Scientific Research Publishing

Journal of Software Engineering and Applications, 2024, 17, 571-615
https://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

ProTSA: A Testing Process for Automotive Software Domain

Renato Rafael Arcanjo[1], Luiz Eduardo Galvão Martins[1], Dirceu Lavoiser Fernandes Graci[2]

[1]Department of Inovation and Science, Federal University of Sao Paulo (UNIFESP), Sao Jose dos Campos, Brazil
[2]Department of Automotive Electronics, Faculty of Technology of Sao Paulo (Fatec), Santo Andre, Brazil
Email: renato.rafael@unifesp.br, legmartins@unifesp.br, prof.dirceu@yahoo.com.br

How to cite this paper: Arcanjo, R.R., Martins, L.E.G. and Graci, D.L.F. (2024) ProTSA: A Testing Process for Automotive Software Domain. *Journal of Software Engineering and Applications*, **17**, 571-615.
https://doi.org/10.4236/jsea.2024.177033

Received: June 5, 2024
Accepted: July 20, 2024
Published: July 23, 2024

Abstract

This study evaluates the development of a testing process for the automotive software domain, highlighting challenges stemming from the absence of adequate processes. The research demonstrates the application of Design Science Research methodology in developing, an automotive software testing process—ProTSA, using six functional testing modules. Additionally, the study evaluates the benefits of implementing ProTSA in a specific Original Equipment Manufacturer (OEM) using an experimental single-case approach with industry professionals' participation through a survey. The study concludes that combining testing techniques with effective communication and alignment is crucial for enhancing software quality. Furthermore, survey data indicates that implementing ProTSA leads to productivity gains by initiating tests early, resulting in time savings in the testing program and increased productivity for the testing team. Future work will explore implementing ProTSA in cybersecurity, over-the-air software updates, and autonomous vehicle testing processes.

Keywords

Verification and Validation, Automotive Software, Automotive Systems

1. Introduction

The integration of electronics in the automotive industry began in the 1960s, marking significant advancements in electronic and software applications within the sector [1]. This progress has led to an increased availability of safety features in vehicles, including Anti-Lock Brake Systems (ABS), Electronic Stability Programs (ESP), and Electric Power Steering (EPS), all of which are implemented through software embedded in Electronic Control Units (ECUs). Additionally, comfort features such as entertainment and cellular communication are also being made available through embedded software [2].

The application of software development models was necessary to bring organization to the automotive industry and drive the transformation from purely mechanical systems to solutions driven by software.

The literature broadly highlights waterfall software development, incremental development, and reuse-oriented software engineering models as examples used in software development. It's noted that these models are not mutually exclusive and can be used together [3]. In the automotive industry, the Software Development Life Cycle (SDLC) often follows a V-shaped model derived from the waterfall model. On one side of the V-model, the development activities are described, and on the other side are placed the Verification and Validation (V & V) activities [4].

This organization sought to bring a certain organization to the software development sector, however, in the automotive domain there are challenges to be overcome.

A study by the National Highway Traffic Safety Administration (NHTSA) of the U.S. Department of Transportation analysed data between 1966 and 2023 and found that the first software recall occurred in 1994. After that date, there have been over 1000 recalls involving software that potentially affected over 70 million vehicles. In the year 2023, 15% of recall processes were related to software [5].

Analysing the perspectives on solution implementation and challenges in applying software in the automotive domain, a software functional testing process was developed due to absence of available and structure steps for the V & V teams, risks of using only tacit knowledge and specific lose this knowledge in case a specific tester leave out the Original Equipment Manufacturer (OEM) studied in this work. Its aim is to enhance the functional testing process for a specific Original Equipment OEM in the automotive industry due to the involvement of some participants of this work in V & V process for related ones. The specific objectives include increasing automotive software quality, optimizing testing resources, boosting productivity during testing, and improving the overall effectiveness of the automotive software testing process.

2. Background and Related Works

2.1. Definitions

2.1.1. Automotive Software

The automotive industry is increasingly investing in innovative solutions through embedded automotive software, which can create new business opportunities and revenue streams for companies [6]. Automotive software is characterized by specific traits, such as its heterogeneous nature, numerous communication processors, management of various vehicle variants, and specific safety and reliability requirements [7]. Additionally, automotive software often manages a vehicle domain but interacts with other domains like human-machine interface, powertrain, brakes, and telematics. Changes in automotive software requirements also affect interacting domains, emphasizing the need for seamless integration for comprehensive vehicle testing and approval. [1] highlights the main differences between automotive software and laptop/mobile software, particularly focusing on reliability, functional safety, and real-time behaviour. Automotive software must exhibit extreme reliability throughout a vehicle's lifecycle, especially concerning functions like ABS and ESP, which must be fail-safe as they are activated when the driver is at risk. Moreover, real-time behaviour is critical, as seen in the gearbox's software needing to execute gear change strategies within milliseconds to prevent mechanical damage to the vehicle's powertrain or compromise its functional safety.

2.1.2. Software Testing Motivation

The software development process is a complex activity, which demands many hours of effort from different professionals with different degrees of knowledge and requires a lot of skill and interpretation in building the software. That is, there is the possibility of introducing errors in the software development process and to minimize the risks of these errors remaining in the software when delivered to the customer, a software testing process, also called V & V is performed [8].

2.1.3. Static and Dynamic Testing

The V & V activities in software testing, according to [8], can be divided into two main groups. The first group, called static testing, evaluates software using functional models or specifications without the need for the software to be operational. The second group, dynamic testing, requires functional software to evaluate it dynamically.

2.1.4. Type Approval and ISO 26262/ASIL

Type approval is a crucial process that verifies whether a software, system, or vehicle meets the minimum requirements set by legislation and is approved for market launch and operation [9] [10]. In the automotive field, there are two types of approvals as outlined by [11]: Third-party certification, where a certifying entity verifies the product against specified requirements, and self-approval, where the OEM itself approves the product and provides technical evidence to the certifying entity.

This research also incorporates the concept of ISO 26262 for functional safety in automotive systems, as discussed by [12]. ISO 26262 standardizes automotive functional safety, focusing on electrical and electronic components (E/E). The standard introduces the Automotive Safety Integrity Level (ASIL) classification system to categorize hazards related to E/E systems. ASIL levels range from A (lowest risk) to D (highest risk), with ASIL D requiring the most stringent safety measures due to its high-risk probability to users and bystanders [13].

2.2. Related Works

For the development of this work, a systematic literature review (SLR) entitled

"VERIFICATION AND VALIDATION OF EMBEDDED SOFTWARE IN AN AUTOMOTIVE CONTEXT: A SYSTEMATIC LITERATURE REVIEW" was carried out and published by [14] and within the established selection criteria, two SLR were selected, one on software verification and another on the validation process for automotive engineering. A third work related to the objectives of the RSL was identified and added in the conduction of the RSL, as it was cited in several other articles that discuss and define concepts for Automotive Software Engineering (ASE).

The first SLR developed by [15] investigates the current V & V requirements and regulations, the weakness of the current process of testing and challenges and opportunities to ensure the safety of autonomous driving vehicles. In this review, databases such as SpringerLink, IEEE Xplore, ACM Digital Library and Wiley were consulted. After applying the Inclusion Criteria (IC) and Exclusion Criteria (EC), two hundred articles were obtained. It has been identified a substantial number of types of testing techniques, frameworks, and philosophies.

It was concluded that the ISO 26262 standard is not ideal for the V & V process of autonomous vehicles. This is because the requirements of an autonomous vehicle cannot be defined when test cases are initially defined because the behaviour of this type of vehicle is dependent on the environment in which it operates. This means that the vehicle's response differs depending on the boundary conditions, which are not covered by ISO 26262. In SLR, there is a consensus that the fault injection method can encounter many faults in different environments and that a formal method of testing and mutation testing may be required for the V & V process of autonomous vehicles. In summary, [15] concluded that it is not possible to define all the requirements for the V & V process of autonomous vehicles because their response depends on the environment in which the vehicle is operating and cannot be predicted. This is one of the biggest challenges for testing this type of vehicle and for the overall development process. The ISO 26262 should be updated to accommodate guidelines for the development of autonomous vehicles.

