International Journal of Scientific Research & Engineering Trends
Volume 10, Issue 5, Sept-Oct-2024, ISSN (Online): 2395-566X

Problem in Reviewing Software Testing in the Current Decade

Professor Dharmaraj S Kumbar
KLE's SSMS BCA College Athani,Dist Belgaum

Abstract- Software testing is an inalienable part of the software development life cycle, which directly influences product quality, stability, and, finally, user satisfaction. While the complexity of software systems is growing, testing methodologies face growing challenges related to a lack of coverage, high costs, or time-to-market. In this respect, 56% of organizations currently report that one of the biggest pains is certainly a lack of test coverage, while 40% report high operational costs as a significant barrier to effective testing. This paper looks at these key challenges and some of the emerging trends—impelling AI-driven testing, integration with DevOps, automation, and low-code/no-code platforms—that are rewriting this landscape. With such modern solutions to solving difficulties, organizations will be able to optimize the testing processes, enhance productivity, and deliver quality software that aligns with customer expectations and market demand.

Index Terms- AI-driven testing, landscape, customer expectations, operational costs, handicapped, simplification.

I. INTRODUCTION

Software testing is essential to preserving the dependability and quality of software. Still, testing confronts increasing trends and on-going obstacles in spite of its enormous success.

Reviewing software testing problems makes clear the need for increased testing efficiency, effectiveness, and consistency with customer objectives.

Software testing is the indispensable activity meant for making software products reliable, qualitative, and able to meet user expectations. At the same time, several challenges are still there that are more common in being faced by organizations: 56% insufficient test coverage, while 40% deal with high operational costs. Additionally, time-to-market pressures, together with restricted automation capabilities, keep test efficiency handicapped. Given modern technologies continue to advance, there is a vital need to revisit and re-evaluate testing strategies. This article reviews the new trends in AI-powered testing, DevOps practices, automation, and low-code/no-code solutions; provides practical means for overcoming those challenges. In general, the implementation of such kinds of innovations will result in the simplification of processes related to the testing, reduction of costs, and improvement of the general quality of software.

Present Difficulties Include

- Inadequate testing complexity and range.
- High Operational Costs with Limited Resources
- Concerns about time-to-market and extended testing periods.
- Limited Automation and Test Performance.
- Errors Made by Humans

II. NEW DEVELOPMENTS IN SOFTWARE TESTING TECHNIQUES

- Development and Operations (DevOps),
- Artificial Intelligence,
- Automation Testing,
- Low Code No Code Automation,
- Accessibility Testing, and Performance Testing.

III. PROBLEM IN REVIEWING SOFTWARE TESTING

1. Advances in Testing?

Testing has come a long way in the past ten years because of developments in quality assurance techniques, rising business demands, and technological innovations. These are a few of the most remarkable developments:

The testing sector has been heavily influenced by agile and devops. Shift-left testing is a technique that we have been utilising in recent years.

It allows testing engineers to begin testing as soon as a section of the code is readily available and prevents them from waiting until the entire development process is finished. This method is not only successful, but it also helps to minimise the time it takes to get the product to market[1].

© 2024 IJSRET

Another strategy is agile, which altered the testing procedure. Agile teams work directly with clients to get input at every stage of the process. This guarantees that the application is continuously becoming better. Defects may be spotted early in the development process with the use of agile testing.

2. Has Testing Cost Fluctuated?

Indeed, testing expenses have changed over time for a number of reasons. For example, there is now a wider range of open-source and automated technologies available for software testing, which has eliminated the need for manual testing. Another development is testing via the cloud. Customers simply pay for what they choose to use with cloud testing, which reduces infrastructure expenses.

In certain instances, software costs are rising rather than falling, and typical reasons include-

- When the software grows in size
- Revised guidelines for security testing.
- growing importance in AI/ML testing.

Overall, we can state that the tech industry's acceptance of technology has succeeded in an overall reduction in the cost of software testing in recent years.

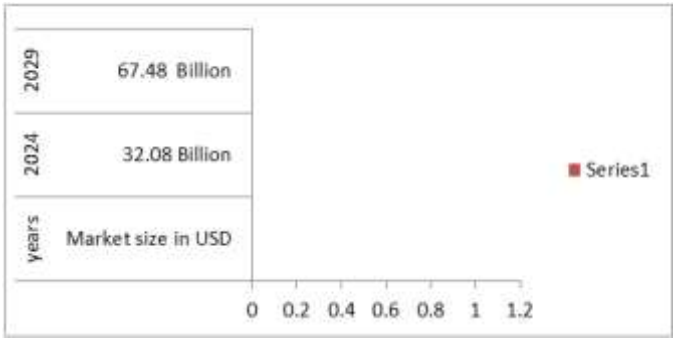

Fig1.automation testing market

What is the Fastest Way to Bring the Software to the Market without Affecting the Testing Phase?

