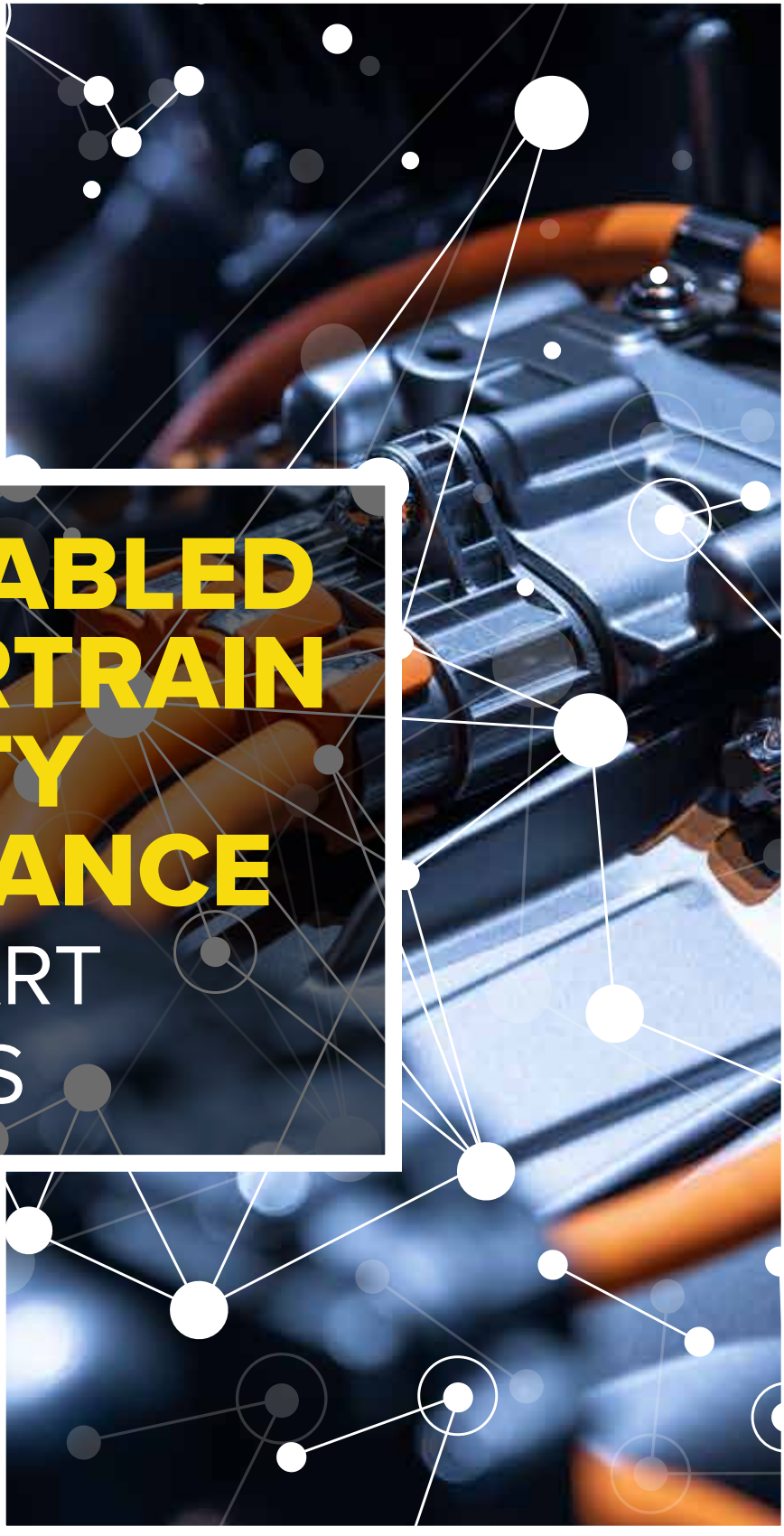
happiest minds
The Mindful IT Company
Born **Digital** . Born **Agile**



IOT-ENABLED POWERTRAIN QUALITY ASSURANCE FOR SMART VEHICLES

INTRODUCTION

IoT plays a crucial role in revolutionizing the powertrain industry as demand for more connected and smarter vehicles continuously increases. Vehicle data is continuously transmitted to the cloud before being evaluated and compared with existing diagnostic, quality, development and production data. With respect to this, smart algorithms and digital powertrain models in the cloud enable quick and reliable detection of any irregularity at the system and component level.