[16] carried out a second SLR to identify the research activity intensity, most common topics, research method types, contribution of studies, challenges, and promising future in the Automotive Software Engineering (ASE) domain. They collected 679 articles from various areas of research. At SLR it was demonstrated that the literature on Automotive Software Engineering (ASE) in the 1990s increased substantially. After that decade, scholarly interest waned, although it revived from 2007 onwards. After that date, research interest in ASE increased constantly, and they also observed an evolution in the topics of the articles over the years. In the 1990s, the most common topics related to the compliance support group, project support group, and requirements engineering. After 2007, the most common topics were systems/software architecture and design, followed by systems/software testing. Therefore, the most common types of studies published were evaluation surveys, which typically involved implementing a proposed solution and presenting the results obtained through case studies or

surveys. In relation to the most relevant topics, proposals and lessons learned were structured, which highlighted the importance of good management of the development and testing of the ASE. Although the authors have demonstrated that the ASE theme gains relevance in the academic environment, such studies often do not consider the practical applicability from the industrial point of view. Therefore, they concluded that topics such as V & V software should have empirical evaluations of the proposed methods under real practical conditions that can happen in the industry domain.

A third work added to this SLR was published by [7], which investigates the characteristics of Automotive Software. They described several features of ASE and distinguished between regular software and automotive software. A characteristic of automotive software is its heterogeneous nature due to different functionalities, e.g. powertrain, infotainment, driver assistance. This heterogeneity results in a lengthy system integration process, many vehicle variants with which the software needs to be compatible, and specific reliability and safety requirements.

These studies by [15]-[17] provided a good overview and were a good reference for the development of a testing process for automotive software domain. These studies highlighted gaps in standards for self-driving cars, the necessity for more research in ASE, and distinctions between automotive and regular software. However, other methodology techniques were applied in the development of this work and will be presented in the following sections.

3. Methodology

3.1. Research Questions (RQs)

The utilization of software to facilitate technologies in a flexible manner is prevalent across various sectors, notably within automotive industries. Nonetheless, there has been a rise in the number of vehicle recalls, particularly associated with software issues. This trend underscores the absence of structured and efficient processes for early defect detection.

Table 1, contains the research questions (RQs) and their corresponding objectives for developing a software testing process in the automotive domain.

Table 1. RQs of software testing process development projects in automotive domain. Source: Authors.

ID	Research Questions	Objective
QP1	What software testing techniques and strategies in the automotive environment improve software quality?	Identify techniques and strategies that, applied to software development and testing, improve its quality
QP2	How does the proposed testing process improve the productivity of automotive software testing teams?	Identify the benefits of improving a testing process linked to team productivity
QP2.1	What is the time saving?	Identify technique to measure productivity gains

Continued

QP3	Does the proposed testing process improve the effectiveness of defect detection?	Understand what are the current factors that hinder the effectiveness of an automotive testing process

3.2. Design Science Research Methodology

The methodology employed in this work to propose alternatives for addressing the increase in recall processes and late detection of software defects in the automotive domain is depicted in **Figure 1**, following the flow of the Design Science Research methodology by [17].

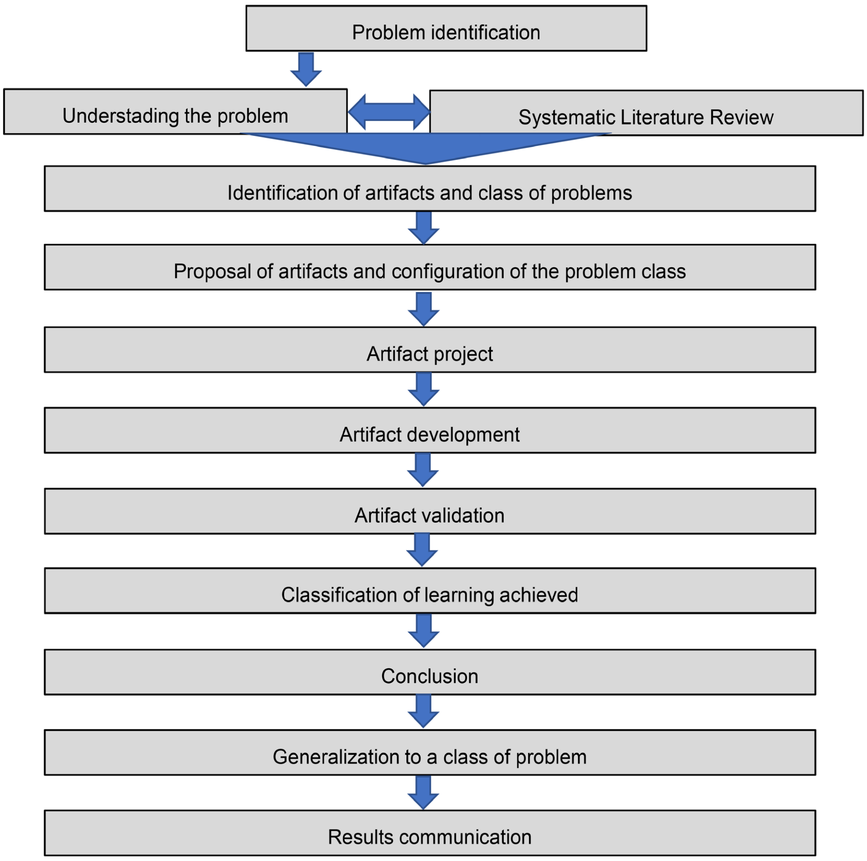

Problem identification
Understading the problem
Systematic Literature Review
Identification of artifacts and class of problems
Proposal of artifacts and configuration of the problem class
Artifact project
Artifact development
Artifact validation
Classification of learning achieved
Conclusion
Generalization to a class of problem
Results communication

Figure 1. Design science research methodology flow.

Design Science Research methodology was applied to identify and demonstrate the problem initially in Section 1. This involved conducting a Systematic Literature Review (SLR) to understand the artifacts used, based on data extracted from the RSL and discussions with specialists. The study progressed to map the stakeholders, as outlined in **Table 2**, to comprehend their roles and duties. Furthermore, the desires and objectives of the stakeholders were delineated, as shown in **Table 3**. Through mapping process facilitated the definition of project requirements, classification of problems, commencement of artifact development, application of an evaluation artifact, and subsequent classification and explanation of results, contributing to the dissemination of knowledge on the subject.

Table 2. Describes the stakeholders mapped in the project, as well as their main duties and responsibilities. Source: Authors.

ID	Stakeholders	Assignments	Responsibilities
1	Automotive Software Testers	Perform validation tests and support methodology development	Validation and Execution
2	Automotive Software Developers/Suppliers	Evaluate the benefits that the methodology brings to automotive software development	Validation
3	Coordinators of functional areas related to the testing and development of automotive software	Verify if the methodology brings technical and financial benefits to the department	Validation
4	Drivers Testing Vehicles with Embedded Software/Owners	Evaluate the result of the maturity of the released software, after testing with the elaborated methodology	Validation

Table 3. Describes the desires and goals of stakeholders mapped in the project. Source: Authors.

ID	Stakeholders	Wishes
1	Automotive Software Testers	Clear Steps for Performing Software Testing
2	Automotive Software Developers/Suppliers	Early feedback of software test results
3	Coordinators of functional areas related to the testing and development of automotive software	Elimination of software defects, report of software tests delivered on time and carried out at low cost
4	Drivers of Testing Vehicles with Embedded Software/Owners	Reduction of maintenance downtime due to software failures

Finally, the conclusion encompassed the generalization of results to a problem class, followed by communication of the findings, which will be elaborated in subsequent sections.