Innovative approaches include the DevOps development process, which encourages working together between both development and operations teams in order to accelerate the delivery of products. It is becoming more popular in software testing as a way to improve test quality and speed.

The software testing process is greatly improved by the DevOps development process in several ways:

Automated inspection- Automated methods for testing are used by DevOps to run tests fast and effectively. This contributes to shorter testing times and more accurate test findings. current examination Continuous testing is made possible by DevOps during the software development lifecycle. This can help in the early detection of mistakes and problems.[2]

In addition to doing basic functionality testing, the following practical testing techniques can help decrease security vulnerabilities:

Vulnerability Test

The vulnerability test is a crucial initial step in network security, scanning systems, or network resources to identify potential safety risks before deploying security features.

The automatic scanning schedule can be scheduled weekly, monthly, or quarterly, depending on the organisation.

Security Scanning

Security scanning identifies system and network vulnerabilities and develops mitigation strategies using automatic and manual scanning methods. It's crucial to consider various factors during network security scanning.

- Security testing should use both test and live data, with different IP addresses for each environment.
- Frequent testing depends on risk factors, with increased IT infrastructure requiring more scans.

Penetration Scanning

A penetration test, conducted by a security specialist, identifies system vulnerabilities and assesses their potential danger, providing mitigation solutions for potential attacks.

SISA(Selective Arterial Secretagogue Injection test) offers automated vulnerability assessment and penetration testing services, including trained developers and ethical hacking, to help organisations avoid data breaches and adhere to industry standard practices like

- requirement analysis
- threat identification
- vulnerability evaluation
- post-exploitation and
- reporting.

3. Risk Assessment

Risk assessment uses threat modelling to estimate a threat's ability to exploit environmental flaws, enabling mitigation or acceptance of residual risk from less likely dangers.

- Tasks associated with risk assessment:
- Identify every potential threat involved in the procedure.
- Concentrate on these risks depending on their chance of existence and what happens if they occur.
- Conduct qualitative risk analysis for risks with high priority.
- Do a quantitative risk analysis on issues that are of medium priority.
- Suggest relevant controls and risk-reduction strategies.

Regular risk assessments are crucial for services-based businesses like TCS, Wipro, and Infosys to protect against known weaknesses and enhance their security perspective in today's data-insecure world. These assessments are conducted

© 2024 IJSRET

monthly or quarterly, considering factors like size, company type, and ecosystem interconnection.

4. Security Audit
An internal security audit is a comprehensive evaluation of an organization's information security measures, aimed at protecting systems from malicious code and data from hackers.

Regular audits can help identify and fix security vulnerabilities. Techniques include code review, fuzz testing, penetration testing, and internal personnel or independent contractors like SISA(Selective Arterial Secretagogue Injection test).

Code review involves manually checking code for flaws like buffer overflows and SQL injections.
Fuzz testing injects random data into a system to identify flaws.
Penetration testing simulates external threats and attempts to gain access.

5. Ethical Hacking
Ethical hacking is a crucial security testing tool, as it allows for a fresh perspective on a system, identifying vulnerabilities that can be exploited by hackers, as it is difficult to identify every vulnerability through technical or manual testing alone.
Malicious hacking alters system databases or steals user information, while ethical hacking, also known as "white hat hacking," intentionally breaks into computer systems to reveal vulnerabilities, rather than causing harm or destruction.

In ethical hacking, the three methods of scanning are as follows:
- Port Examining
- scanning of networks
- Vulnerabilities Assessment

6. Assessment of Posture
An analysis of the condition of an organization's security controls at the moment is done through a security posture assessment.

The assessment may also assist in identifying current risk areas and provide modifications or enhancements that will raise the level of protection for covered assets as a whole.

The assessment involves a penetration test to simulate an attack on the organization's systems, followed by an analysis of current security procedures, providing a report identifying flaws and suggesting solutions.

There are other specialised forms of testing in addition to the seven categories of security testing.

- Security Testing of APIs
- Mobile application security
- Network Security Testing .[3]

Ref- https://www.sisainfosec.com/blogs/10-types-of-security-testing-techniques/

7. When is it Enough Testing?
Determining sufficient testing is crucial, as it cannot predict enough parts. Criteria include time and financial limitations, test coverage, quality measures, and user input.