OEMs receive health data on powertrains and thus avoid unplanned failures. Workshops will also benefit from information in the cloud for better planning and optimal implementation of their work process.

Powertrain systems in electric vehicles benefit from the realization of IoT-driven insights with respect to battery health, optimal fueling infrastructure, and regenerative energy handling. In hybrid powertrains, IoT leads to perfect coordination between the internal combustion engine and the electric motor so as to deliver better efficiency and performance

.

The global IoT market for the automotive industry was estimated to be USD 131.2 billion in 2023 and is expected to grow to USD 322.0 billion by 2028. The market size will be more focused on autonomous driving, which will involve many IoT devices communicating with each other at a lower level.

IMPACT ON **AUTOMOTIVE INDUSTRY**

IoT in the automotive industry, especially in powertrain, has a significant impact on gearbox providers. IoT-enabled gearboxes provide enhanced efficiency, real-time monitoring and predictive maintenance. The following are the reasons for the adoption of IoT in the gearbox field:

1 Performance Optimization

Different sensors embedded in gearboxes measure parameters like speed, temperature, vibrations, torque, etc., and they are continuously monitored. This collected data can be shared with gearbox providers to identify anomalies early and enable them to take preventive actions which leads to improved efficiency and reliability.

2 Reducing Downtime

After collecting data from IoT sensors, predictive analysis is applied to the data to forecast potential failures. This helps gearbox providers for scheduled maintenance, leading to a reduction in downtime.

3 Remote Diagnosis

IoT connectivity enables remote diagnostics allowing gearbox providers to diagnose issues without the need for physical inspection, hence better customer satisfaction.

4 Continuous Revenue in Business

IoT has enabled gearbox providers to shift their business from selling products to offering data-driven services such as performance analytics and optimization recommendations to their customers.

Though there are a lot more opportunities that are introduced with the advent of IoT, gearbox providers need to invest in new technologies, training, and infrastructure to integrate IoT into their products, leading to slow adoption, especially by smaller OEMs. A few common challenges are listed below:

Data Management

A large volume of data related to speed, temperature, vibration, etc., is generated by the gearbox. It is difficult for gearbox providers to manage and analyze this real-time data and provide meaningful insights to customers. Hence, there is a need to identify analytics tools that help them to properly interpret the data, leading to accurate predictive maintenance.

Data Security

Data security has become a major concern since a vast amount of data will be collected by IoT sensors and sent to the cloud. Advanced encryption techniques are required to protect data transmitted between systems, ensuring integrity and confidentiality.

Interoperability Testing

The gearbox needs to communicate with other components in powertrain systems (For example, engine and braking system). Due to the inconsistent standards followed in this industry, these components might be from another manufacturer that uses different communication protocols. This leads to interoperability issues.

CHALLENGES IN VALIDATION

IoT systems in the powertrain involve many interconnected components that include sensors, actuators, mobile and web applications as well as cloud platforms. Validation of the seamless interaction between these components is complex and hence requires end-to-end testing to ensure cohesive functioning between these components.

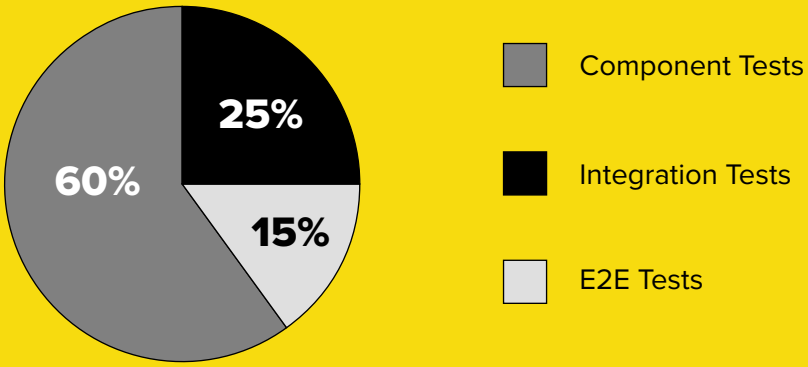
An IoT testing solution typically involves a combination of device layer and application layer testing.

Device Layer Testing: Covers the functionality of IoT hardware, sensors, and small computer systems. Validation of these should guarantee the units function as per designated roles.

