Automating Testing with LabVIEW: A Requirements-Driven Approach

Carlos Renato dos Santos[1,2][0009-0003-2132-2164], Adriano Costa Pinto[1,2][0000-0002-5198-4623], Fabio Francisco de Carvalho Caballero[2][0009-0003-1425-591X], Luciana Pereira Simões[2][0009-0006-0847-7032], and Alison de Oliveira Moraes[1,2][0000-0002-6493-1694]

[1] Instituto Tecnológico de Aeronáutica, São José dos Campos, Brazil
[2] Instituto de Aeronáutica e Espaço, São José dos Campos, Brazil
carlosrenato@ita.br
adrianoacp@ita.br
caballeroffcc@fab.mil.br
lucianalps@fab.mil.br
aom@ita.br

Abstract. Due to safety and performance requirements, lithium-ion cell testing is critical in aerospace applications. Manual testing methods are error-prone and inefficient, leading to inconsistent results and poor traceability. This paper presents a requirements-driven methodology for automating Li-ion charger testing using LabVIEW. The approach utilizes the IEC/ISO/IEEE 29148-2018 writing requirements syntax to assist the software design, improving maintainability and reducing technical debt. A case study illustrates the development of an automated system for capturing charging curves, linking each code section to specific requirements. The solution improved debugging, documentation, and test coverage while reducing misunderstandings of informal specifications. Although direct performance comparisons are limited, automation allowed operators to focus on parallel tasks, enhancing man-hour efficiency. Results show that requirements traceability improved code clarity and streamlined communication between stakeholders. This structured method can be applied to other graphical programming environments where test coverage, maintainability, and resource optimization are essential.

Keywords: Requirements Engineering, Test Automation, LabVIEW, Verification and Validation, Traceability in Testing

1 Introduction

Testing lithium-ion (Li-ion) cells is a critical task in aerospace applications due to stringent safety and performance requirements. Traditional manual testing approaches, while common, are prone to human error, time-consuming, and often lack the traceability necessary for rigorous validation. These limitations can compromise the reliability of the tests and hinder future software maintenance and scalability. LabVIEW, a graphical programming environment widely adopted

Proceedings of the 28th Workshop on Requirements Engineering (WER2025),
August 20-22, 2025, Rio de Janeiro-RJ, Brazil.
https://doi.org/10.29327/1588952.28-28

for test automation, offers powerful tools for data acquisition and control; however, it poses challenges in documentation and long-term maintainability, often resulting in poorly structured "spaghetti" code [2].

To address these challenges, this paper presents a requirements-driven methodology for automating Li-ion charger testing using LabVIEW. By applying formal requirements engineering practices, based on the ISO/IEC/IEEE 29148:2018 [4] standard, the approach ensures that the system is functional, traceable, and maintainable. Each code segment is tied to a specific requirement, enhancing Li-ion battery test coverage, debugging, and stakeholder communication.

A case study demonstrates how the automated system captures charging curves while aligning software behavior with defined requirements, resulting in improved documentation, operator efficiency, and software clarity. This foundation supports broader application in test automation, where reliability and traceability are essential.

The rest of this work is organized as follows: Section 2 details the device under test (DUT), the test challenges, and an overview of the development environment. Section 3 presents the solutions and results, thus discussing the requirements-driven testing strategy. Section 4 concludes with the benefit of aligning LabVIEW tests to formal requirements.

2 Case Description

This section provides the context and technical background of the case study: the development and automated testing of a Li-ion cell charger using LabVIEW. The test setup was designed to ensure a structured and reliable evaluation of Li-ion cell charging curves, following the standard constant current/constant voltage (CC/CV) profile.

Lithium-ion (Li-ion) cells are widely used in aerospace and other safety-critical applications, where their quality and performance must be rigorously tested [6]. A key aspect of this evaluation is analyzing the charging curve, which provides critical insights into the cell's behavior. Ensuring that a cell follows the expected charging profile is essential to verify its suitability before subjecting it to further performance tests.

Li-ion batteries were the leading cause of catastrophic events, as mentioned by Williard [7]. Furthermore, the Federal Aviation Administration (FAA) released reports (see Figure 1) containing trend data on the number of Lithium battery incidents on aircraft [1].

Manual testing of this device typically involves connecting lab instruments (power supplies, loads, thermocouples), configuring test conditions, and manually logging measurements during the charge cycle. This method is cumbersome and time-consuming, leading to inconsistency and repeatability issues. Table 1 summarizes the manual testing challenges.