3.3. Design Science Research Methodology

After mapping the responsibilities, attributions, and desires of stakeholders in the automotive domain, an artifact design for an automotive software testing process was developed. This design encompasses a set of activities with defined inputs and resources, serving as a reference for ensuring a software V & V process specific to the automotive context. To make the Automotive Software Testing Process (ProTSA) available, a process specification is developed. In the specification, the ProTSA modules were detailed: Login, Config Menu_ and test steps, 1 - Integration between functions_FC & FS, 2 - Integration between software, 3 - Integration between CAN signals, 4 - Parameter integrity, 5 - Maturity of ECUs_SW_functions and 6 - Fail Safe Test, in order to make ProTSA available in the online format of ProTSA for easy access and expert evaluations. Afterward, a website was implemented to enable testing and documentation in a more agile and easily accessible way, which is available at https://protsa.infocept.com.br/.

3.4. Design Science Research Methodology

The evaluation of ProTSA involved the creation and execution of a software test case for integrating functionalities between two ECUs, specifically the ABS and Chassis ECUs. The test case was distributed to specialists in electronic format for execution. Subsequently, the specialists were asked to provide feedback by completing a survey provided by the authors of this work. To assist in the evaluation process, an online tutorial was created and made available on the YouTube platform, covering each of the ProTSA modules. You can find the tutorial at the following weblink:
https://www.youtube.com/playlist?list=PLH43-llgcycsv3bKLeygtAao3nr9h9-4X.

Subsequently, e-mails were sent to eighteen professionals linked to software V & V processes in the automotive domain and who have technical knowledge and contributions in the automotive area to evaluate the ProTSA.

3.5. Results and Analysis

In this study, a set of questions was administered using the Likert scale, a widely used method for gauging individuals' satisfaction or opinions on various subjects such as products, processes, or games. The Likert scale, devised by psychologist Renis Likert in 1932, typically consists of scales ranging from 5 to 9 points. The odd number of points ensures a neutral option, with responses ranging from strongly disagree to strongly agree [18].

The Likert scale used in this study employed a 5-point scale with positive statements, where scale 1 indicates "Strongly disagree," scale 5 denotes "Strongly agree," and scale 3 represents a neutral position— "Neither agree nor disagree." These questions aimed to gather participants' opinions on aspects such as resource optimization, productivity, quality, effectiveness, clarity, applicability, and inconsistencies within the ProTSA framework. The results measured on the Likert scale will be used to create graphs alongside mean and standard deviation calculations, providing a visual representation of participants' perceptions regarding the objectives of the research, learning outcomes, and overall completion of the study phases.

3.6. Threats to the Validity of Results

The potential sources of threats identified in the results refer to the nature of the domain studied. Since the area of study was the field of Software in the automotive area, a test case and survey were applied in the domain of Brakes and Control of commercial vehicle chassis. Therefore, generalizations will only cover the domains mentioned. In addition, the ProTSA evaluation invitation was sent to eighteen experts from a multinational manufacturer of commercial vehicles and working with software V & V steps, to which a total of fourteen experts answered and four experts did not respond to the invitation.

Since this is research in a specific sector and domain, the results should be interpreted as a representation of that sector and domain mentioned. However, particularities inherent to the company's process and different contexts may cause variations. In addition, the level of knowledge of each participant is a threat to the validity of the results. Although all participants are active in software V & V processes, some are more familiar with some of the steps proposed in ProTSA than others, and this can influence different perceptions of the process and suggestions for improvement.

Therefore, the results obtained offer a view of the use of ProTSA in the domains of Brakes and Chassis Control in an automotive environment and based on the level of knowledge of the participants who responded to the survey. In short, for future applications, users are expected to interpret the results and consider the limitations and degree of knowledge of the participants in this work.

4. Development

4.1. Idealization of ProTSA

After conducting and publishing a systematic literature review (SLR), the authors [14] focused on software testing within the automotive domain, results were obtained to understand methodologies, test types, regulatory frameworks, and challenges prevalent in the automotive industry. Subsequently, individual meetings were conducted with ten professionals from various Original Equipment Manufacturers (OEMs) with diverse experience levels in software development and testing within the automotive sector. The primary goal was to identify the key challenges specific to the OEM's processes. The study revealed that most tests conducted were functional tests, which validate software outputs against predefined inputs, and application tests, which assess the software's compatibility across different vehicle versions and external environments.

The mapped and detailed difficulties related to the current automotive software testing process in the OEM under study are presented in **Table 4**.

Table 4. Describes the difficulties in relation to the current automotive software testing process in OEM. Source: Authors.

ID	Difficulties	Stakeholder Concerns
1	Integration Between Functions	How to integrate functions and ECUs that are interconnected?

Continued

2	Software integration of ECUs at different stages of development	How to do *software* integration between ECUs that are in different stages of development?
3	CAN Messaging Integration	How to deal with CAN message integration issues?
4	Parameter integrity	How to ensure the integrity of the parameters of the ECU's that will be released in series?
5	Maturity of ECUs/*software*/ functions for series	How to identify that an ECU/*software*/Function is at a suitable maturity to be released serially?
6	Unrepresentative prototypes	How to deal with the lack of representativeness of prototypes?
7	Software release management	How to manage a software release process?
8	Prioritization of tests and development of SW/ECUs/functions	How to identify which specific ECU demands prioritization in testing and development?

4.2. Objectives of ProTSA Modules

In the next step, discussions were held with advisors to determine the prioritized difficulties for creating the ProTSA concept and identify which challenges would be addressed within the ProTSA project. The authors outlined six steps or modules to face some of the identified difficulties. Each module is detailed in Table 5.

Table 5. Describes the modules and their objectives. Source: Authors.

ID	Step/Module	Objective
1	Integration between functions	Enable properly operation between distinct functions that exchange messages with each other.
2	Integration between software	Enable the integrated operation between different ECU software that interact with each other.
3	Integration between CAN signals	Analyse in advance the availability of the CANs messages necessary for the correct interaction between ECUs.
4	Parameter Integrity	To ensure defect-free serial production of different vehicle variants with a single software and ECU by applying a separate set of parameters to each vehicle version.
5	Maturity of ECUs	Evaluate the maturity of the product, software, and vehicle to finish the execution of the tests and release them for serial production.
6	Fail Safe test	Monitor the progress of solutions to failures, errors or defects of software applied to vehicles and define priorities for the allocation of workforce for the progress of the software V & V.

The ProTSA concept aims to address at least six difficulties highlighted by experts in automotive software testing. The first module, "Integration between functions," focuses on resolving integration challenges by proactively identifying and prioritizing the requirements necessary for functions to interact correctly during testing and development.

The second module, "Integration between software," is designed to map interaction needs using block diagrams, enhance visualization of test and development scenarios, track function releases across different software versions, and facilitate test alignment through regular meetings, promoting transparency during testing processes.

The third module, "Integration between CAN signals," targets challenges related to the availability of CAN signals during functional tests, often discovered late in the project's lifecycle, typically in prototype vehicles. This module aims to cross-reference information to determine the availability of each CAN signal, helping to address issues encountered during testing.

In the fourth module, "Parameter integrity," the goal is to guarantee that vehicles are manufactured with parameters that do not lead to failures during production or in operational use. This module involves setting test priorities, creating spreadsheets for user-loaded parameters, suggesting new parameters for testing or release, and establishing workshop schedules for parameterization and documentation tests.

In the fifth module, "Maturity of ECUs," the objective is to evaluate the maturity of software within Electronic Control Units (ECUs). This module fosters information exchange on test results among the automaker, suppliers, and testers. It also provides percentage-based information and bar graphs to assess the current progress and maturity level of ECUs.

Finally, module 6—Fail Safe Test aims to help identify failures in advance, map the solutions found and applied, and serve as a reference for prioritization and allocation of resources through graphs of failures by ECUs or vehicles.

To enable the online testing process to be more agile to understand and evaluate by experts, ProTSA was implemented through a website, and made available at https://protsa.infocept.com.br/. In addition, through the administrator environment that will be presented in the following sections, the possibility of configuring the *online* environment for the needs of different OEMs will be demonstrated.