Application Layer Testing: Covers the testing of mobile and web interfaces, and backend applications for incoming data from IoT devices to confirm correct reception.

In today's era, where automation has become a priority, there is more stress on leveraging automated testing solutions. According to the "Test Automation Report by SmartBear", real-world test automation distributions are shown below:

Automation Test Breakdown



Though there are numerous automation testing solutions available in the market for component testing (eg: Appium for mobile, Selenium for Web UI, Requests Library for Cloud APIs, UART for the device, etc.), organizations still face hiccups in achieving end-to-end automation as there are no ready-made solutions that suit their requirements.

Hence, a lot of important system functionalities will remain untested via automation, creating a gap of 30-45% of automation test coverage and causing several key issues:

Insufficient Test Coverage

Many components are not tested in combination with others, leading to missed integration issues.

Manual Interventions

Automation efforts are fragmented, requiring manual effort to connect disparate tools or systems.

Reduced Reliability

Incomplete test automation coverage may result in potential issues being missed before deployment leading to reduced reliability.

One needs to find a way to integrate all these components together to achieve full end-to-end test automation coverage. Let us examine how a unified automation framework can address this issue.

SOLUTION
A CONSOLIDATED IN-HOUSE FRAMEWORK

It is a proprietary framework that can smoothly integrate all the components of the IoT ecosystem to achieve full test automation. Such a framework can reuse existing open-source solutions for component tests and introduce custom-designed modules to handle integration and end-to-end tests.

Salient Features of the In-House Solution

Cross-Component Interaction
The framework needs to be capable of testing various device interactions, including sensor simulation triggers, APIs, backend services, and mobile and web UIs, validating the transmission of data between components.

Personalized Wrapper
We can build a custom structure that wraps around all components within existing automation solutions, such as simulators mobile automation, and API testing tools. We create custom wrappers around them to support BDD or keyword-driven development.

Seamless Interaction
The test strategy needs to be capable of performing full end-to-end test scenarios by simulating the devices, mobile and web user interfaces, as well as cloud APIs in the same way they solve actual use case scenarios.

CI/CD Pipeline Integration
The framework should be integrated into the CI/CD pipeline so that tests can run automatically with each code change and provide rapid feedback to developers.

Device Simulator /Hybrid Board
It is necessary for the framework to design device simulators or hybrid boards for comprehensive testing of the device and flashed firmware.

Benefits of this Approach

Detailed Test Coverage
To make sure that all the features are tested extensively, the testing procedure must bring all components together.

Accelerated Feedback Loops
By automating the end-to-end testing process, issues can be found at an early stage, reducing manual efforts.

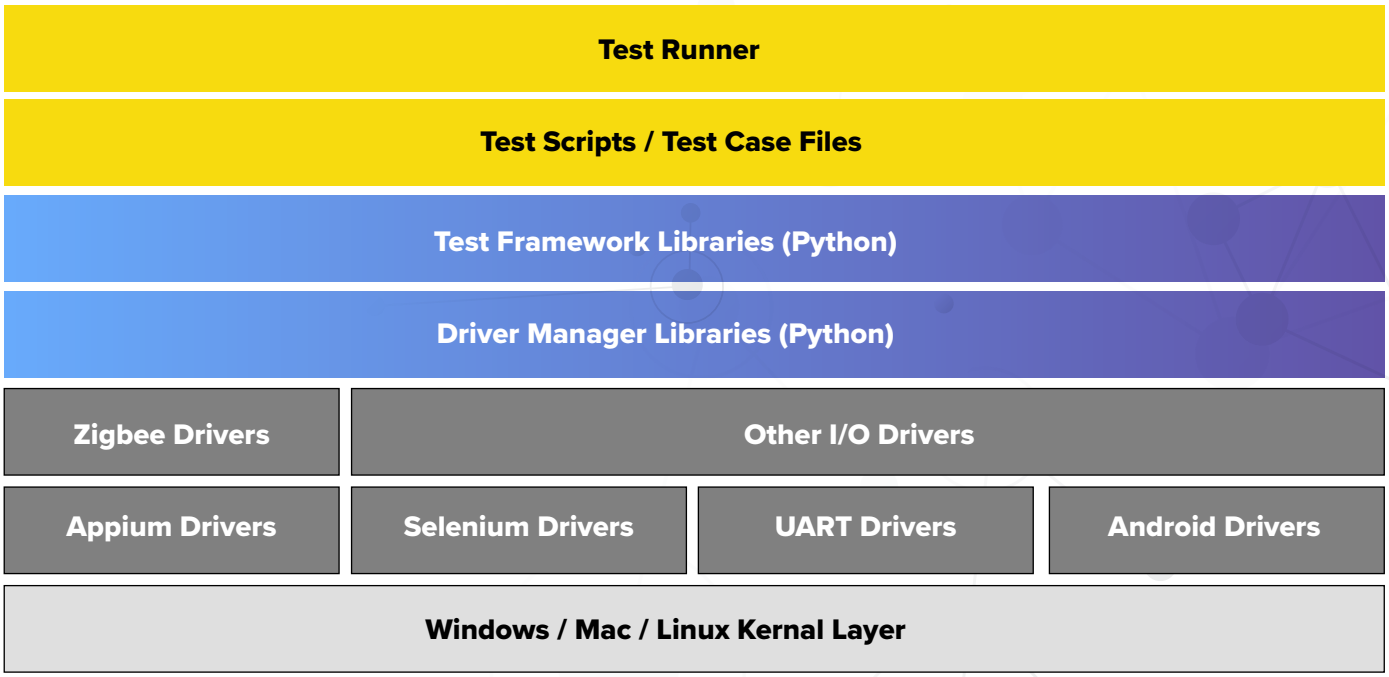
Ability to expand
It supports adding new test cases and functionalities without significant rework, allowing the framework to grow with a user's testing needs.