To address these challenges, the team implemented an automated testing using LabVIEW with supporting test hardware and a structured test management process. Figure 2 presents the automation hardware test setup scheme

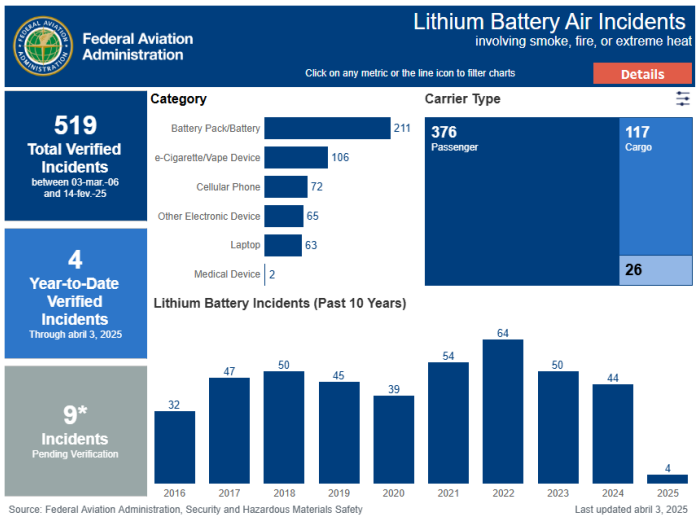

Fig. 1. Lithium Battery Air Incidents according to the FAA [3].

Table 1. Testing Challenges in Manual Validation of the Li-ion Charger

Challenge	**Impact**	**Root Cause**
Manual Setup	Slow and error-prone	No reusable configuration scripts
No Traceability	Hard to audit test reports	Manual test execution and logging
Long Charge Cycles	Blocks test throughput	Full CC/CV cycle takes 3–5 hours

and outlines the key components of the test system. A brief description of each equipment in the cell's testing. The power supply will provide the voltage and current to charge the Li-ion cell. At the same time, the data acquisition will be responsible for monitoring the temperature during the charge using a PT100 temperature sensor. The computer hosts the LabVIEW application.

LabVIEW offers a streamlined approach to user interface design. Its simple drag-and-drop functionality allows for the intuitive placement of various indicators and controls. This capability significantly enhances the efficiency and effectiveness of creating user interface applications. Furthermore, LabVIEW made it easier to represent state.

While LabVIEW provides a powerful platform for developing intricate visual tests, it also presents particular challenges associated with visual programming. The potential for overlapping connection lines can result in a complex design that may become difficult to maintain. This phenomenon, often called LabVIEW "spaghetti" code [2], highlights the importance of careful design practices to ensure clarity and maintainability in programming. Figure 3 shows poorly designed LabVIEW code with overlapping lines, making it difficult for the designer to debug and maintain.

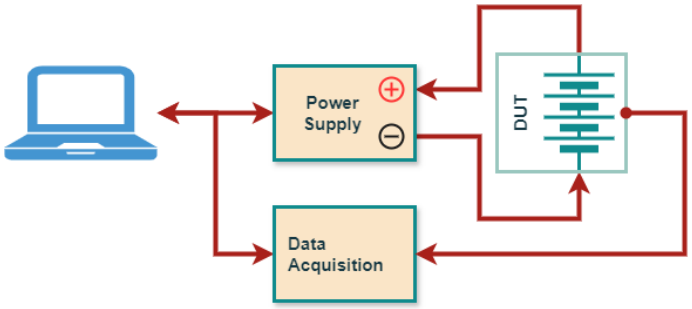

Fig. 2. Proposed hardware setup to evaluate the Li-ion cell charge curve.

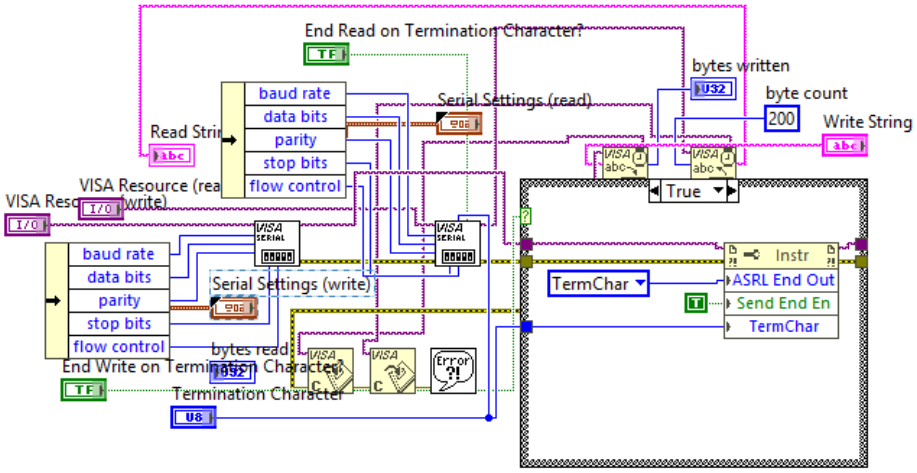

Fig. 3. Example of LabVIEW "spaghetti" code style.

