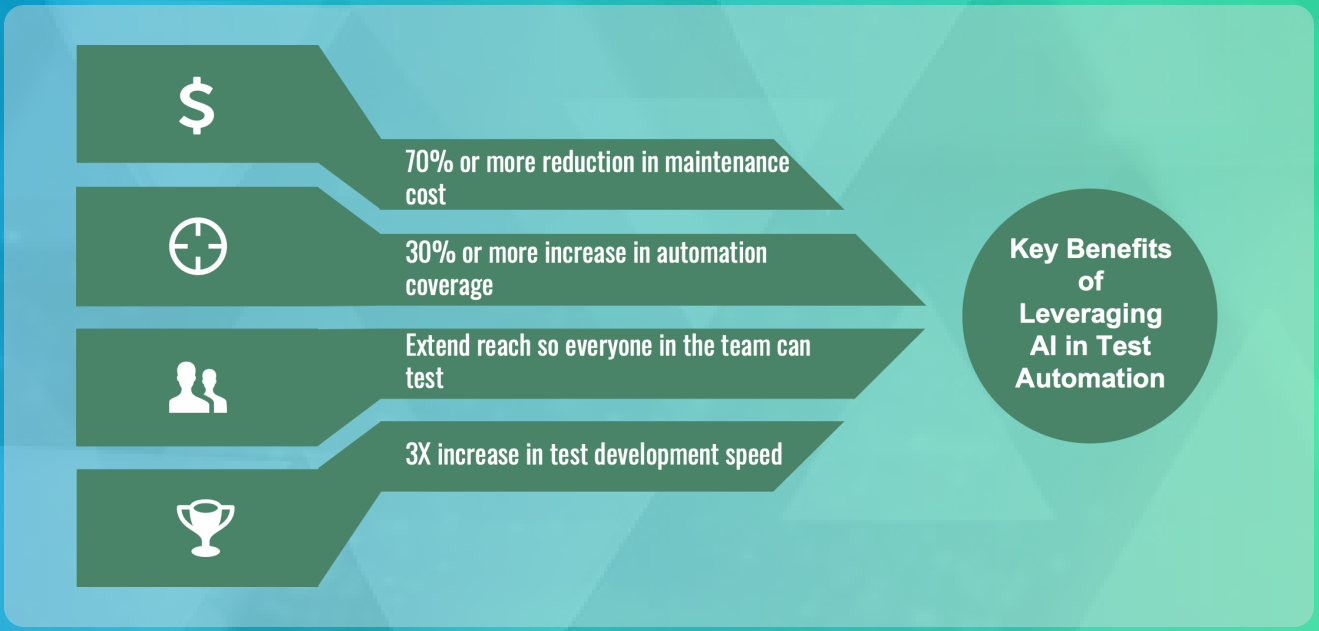
According to a poll conducted by PwC, millennials are uneasy with inflexible business structures and are turned off by information silos. According to the survey, 59 percent of millennials regard an employer's offer of cutting-edge technology to be critical when contemplating a career, 41% said they prefer to communicate remotely at work than face to face or even over the phone.

AI-based automation offers similar benefits to IT, much as voice-activated AI assistants like Siri, Alexa, and Cortana are increasingly being utilised to better understand and optimise consumer experiences based on buyer behaviour, purchase decisions, and post-sales customer support contacts. IT firms can use these technologies to strengthen relationships with business stakeholders and learn how to create more engaging service experiences.

AI-based automation fastens the shift to the left by allowing IT to recognise and predict:

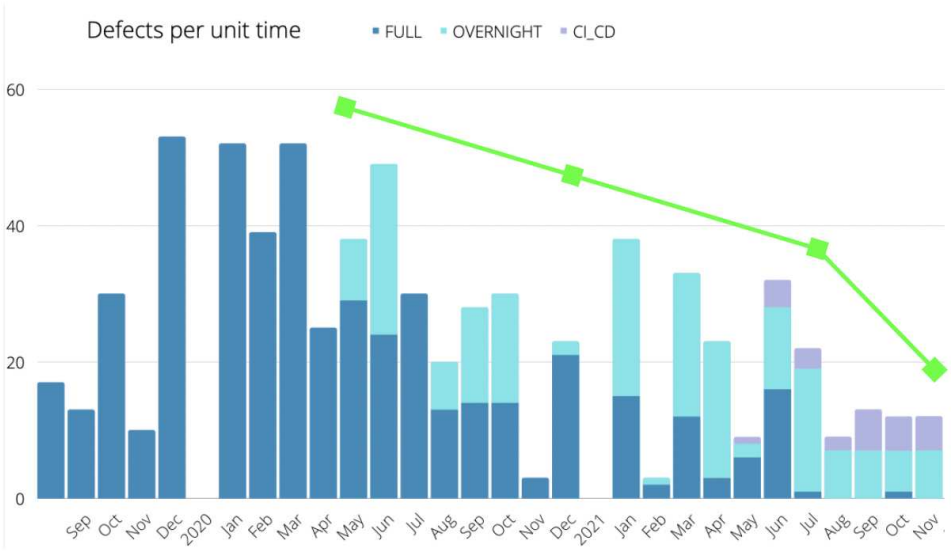
- Best delivery and service channels, timings, and practices for workplace technology
- Processes that are manual and repetitive and can be performed through lower-touch, lower-cost delivery channels
- What services are producing the most questions, and how can they be addressed in a proactive manner.

What role does AI Automation play in speeding up the shift-left?



Key Benefits of Leveraging AI in Test Automation

A Cryo-Specimen Management Company - Scaling a startup over 2 years



Defect discovery shifted left and defects went down

Case Study
Reusability: Rapid Development Rapid Healing
Saved SME Time: Reduced time from Months to Hours
Streamlined: FDA 21 CFR part 11

Customer Testimonial: The Webomates CQ platform enabled TMRW to begin testing the MVP of its Automated Cryo Specimen Management Platform for IVF. “We have continually tested, built, and verified our software without sacrificing time. It helped us not only to move faster by focusing our time and budget on feature development instead of feature verification but also helped us to have auditable information in order to meet regulatory compliance.”

Amit Gupta
CIO

Conclusion

In the software testing community, the shift-left approach is a highly important and popular practice. Its fundamental goal is to better incorporate testing into software/system engineering, allowing for the detection of errors earlier in the development cycle, when they are less expensive and easier to correct. Traditional, incremental, Agile/DevOps and Model-based testing are four major versions of shift left testing that is less well-known both inside and outside of the testing sector. Each of these versions relies on the others to significantly increase testing efficacy, efficiency, and even scope.

As previously stated, Shift Left testing necessitates frequent testing. As a result, automating the process can save you a lot of time and money. This procedure is simple to automate. It also aids in the development process's acceleration.
To execute a successful Shift Left project, you must work as a team, test early, and automate the process.

In the end, the benefits of shift left testing will undoubtedly outweigh the disadvantages. Testers will find themselves delegating some of their responsibilities to developers and handing them additional testing tasks. In more experienced teams, testers take on the role of "coaches," teaching developers how to create better code, avoid problems, and perform their own unit testing. The benefit is that the tester, who was previously occupied with writing test cases, now has more time to delve deeper into the product, working on business cases, penetration testing, performance testing, and adopting AI-based testing solutions.

Artificial intelligence, in general, and test automation, in particular, have had a considerable impact on testing tools and procedures. An examination of the present AI-enabled tools reveals that, while many new capabilities are being introduced, several of them are still in their development.