

Test Smarter Not Harder with Agile Testing

Test early, test often, and adapt to change.

White Paper

Test Smarter
Not Harder

Abstract

Testing plays a crucial role in software development, ensuring that the software meets the required quality standards. It also serves as an **integral part** of a software development life cycle.

Its importance is highlighted in several key aspects, including:

- cost effectiveness
- enhancing customer
- preventing failure
- supporting continuous improvements

However, sometimes testing teams face the challenge of not having test data available during the **testing phase**.

This white paper discusses the **issues** and **proposes** agile testing strategies to tackle this situation effectively.

It presents approaches such as:

- exploratory testing
- ad hoc testing
- equivalence partitioning
- boundary value analysis
- Test-Driven Development (TDD)
- collaborative testing
- risk-based testing

The paper emphasizes the importance of close collaboration between **development** and **testing teams** and offers recommendations for handling the absence of test data in an agile environment.



Source: Microsoft Copilot AI

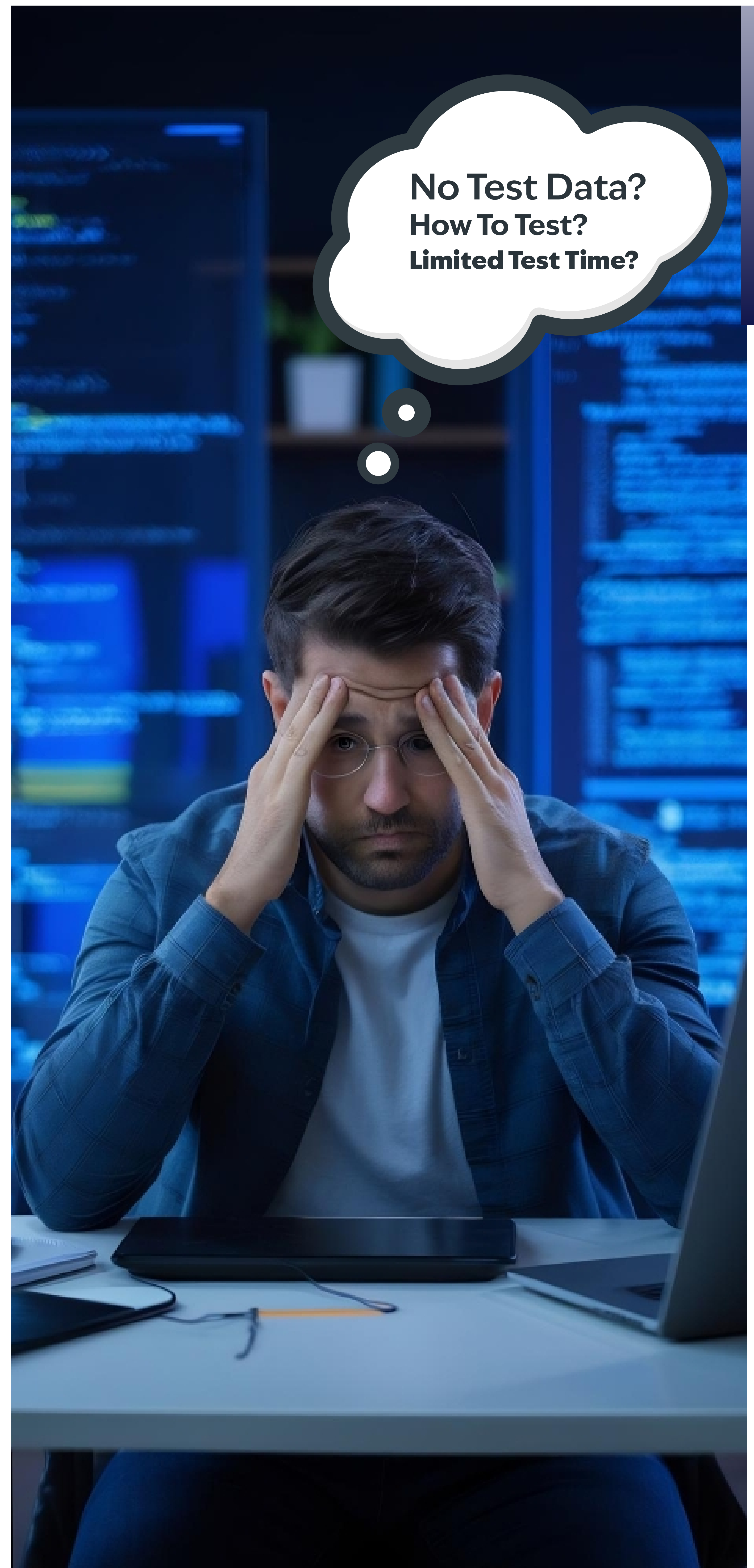
“In Agile, testing is not a bottleneck. It’s a continuous feedback loop that drives development.”