A requirement-driven approach was employed to address the inherent LabVIEW challenges. The requirements were compiled through discussions with the individuals responsible for designing the test and the operators who will utilize the completed application. These individuals conveyed their expectations regarding the application, which have subsequently been translated into requirements.

The written requirements follow the ISO/IEC/IEEE 29148-2018 [4] syntax. The requirement must have a subject, an action, and a constraint of action. Being a condition or an object is used when necessary [5].

For this case study, 105 requirements were elicited. Due to the page limitations of this work, it is not possible to present the entire list of requirements. However, the list can be found at `https://drive.google.com/file/d/1s6y9BScqp8_8W11SGlSSlfCM9RwB_AxS/view?usp=sharing`.

3 Solutions and Results

This section presents the proposed approach for automating the validation of the Li-ion cell charger using a requirements-driven test architecture in LabVIEW. It details the strategy, test framework design, execution workflow, and measurable results obtained through automation.

The requirements gathered through conversations with individuals involved in the evaluation process of Li-ion cells were organized in a spreadsheet, making it easier for everyone to track development. The spreadsheet has five fields:

- **Id**: Identification of the requirement;
- **Type**: Indicates the type of requirement: functional (F), user interface (UI), or hardware interface (HI);
- **Reference**: It is a reference to what part of the code this requirement is involved; For example: Voltage, current, temperature, hardware, initialization, software, stop, exit;
- **Done?**: It is an indicator to the designer and the manager monitoring how complete the application is regarding the total number of requirements; and
- **Requirement**: The requirement is expressed using the standard ISO-IEC-IEEE 29148-2018 syntax in this field.

The first approach was to design the human-machine interface (HMI) (see Figure 4) based on the compiled discussions with the individuals responsible for operating the application, which were converted into requirements. Table 2 exemplifies some requirements for designing the HMI. The column "done?" was omitted for space's sake.

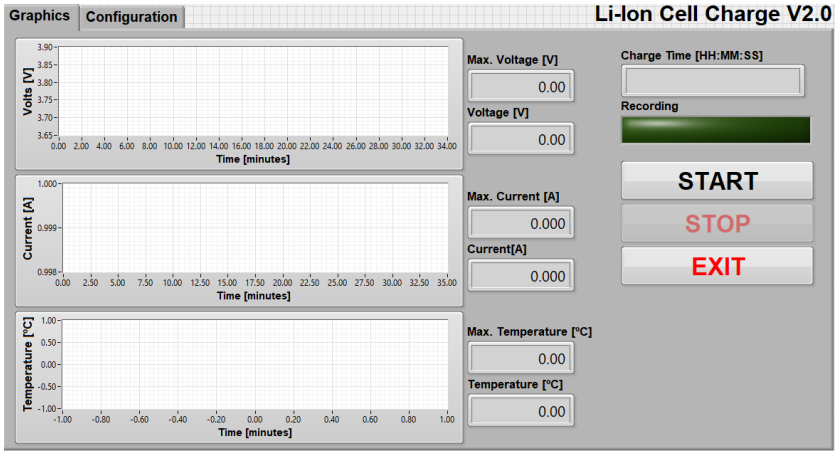

Fig. 4. Prototyped User Interface.

Table 2. Requirements used to design the HMI.

ID	Type	Reference	Requirement
12	UI	Voltage	The monitored voltage shall be displayed graphically on the screen.
28	UI	Current	The numerical form shall be represented with three decimal places of precision.
51	UI	Start	The software shall have a button to start the charge.
55	UI	Stop	The software shall have a button to stop the charge.
72	UI	Time	The software shall display the elapsed charging time on the screen.

Once the application's users approve the HMI, the rest of the development should focus on the internal Li-ion charge requirements. The development takes advantage of the requirement-driven approach by using one of the requirements' inherent characteristics: all the state requirements must be verifiable. Thus, each requirement implemented can be verified, assisting in delivering a complete application regarding the elicited requirements.

The initiative of a requirements-driven approach has led to the development of maintainable code. Additionally, once the requirements are satisfied and linked to specific sections of the code, this approach enhances the documentation and contributes to the overall quality of the work. As previously noted, LabVIEW faces challenges concerning sufficient documentation; therefore, a requirement-driven methodology can provide significant advantages for both users and developers regarding documentation.

By segmenting the program into distinct tasks, we can significantly reduce the likelihood of generating "spaghetti code." This approach allows the programmer to focus on individual components of the code, enhancing overall clarity and maintainability. This, in conjunction with the outlined requirements, ensures the code's completeness. Figure 5 presents the areas where the requirements have been successfully met.

Before this approach, the "requirements" were only verbalized, and when the software was "ready", the individuals realized that new features were needed. Furthermore, the requirement-driven approach leads to an application with test coverage and quality, thus achieving 100% traceability, where each requirement has at least one corresponding test case. Moreover, the automated test can detect latent design issues like minor current overshoots during charge that were not caught in earlier manual tests.