4.3. ProTSA Development, Project Registration, and Users

On the ProTSA online platform, an environment was created for user registration, as shown in **Figure 2**.

Subsequently, a step was developed for the registration of a project in the ProTSA environment (as shown in **Figure 3**). For this purpose, the main information in the test project for each ECU was added. Additionally, in this step, you can also select the ECU (**Figure 4**), function (**Figure 5**), network type data (**Figure 6**) and other data, such as CAN signals and parameters.

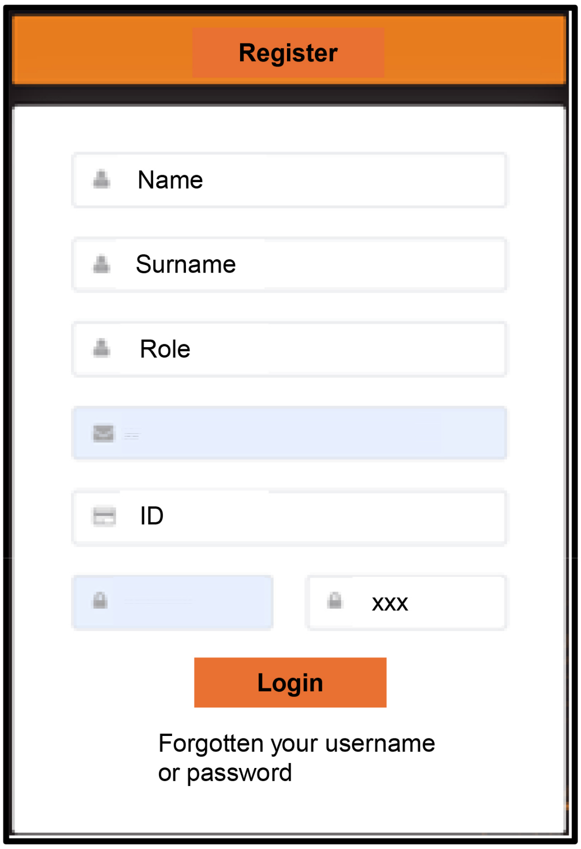


Figure 2. Website user registration for ProTSA.

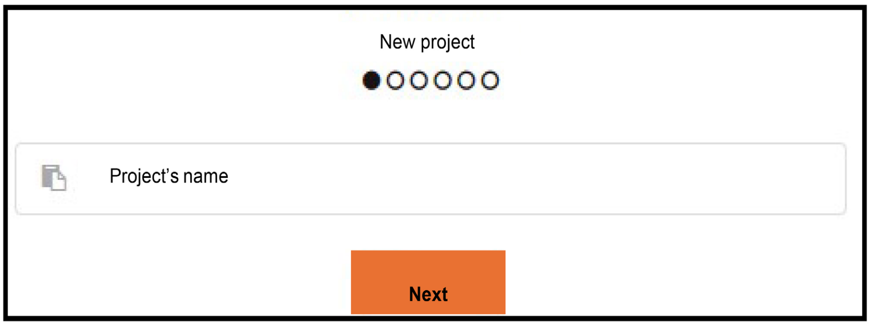
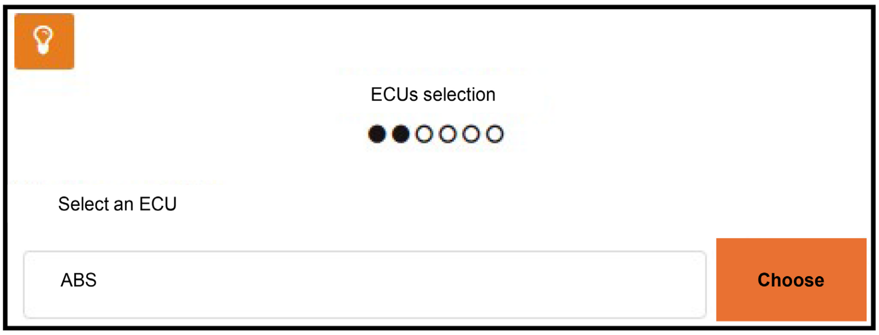


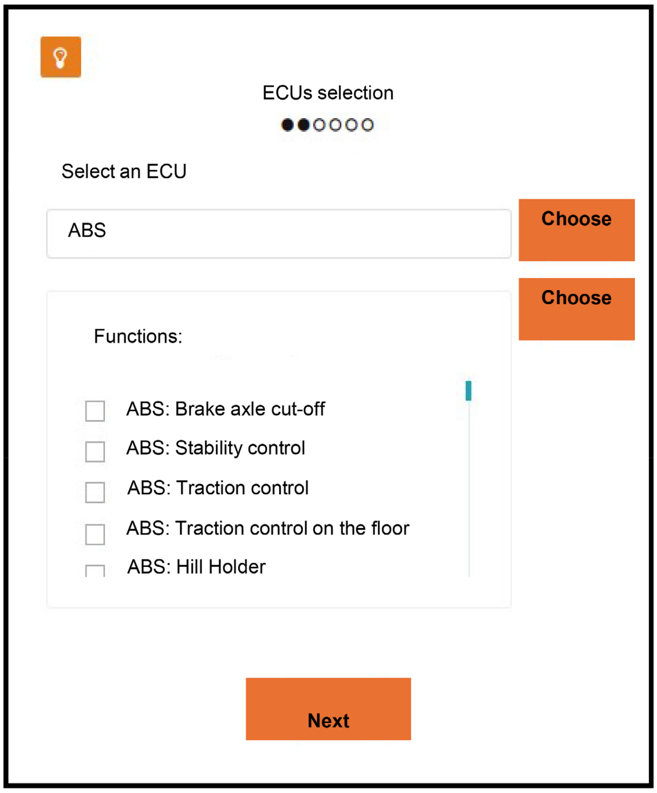
Figure 3. Project name creation in ProTSA online.



Select an ECU

ABS

Figure 4. ECU selection in ProTSA online.



ECUs selection

●●○○○○

Select an ECU

ABS

Functions:

□ ABS: Brake axle cut-off

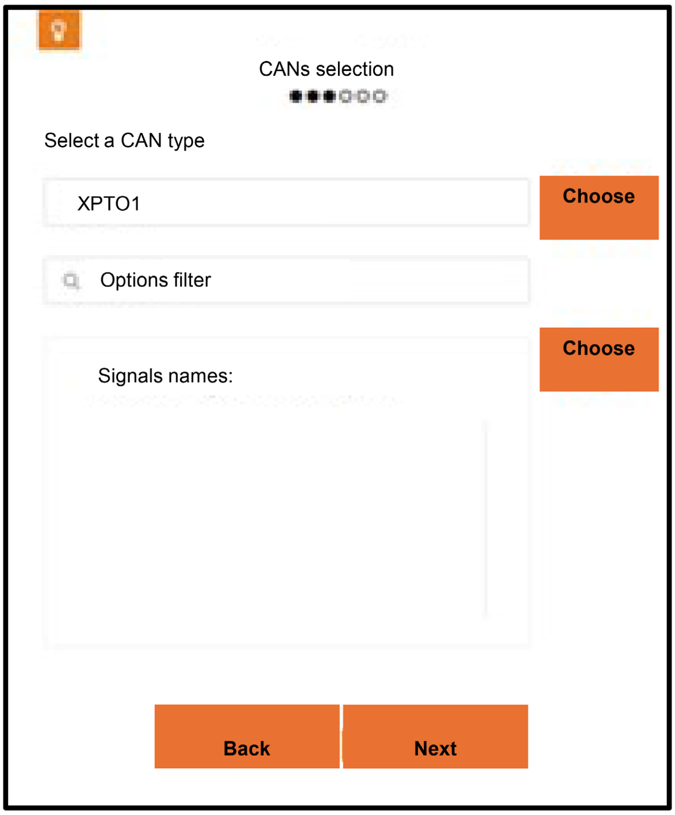
□ ABS: Stability control

□ ABS: Traction control

□ ABS: Traction control on the floor

□ ABS: Hill Holder

Figure 5. Selection of function in ProTSA online.