- Mike Cohn; certified scrum trainer, co-founder of the Scrum Alliance and a renowned Agile software development expert.

Introduction

In an agile development environment, the testing phase is crucial to ensure the quality and functionality of the software product.

As the complexity of software systems increases, so does the need for effective testing strategies to ensure robust and reliable applications.



However, if and when testers have **no test data** to rely and work upon, this becomes a concern. The absence of the data can **hinder the timely execution** of test cases.

This white paper addresses this challenge and provides agile testing strategies to overcome it.

Agile Testing Approaches in the Absence of Test Data

Exploratory Testing

Exploratory testing is an **approach that relies on testers'** expertise, experience, and intuition. It is a critical approach when there is a lack of data.

In the absence of predefined test data, testers can explore the software application by designing tests, simulating various scenarios, executing them, analyzing results, and learning from the outcomes.

This will allow testers to verify the system's behavior dynamically. Using this approach can help uncover defects and issues that may not be captured by pre-planned test cases.

Ad Hoc Testing

Ad hoc testing involves **executing tests without specific test cases or plans**, an informal and unstructured approach.

This method allows testers to explore the application spontaneously, relying on their intuition and experience to identify potential defects. They can perform **on-the-fly testing**.

Ad hoc testing is useful when test data is unavailable, as it allows testers to adapt quickly and apply their knowledge to validate the system. Some of its characteristics include error guessing, spontaneity, and a focus on new discoveries.

Equivalence Partitioning (EP) and Boundary Value Analysis (BVA):

EP and BVA are **two essential black-box testing techniques** that enhance the efficiency and effectiveness of software testing. These techniques help **identify critical areas** of the system by dividing the input space into equivalent classes and testing representative values from each class. This approach aids in achieving coverage and ensures that the system behaves as expected.

The purpose of EP is to reduce the number of test cases while ensuring comprehensive coverage by selecting representative values from each partition, while BVA is based on the principle that errors are more likely to occur at the boundaries than in the middle of input ranges.

Agile Testing Approaches in the Absence of Test Data - cont.

Test-Driven Development (TDD)

Test-Driven Development is an agile approach that involves **writing automated tests** before **implementing the functionality** and before **coding new features**.

This practice not only defines the requirements of the feature but also ensures that testing is integrated into the development process from the outset.

In the absence of test data, the development team can utilise TDD to drive the development process. They can create tests based on the expected behavior and requirements, guiding the implementation of features.

Collaborative Testing

Close collaboration between development and testing teams is essential when test data is not available.

This approach emphasizes **teamwork and cooperation among various stakeholders**, including developers, testers, product owners, and sometimes even end-users.

This method aims to enhance the quality of software products by leveraging the diverse skills and perspectives of team members throughout the development process. This means, teams work together to identify potential test scenarios and use cases. By brainstorming and creating test data on-the-fly, they can verify the system's behavior effectively.