Apart from this we can also consider the following points.
Absolutely there are a number of methods for figuring out when to end testing, such as:

Correcting Significant and Important Flaws
- It is imperative that all significant and critical flaws are resolved prior to the release of a new product version.
- reaching a certain threshold for open faults
- Once the number of open defects hits a certain threshold, you can cease testing.
- completing all of the scheduled exams
- Once all scheduled tests, including repeated cycles of some or all tests, have been completed, you can cease testing.
- Finishing regression testing
- Once all scheduled regression tests are finished, you can cease testing.
- Assessing the extent of test coverage
- When you have completed thorough testing that addresses every need of the programme, you may conclude your testing.
- Taking time restrictions into account
- If testing is taking too long and turning into a bottleneck, you may run out of time. .[4]

Ref- https://www.scaler.com/topics/software-testing/when-to-stop-testing-exit-criteria-in-software-testing/

8. How Do Formal Approaches Apply to Testing?
Actually, it depends on the area. In situations where the problem's dimensions make traditional testing not practical, I would consider it.

Another thing to consider is that the company needs to either (i) have sufficient funds to cover the technical risk associated with this solution, or (ii) have tried and failed to use other approaches (such as testing and software rewriting) to address the complexity issue. [5]

Ref-https://www.quora.com/Where-are-software-formal-methods-used-How-practical-is-it-to-apply-formal-method-techniques-in-the-development-of-software

© 2024 IJSRET

There are several advanced testing methods specifically designed for critical mission systems like avionics. In what ways can these techniques be effectively used in other sectors?

The answer is Yes. Avionics can be used in expertise areas like:

Here are some examples of successful adaptations of avionics techniques in various industries:

Various fields, such as the automotive sector, healthcare industry, and image processing domain, rely on technology for innovation and advancement. In addition, Industrial Automation has proven to be crucial in controlling systems through robots, streamlining processes for increased efficiency. Energy and Utilities sectors are increasingly focusing on sustainability efforts, utilizing technology to develop more eco-friendly solutions. Furthermore, in the realm of GPS navigation, advancements continue to make travel more convenient and accessible. These diverse applications highlight the extensive reach and impact of technology across various industries, shaping the way we interact with and navigate the world around us. [6]

Ref- https://www.quora.com/What-might-the-future-of-avionics-bring

Which Strategies Work Best for Solving the Test Oracle Problem?
Blockchains require an Oracle application from a third party in order to communicate with the off-chain world because they are unable to retrieve or deliver data directly from or to an external data source. Said another way, the Oracle problem represents the mistrust and security issues that arise when smart contracts and third-party oracles interact.then what is the solution for oracle problems.

The first fully decentralized Layer-2 Oracle solution in the world, Umbrella Network provides affordable, scalable, fast data. It offers more than a thousand data pairs and processes batches using Merkle trees.

IV. CONCLUSION

In summary, the significance of effective software testing cannot be overstated as it plays a pivotal role in ensuring the quality of a product and ultimately leading to user satisfaction. However, many organizations encounter formidable challenges along the way, with common issues arising from insufficient test coverage and the burden of high operational costs. To navigate these challenges successfully, it is crucial for organizations to adopt and integrate modern methodologies that have proven benefits. Examples of such methodologies include AI-driven testing, which leverages artificial intelligence to enhance testing processes, and DevOps integration, which streamlines collaboration between development and operations teams. By embracing these cutting-edge techniques, organizations can greatly improve the efficiency and effectiveness of their software testing practices. The ultimate goal of implementing these methodologies is to achieve the delivery of software that not only meets but exceeds market demands, setting a new standard for quality and innovation. This proactive approach to software testing ensures that organizations are better equipped to stay competitive in today's dynamic and fast-paced technology landscape.

REFERENCES

1. https://www.planview.com/resources/articles/how-agile-method-transforms-software-testing/ and s://www.lambdatest.com/blog/software- testing-trends/
2. https://www.google.com/search?q=how+DevOps+development
3. https://www.scaler.com/topics/software- testing/when-to-stop-testing-exit-criteria-in-software-testing/
4. https://www.testrail.com/blog/define-enough-testing/#:~:text=The%20time%20for%20testing%20ran%20out&text=You%20may%20have%20something%20new,that%20it's%20time%20to%20stop.
5. Ref-https://www.quora.com/Where-are-software-formal-methods-used-How-practical-is-it-to-apply-formal-method-techniques-in-the-development-of-software
6. Ref- https://www.quora.com/What-might-the-future-of-avionics-bring

© 2024 IJSRET