CANs selection

●●●○○○

Select a CAN type

XPTO1

Options filter

Signals names:

Figure 6. Data network type selection in ProTSA online.

4.4. ProTSA Environments

In ProTSA, there are two environments. The first, named the administrator environment (**Figure 7**), allows access to update the list of ECUs, CAN networks, parameters, and add new vehicle portfolios. This access also enables configuring weights for questions used to classify functions in the software test project, making ProTSA more adaptable to different projects.

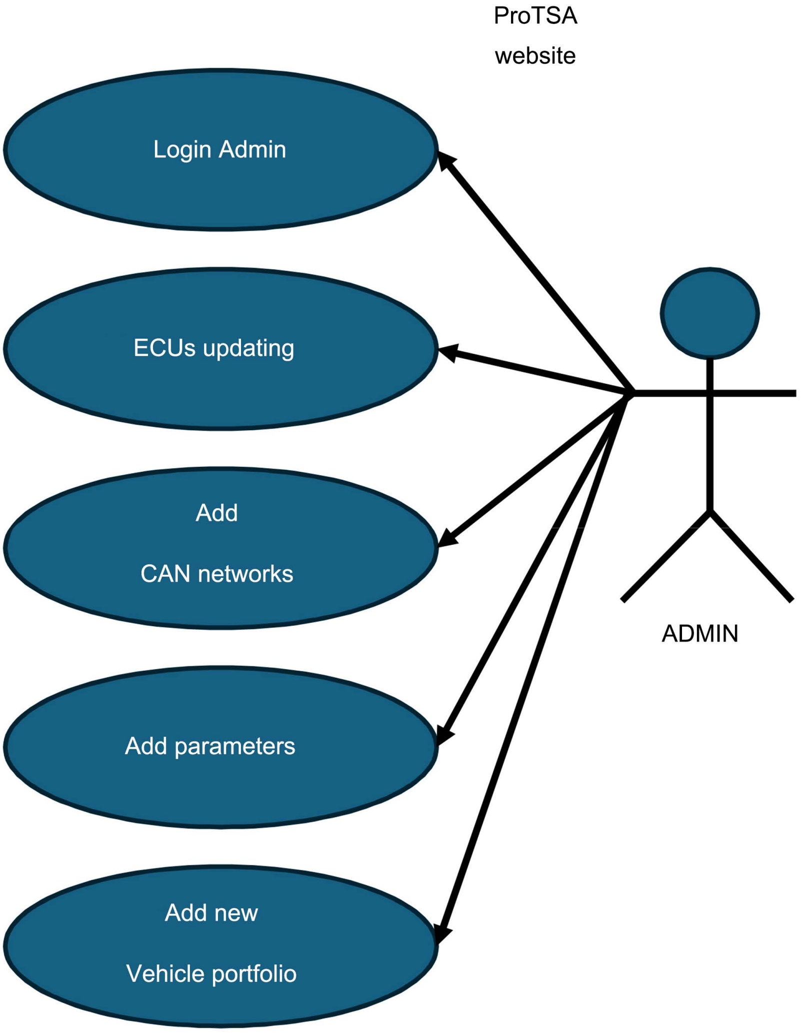


Figure 7. Diagram of connecting actors with use cases for Admin access.

In the second environment of ProTSA, users responsible for conducting or coordinating tests can access six different modules after registering on the website and creating a project. Additionally, there is a diagram of use cases and connection of actors provided for standard user access, as depicted in **Figure 8**.

After the creation of the user and registration of a project, the tester will be given the possibility to navigate through the six modules in a sequenced way, that is, starting with module 1-Integration between functions and ending the tests in module 6-Fail Safe.

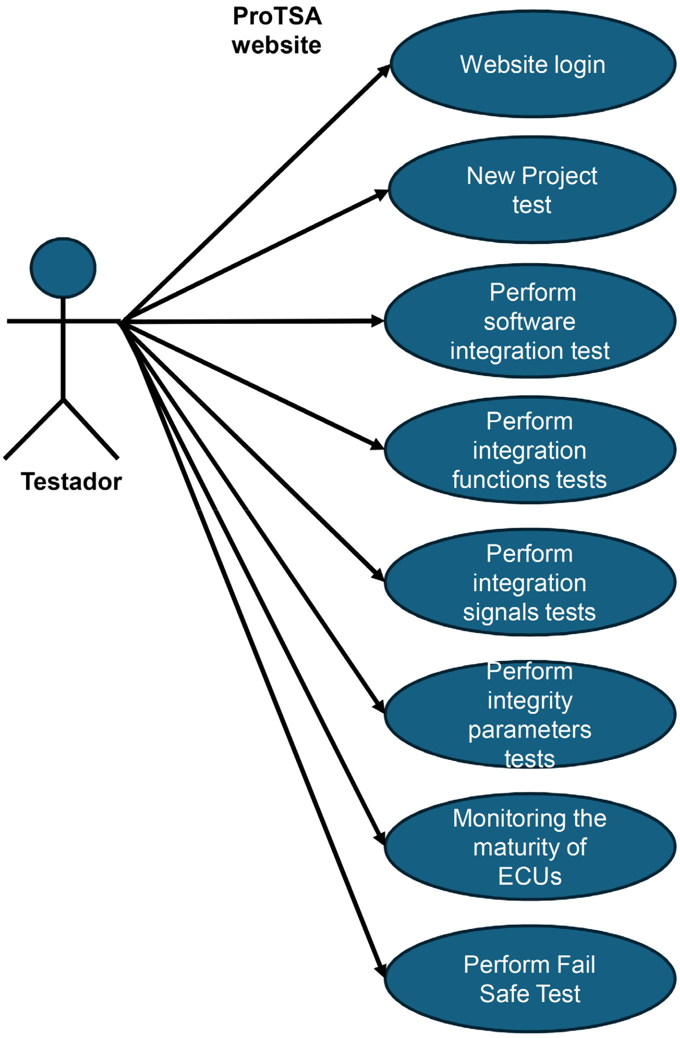

ProTSA website
Testador
Website login
New Project test
Perform software integration test
Perform integration functions tests
Perform integration signals tests
Perform integrity parameters tests
Monitoring the maturity of ECUs
Perform Fail Safe Test

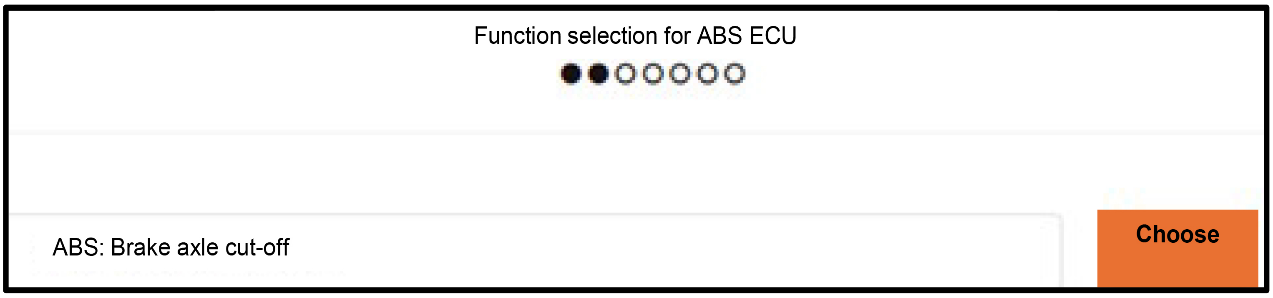
Figure 8. Diagram of connecting actors with use cases for user/tester environment access.

4.5. Integration Module between Functions—Module 1

The main objective of the integration between functions module—module 1 is to enable the structuring of functional tests for integration between the functions of at least two ECUs to avoid problems such as lack of documentary information, unavailability of CAN signals, lack of integration test cases and failure in communication between the stakeholders involved in the tests.

For the purpose, this module was subdivided into four main stages: documentation analysis, CAN signals/messages, classification of client function and service function, test diagram and meetings.