Risk-Based Testing

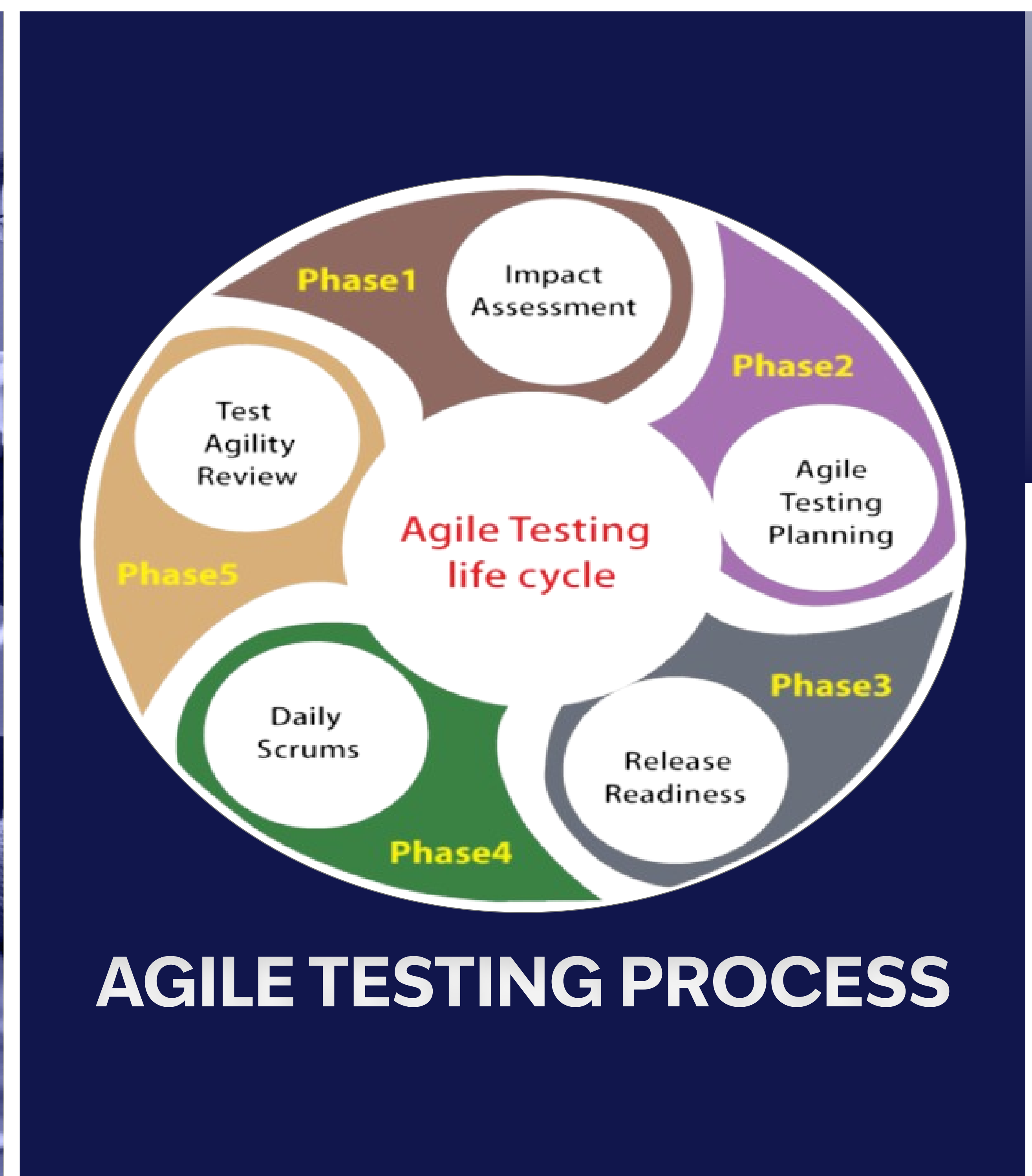
This is a strategic approach to software testing that **prioritizes testing efforts** based on the potential risks associated with different components of a software application.

It aims to allocate limited testing resources to the areas that pose the highest risk to the project's success, ensuring that critical functionalities are tested thoroughly.

This approach ensures that the testing effort is focused on functionalities that are more likely to have a significant impact on the software's performance, stability, or security.

Recommendation and Conclusion

In the absence of test data, agile testing can still be effectively conducted by leveraging various strategies and methodologies that emphasize collaboration, exploratory techniques, and adaptability.



One of the more crucial elements is to maintain effective communication and collaboration between development and testing teams.

Additionally, efforts should be made to bridge the gap in requirements and planning, ensuring that necessary test data is made available for future testing cycles.

“A good tester prevents problems; a great tester finds them.”

- Keith Klain; software testing expert with over 25 years of experience, known for building global QA teams, aligning testing with business goals, and championing workforce development through technology training initiatives. for global financial services and consulting firms in the US, UK, and Asia Pacific.

Creator of Smarter Ecosystem for a better Malaysia

Established in 2000, TM R&D is the innovation arm for TM Group focusing on creating smarter ecosystems to make business and life easier for a better Malaysia. TM R&D's solutions are clustered around four (4) pillars namely Intelligent Platforms, Data Brokerage, Connectivity/Tools and IR4.0/Digital Solutions.

Growing from strength to strength since 2016, TM R&D has won multiple global awards and generated more than 2,800 Intellectual Property Rights (IPRs) and 1,400 digital assets to-date.

TM R&D's innovations are all developed in-house and cut across multiple verticals such as Utilities, Retail, Agriculture, Healthcare and Education with safety and productivity as the top priority.

As TM R&D continues to expand beyond connectivity and into smarter digital ecosystems, its role in TM has become more prominent and exciting.

We are looking for remarkable people to join us. People who are courageous enough to push boundaries, curious enough to experiment with new technologies, and who have the determination to drive new ideas forward. A new opportunity awaits you here in TM R&D.

Be a part of our family at <https://www.tmrnd.com.my/jobs/> or email recruit@tmrnd.com.my

For more information about TM R&D and its products and services, kindly visit to:

www.tmrnd.com.my