Test Efficiency
While building an in-house solution may require a significant upfront investment, the long-term savings from automated testing, in terms of test efficiency, can lead to reduced testing cycles and improved product quality.

FRAMEWORK APPROACH

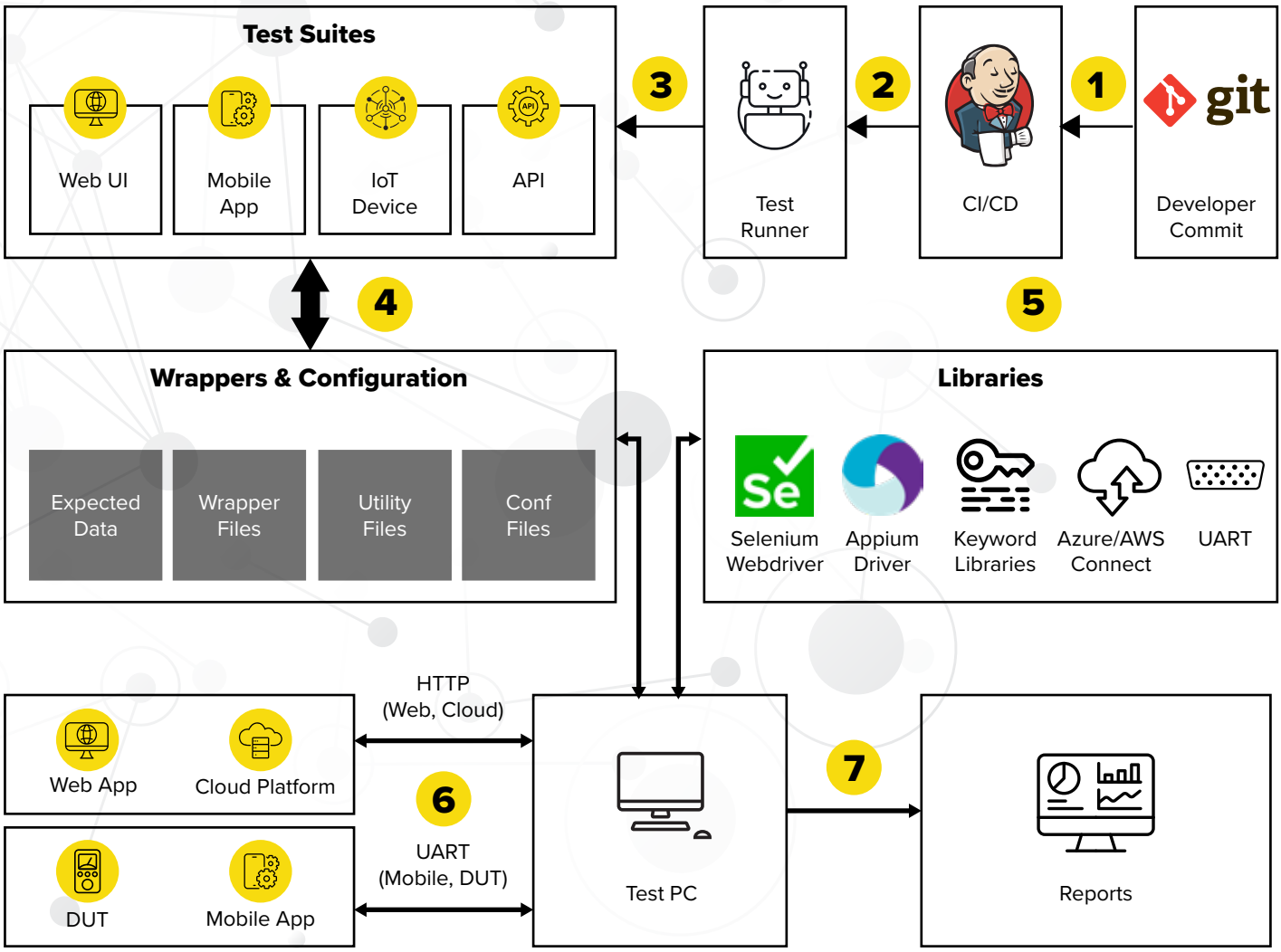
In order to understand how various components interact and align with each other, it is necessary to take a structured approach. Here's a breakdown of the key considerations that will help design and build an effective test framework.