After creating, selecting the project, and accessing the function integration module, the user is asked to select the required data from the first ECU involved in the test, then the selection of the ECU function (**Figure 9**) and at the end will be asked to upload the function documentation (**Figure 10**). It is recommended that the document uploaded to the site is a specification of the function or at least a draft of it. This step will be repeated for the second ECU to collect documents from both ECUs to perform a static test, which will be evaluated in a working group, through meetings to evaluate the proper integration of both ECUs and functions. Therefore, this step will be responsible for discovering, in a more agile way, problems of interaction between ECUs and promoting the integration at the beginning of the project of *the* stakeholders necessary for the success of this test.

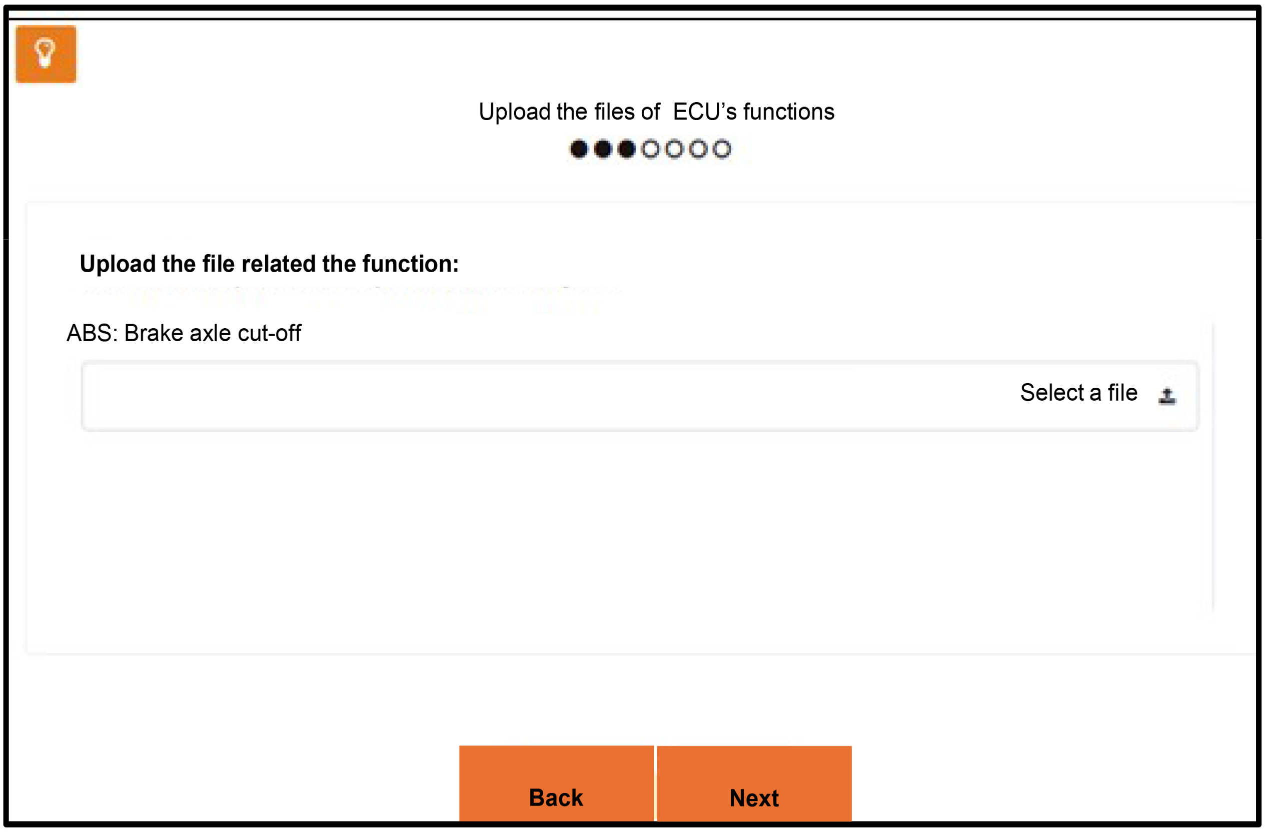


Function selection for ABS ECU

●●○○○○○

ABS: Brake axle cut-off

Figure 9. ECU function selection in ProTSA online.



Upload the files of ECU's functions

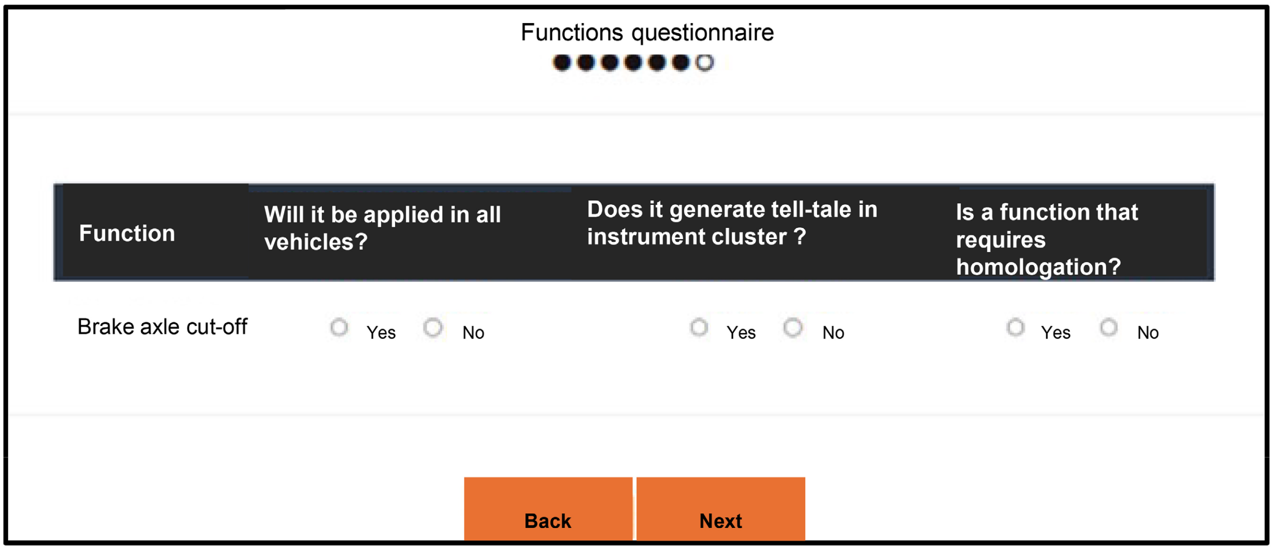
●●●○○○○

Upload the file related the function:

ABS: Brake axle cut-off

Figure 10. Environment to load the function's documentation.

In the next step, information from the specific ECU and function will be requested over questionnaire (Figure 11), to later cross-reference the information from each of the ECUs and generate information on compatibility and availability of the information.



Functions questionnaire

●●●●●●○

Function	Will it be applied in all vehicles?	Does it generate tell-tale in instrument cluster ?	Is a function that requires homologation?
Brake axle cut-off	○ Yes ○ No	○ Yes ○ No	○ Yes ○ No

Figure 11. Online ProTSA Questionnaire for functions.

At this step of the questionnaire for the functions, the user will be asked to answer three questions for each function inserted in the project and, for each question, there is a different weight that can be decided at the beginning of the project by the project participants.

After the questionnaire is answered, the website will make the calculations based on the answers and will rank points, in which the function that obtains the most points will be called the customer function. The term customer function is used for a function in which the end customer can observe or perceive its operation, such as the activation of the hazard warning light during emergency braking, also known as the Emergency Stop Signal (ESS) function, (Figure 12). However, the classification for the function with the fewest points will be given, as a service function, which means that this is a function that provides data for another function to perform an end activity that is perceptible to the end user. In this case, the service function would be a panic braking detection function to later activate the hazard lights via the ESS, an example of a service function is panic braking detection by means of the rapid application of braking to the brake pedal on the vehicle (Figure 13).