The test output is a tab-separated value (TSV) file that contains information about the test. Listing 1 shows the beginning of the TSV file generated by the application. This output format details all the recorded information about the charge, which can be analyzed or plotted using any suitable application.

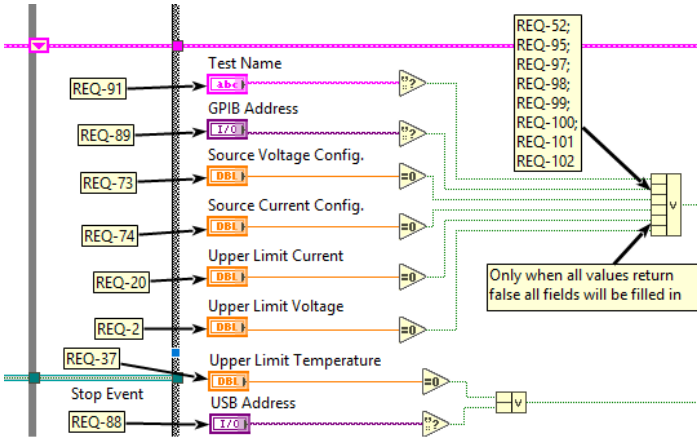

Fig. 5. Block diagram with the requirements comments indicating where the requirements are met.

Listing 1. Format of the TSV file output

```
Test Name:          Cell -3200mA
Date:     07/16/2024
Hour:     11:30
Power supply set voltage:          4.90
Power supply set current:          1.000
Time      Volt [V]  Current [A]    Temperatura [C]
00:00:00          3.84      0.999     25.03
00:00:01          3.83      0.999     25.05
00:00:02          3.82      0.999     25.11
```

4 Conclusion and Lessons Learned

The developed application has effectively monitored and recorded the Li-ion cell's charge voltage, current, temperature, and duration without any malfunctions, improving over earlier efforts that faced clarity and maintainability issues.

The proposed framework enhances testing reliability and maintainability by allowing traceability from high-level requirements to specific features. This methodology integrates requirements engineering into a LabVIEW-based evaluation system for Li-ion cell charging, significantly improving software maintainability and documentation. Linking requirements to code sections reduces technical debt and ensures accurate charging curve analysis.

Although there is no quantitative measure to directly compare the performance of automated testing with manual testing, it is evident that automation allows test operators to focus on other activities rather than monitoring the

entire testing process. This represents an improvement in the efficient use of man-hours, as operators can engage in parallel tasks while running tests.

However, challenges remain, such as LabVIEW's documentation limitations and the complexity of graphical programming. Future research should focus on standardized documentation practices, integrating automated requirement-tracing tools, and extending this methodology to battery performance testing stages.

The automated solution effectively streamlined validation in a safety-critical application by overcoming manual testing challenges like inconsistency and inefficiency, and with further refinement, it can promote broader adoption of requirements-driven development in test automation for improved software quality and lower maintenance costs; this structured, requirements-driven method using modular LabVIEW architecture enhances efficiency, compliance, repeatability, and system robustness, making it adaptable for various test systems where traceability and test coverage are essential.

References

1. Administration, F.A.: Federal Aviation Administration (FAA) Lithium Battery Air Incidents. FAA, edn. (2023), https://www.governmentattic.org/52docs/FAAlithiumBattAirIncidents2023.pdf
2. Conway, J., Watts, S.: A Software Engineering Approach to LabVIEW. National Instruments virtual instrumentation series, Prentice Hall, Professional Technical Reference (2003), https://books.google.com.br/books?id=ICmscdQISHcC
3. FAA: Lithium battery air incidents: Involving smoke, fire, or extreme heat, https://tinyurl.com/FAA-Batteries, accessed on 28 March 2025
4. ISO/IEC/IEEE-29148: ISO/IEC/IEEE 29148-2018: Systems and software engineering - Life cycle processes - Requirements engineering. ISO/IEC/IEEE, edn. (2018), https://ieeexplore.ieee.org/servlet/opac?punumber=8559684
5. dos Santos, C.R., Marques, J.C.: A requirements specification method: An experience report in aerospace. In: WER (2023). https://doi.org/10.29327/1298356.26-20
6. Simões, L.P., dos Santos, C.R., Moraes, A.: A proposed methodology for assessment of li-ion cell suitability and safety for suborbital vehicle applications. Microgravity Science and Technology (2024). https://doi.org/10.1007/s12217-024-10110-2
7. Williard, N., He, W., Hendricks, C., Pecht, M.: Lessons learned from the 787 dreamliner issue on lithium-ion battery reliability. Energies (2013). https://doi.org/10.3390/en6094682