Overview of Layered Approach



Layers	Description
OS Compatibility	Determine the operating systems (OS) and their versions on which the tests will be executed.
Drivers	Both standard and custom drivers required for interaction with specific hardware must be compatible with the chosen operating system.
Driver Managers	The manager should handle all the communication between devices and the framework and be able to provide relevant error logs for debugging in case of hardware failures.
Test Libraries	These libraries expose the keywords and functions in the framework for test execution.
Test Scripts	These are actual test cases that use the libraries to perform test scenarios.
Test Runner	The engine that triggers and controls the execution of tests.

FRAMEWORK ARCHITECTURE



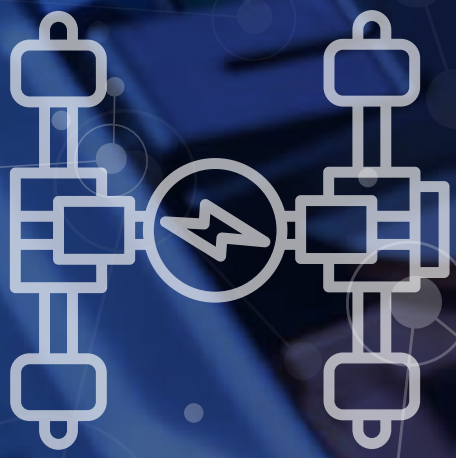
To implement the in-house solution effectively, these are the key requirements that must be considered during framework design:

Device Simulator/Hybrid Board Control Hooks 1	• To thoroughly test the device, we must develop a simulator or specialized hooks for device simulations. This will help discover the flashed firmware behavior for boundary conditions. • Mostly, IoT devices are connected using UART or SSH. We can use Python-based open-source libraries like PySerial and Paramiko, respectively, to access these interfaces.
UI Automation 2	• To test user interfaces for web and mobile applications via automation, our framework must provide the required support. • When selecting a tool for UI automation, a few important aspects to consider include open-source availability, community support, and cross-platform and cross-browser testing capabilities. • It is recommended to adopt a framework-based approach to the Page Object Model, as it offers high visibility and reduces maintenance overhead. • Selenium WebDriver for the web and Appium for mobile can be used effectively for UI automation.