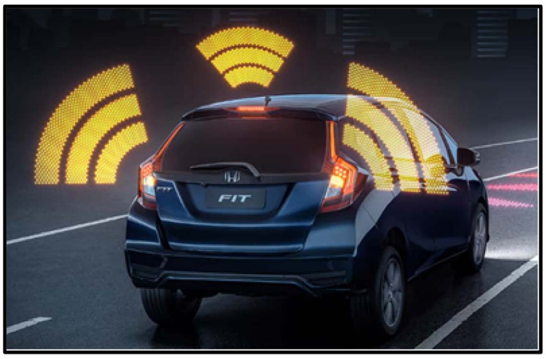


Figure 12. Customer function—Honda emergency stop signal (2016).

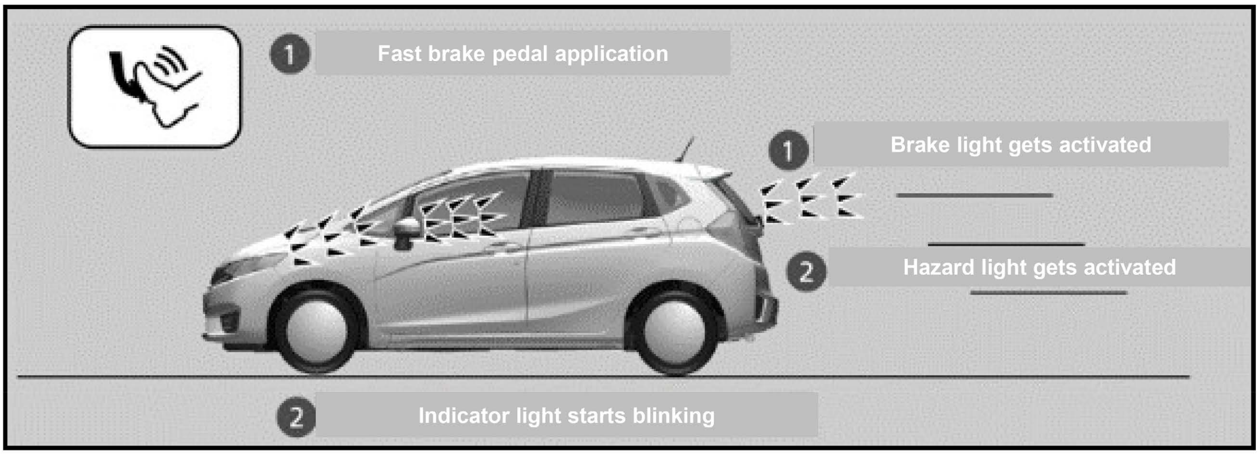
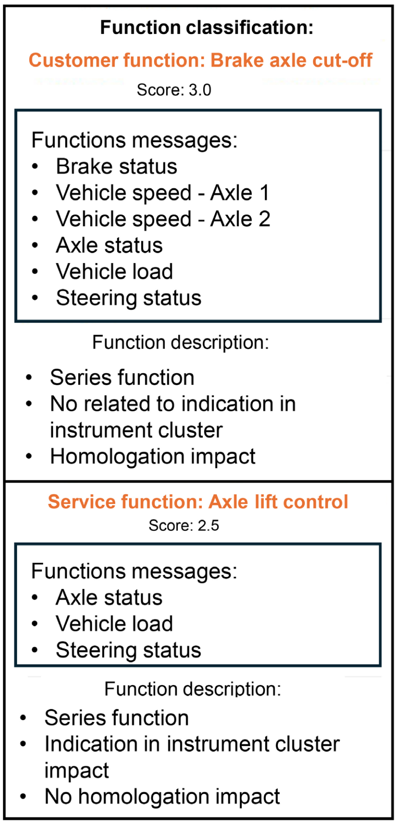

1 Fast brake pedal application
1 Brake light gets activated
2 Hazard light gets activated
2 Indicator light starts blinking

Figure 13. Service Function—Panic detection by brake pedal (2016).

After filling in the project information and answering the questionnaire about the functions, the classification of client and service functions will be made available on the ProTSA online platform (website) (Figure 14). This classification occurs after the website ranks the users' responses to identify functions that lead the development or integration tests, which, in this case, is called the client function, that is, a function that is visible to the end customer, which, in the case of Figure 14, represents the Brake cut-off function on the axle.



Function classification:

Customer function: Brake axle cut-off

Score: 3.0

Functions messages:

- Brake status
- Vehicle speed - Axle 1
- Vehicle speed - Axle 2
- Axle status
- Vehicle load
- Steering status

Function description:

- Series function
- No related to indication in instrument cluster
- Homologation impact

Service function: Axle lift control

Score: 2.5

Functions messages:

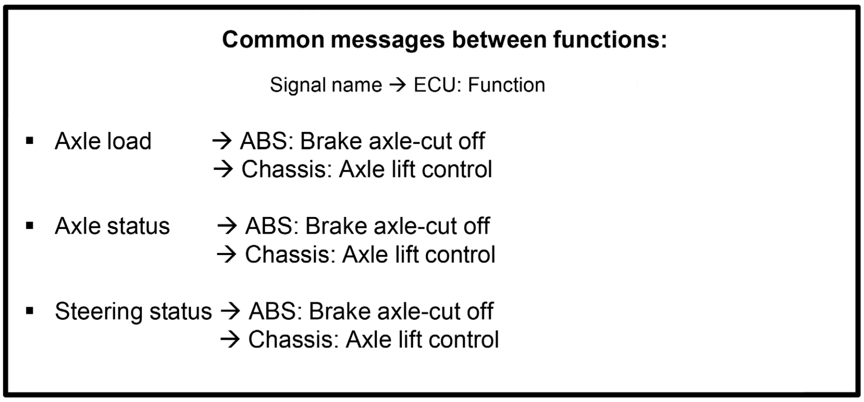
- Axle status
- Vehicle load
- Steering status

Function description:

- Series function
- Indication in instrument cluster impact
- No homologation impact

Figure 14. Classification of client and service functions in ProTSA online.

The output of the first result in ProTSA includes presenting common CAN messages between functions, which is relevant for developers and testers to understand the connection between functions and ensure correct integration. This information is depicted in Figure 15, showing common messages between Axle Brake cut-off and axle lift control functions.



Common messages between functions:

Signal name → ECU: Function

- Axle load → ABS: Brake axle-cut off
→ Chassis: Axle lift control
- Axle status → ABS: Brake axle-cut off
→ Chassis: Axle lift control
- Steering status → ABS: Brake axle-cut off
→ Chassis: Axle lift control

Figure 15. Common messaging between client and service functions in ProTSA online.

ProTSA was designed with communication as a crucial factor for the success of function integrations, testing, and development, as highlighted by [19]. Effective communication among stakeholders helps avoid project rework. Users are recommended to schedule meetings through the ProTSA website, where meeting participants can add notes for follow-ups and schedule additional meetings if necessary.