API Automation **3**	• The automation framework should support API test automation to ensure that the backend system behaves as expected with correct data transmission and response handling between services. • The Python requests library can be used to automate API testing.
Modular Design **4**	• The framework libraries must be organized in a structured and modular format for IoT device testing, UI automation, and API testing. This modularity allows components to be tested separately and integrated later when required.
Test Data Management **5**	• Test data shall be managed properly such that it shall allow easy modifications without the need to update test scripts. • The framework should provide a robust mechanism to generate, manipulate, and store test data, ensuring flexibility and reusability in tests. • External files, such as JSON or YAML, can be used to manage and store variable data.
Central Repository **6**	• Version controlling will effectively maintain the test repository and help developers with code review and pull requests. • Git is an effective version control tool for this purpose.
Test Suites **7**	• The framework should allow the creation of test suites based on specific functionalities or scenarios, providing a more structured testing approach. • The framework should also support a tagging mechanism to organize and execute tests efficiently. • The Robot Framework is a useful tool for creating structured test frameworks. • AI-powered test automation tools (E.g. Github, Copilot) can be used to generate test scripts automatically based on business requirements, leading to a reduction in manual efforts.
Test Reporting **8**	• The framework shall generate detailed, organized and visually-appealing test reports that provide insights into the test status, results and logs for debugging. • The Robot Framework can be used to generate well-organized HTML reports for comprehensive test results.
Test Runner **9**	• A test runner is the central authority to trigger tests and organize automation workflows. • It integrates various testing tools, such as, mobile, web, backend, and device simulators. • The Robot Framework can be used as a test runner.
Integration with CI/CD Tools **10**	• The framework should have the capability to integrate with any CI/CD tool. • The Robot Framework provides plugin support for popular CI/CD tools, ensuring smooth integration into the continuous delivery pipeline.

EXECUTION PROCEDURE

1. The developer shall commit the latest code to the version control system (GitHub/GitLab).
2. Upon detection of changes in Source Code Management (SCM), the CI/CD pipeline starts and automatically triggers the test case execution.
3. Wrappers and configurations fetch environment details and expected data to determine which tests need to be executed.
4. The test runner (Robot Framework) initiates test cases for all the components like; embedded systems, web, mobile, and API components, along with end-to-end scenarios.
5. Tests are executed independently for each component (embedded devices, UI, APIs) while maintaining integration.
6. HTTP request-response and UART communications are used to interact with components under test during execution.
7. Results are stored, and detailed test reports are generated for analysis.



CONCLUSION

As IoT system ecosystem is expanding continuously, it is important to ensure that they function smoothly across devices, UIs, and Cloud APIs services. To achieve full end-to-end testing automation, we need to develop a proprietary solution that can integrate testing across all layers, from device to application. By building a consolidated framework suited to specific project needs and integrating it into the CI/CD pipeline, organizations can bridge automation gaps and ensure comprehensive testing. Excelling in end-to-end testing for IoT devices enables companies to confirm that their IoT solutions are functional, reliable and ready for real-world implementation.

ABOUT THE AUTHOR

Lovish Jindal is an ISTQB certified tester with over 9 years of experience in validating embedded and automotive domain products. He specializes in developing automation frameworks for both Hardware-in-the-Loop (HIL) and Software-in-the-Loop (SIL) testing, ensuring comprehensive end-to-end product validation. His expertise lies in delivering high-quality products through deep involvement in every stage of the testing process.

As a test lead, Lovish is proficient in multiple programming languages, including Java and Python, and skilled in various automation tools such as Selenium, Appium, and Jenkins. He focuses on developing robust automation solutions that minimize manual intervention, ensuring more efficient and reliable test execution.

Lovish holds a Bachelor of Engineering degree in Electronics and Communication from VIT, Vellore.

LOVISH JINDAL
TEST LEAD, PDES

For more information, write to us at
business@happiestminds.com

ABOUT HAPPIEST MINDS

Happiest Minds Technologies Limited (NSE: HAPPSTMNDS), a Mindful IT Company, enables digital transformation for enterprises and technology providers by delivering seamless customer experiences, business efficiency and actionable insights. We do this by leveraging a spectrum of disruptive technologies such as: artificial intelligence, blockchain, cloud, digital process automation, internet of things, robotics/drones, security, virtual/ augmented reality, etc. Positioned as 'Born Digital. Born Agile', our capabilities span Product & Digital Engineering Services (PDES), Generative AI Business Services (GBS) and Infrastructure Management & Security Services (IMSS). We deliver these services across industry groups: Banking, Financial Services & Insurance (BFSI), EdTech, Healthcare & Life Sciences, Hi-Tech and Media & Entertainment, Industrial, Manufacturing, Energy & Utilities, and Retail, CPG & Logistics. The company has been recognized for its excellence in Corporate Governance practices by Golden Peacock and ICSI. A Great Place to Work Certified™ company, Happiest Minds is headquartered in Bengaluru, India with operations in the U.S., UK, Canada, Australia, and the Middle East.

happiest minds
The Mindful IT Company
Born **Digital** . Born **Agile**

www.happiestminds.com