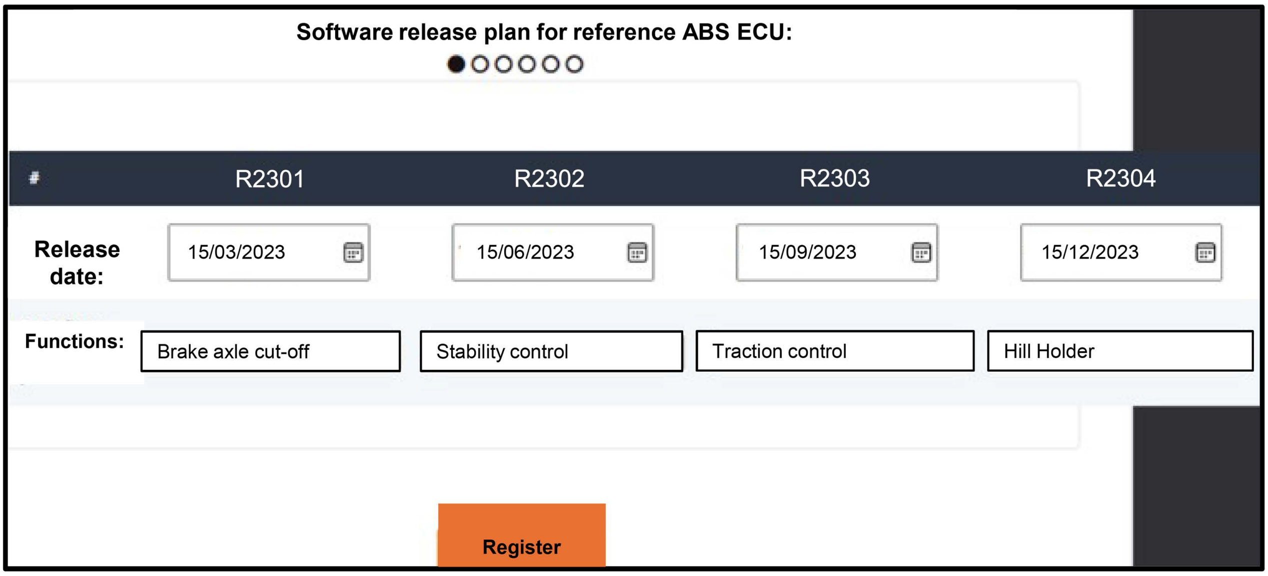
In the last step of the function integration module and based on the analysis of literature, the ProTSA integrates visual communication methods, such as block diagrams, for analysing software requirements and planning software tests, as recommended by [19] [20].

4.6. Software Integration—Module 2

In the software integration module (module 2) of ProTSA, the main objective is to plan software integration tests between at least two ECUs. This involves creating a plan focused on interaction block diagrams between the software, resources needed for the tests, release plans for each software function, and a communication plan.

The first step in software integration is to create a block diagram to visualize the interface between software components in a simplistic way.

The second step of this module, involves creating a plan for the release of ECU software by breaking it down into functions. The proposal in ProTSA subdivides dates into four periods of the year, during which functions are made available for testing. In ProTSA, the four phases of release are depicted in Figure 16, with examples like R2301, R2302, R2303, and R2304. The letter R means release and number 23, is related to the test year, which, in the figure, is 2023, and the sequence between 01 and 04 constitutes the order of the plan for releasing the software's functions. In the online environment, as shown in Figure 16, mentioned above, it will also be possible to register the date of each release and the description of the function.



Software release plan for reference ABS ECU:

●○○○○○

#	R2301	R2302	R2303	R2304
Release date:	15/03/2023	15/06/2023	15/09/2023	15/12/2023
Functions:	Brake axle cut-off	Stability control	Traction control	Hill Holder

Figure 16. The software release plan is broken down into functions in ProTSA online.

The next phase of this module is a step to check the availability of resources needed to run tests, such as software licenses for measuring CAN data, parameterization of ECUs, and availability of proving grounds. This step aims to prevent wastage of time and budget by ensuring that all necessary resources are available before test execution.

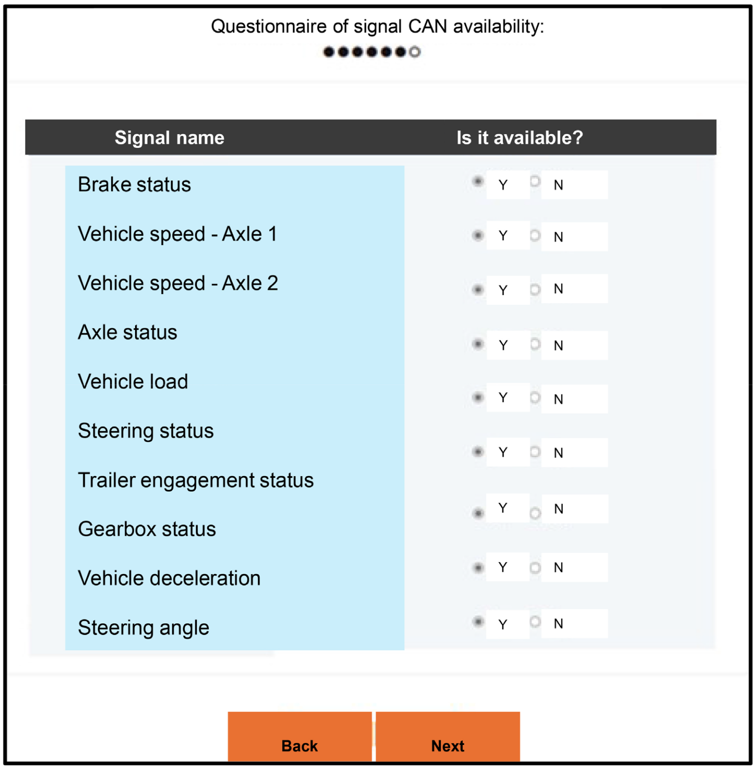
After users register the information from the ECUs and store the block diagrams in ProTSA online, it is suggested that the user use the block diagram previously uploaded to ProTSA online to align stakeholder needs. Additionally, as a result, it is recommended to use the software release plan to carefully plan the sequence of software integration tests. Finally, as in the previous module, the importance of communication for the positive outcome of software V & V was addressed. Following the strategy of the importance of alignments, in this module, ProTSA also recommends scheduling meetings to discuss the information collected and establish plans for software integration tests.

4.7. CAN Signal Integration—Module 3

For the CAN signal integration module—module 3, the primary goal is to proactively identify the absence of CAN signals during software testing in a prototype vehicle. This module is structured into steps for gathering availability information and conducting comparisons of signal availability.

At this step, ProTSA suggests verifying the requirement for CAN signals for each function beforehand through documentary verification, aligning with static testing principles as described by [8] and [21]. This test will be conducted based on the analysis of CAN signal information, which does not necessitate dynamically operational software. Consequently, this test serves to proactively identify potential software failures and mitigate development costs in a software project.

Figure 17 and Figure 18 illustrate the collection of information concerning the availability of the CAN signal for ECU, ABS, and Chassis, respectively. This information is then verified through a static test.

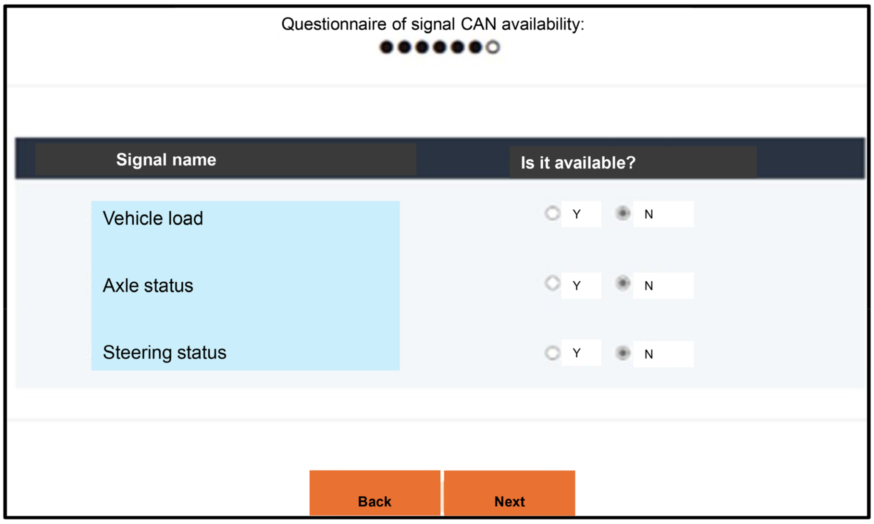


Questionnaire of signal CAN availability:

●●●●●●○

Signal name	Is it available?
Brake status	● Y ○ N
Vehicle speed - Axle 1	● Y ○ N
Vehicle speed - Axle 2	● Y ○ N
Axle status	● Y ○ N
Vehicle load	● Y ○ N
Steering status	● Y ○ N
Trailer engagement status	● Y ○ N
Gearbox status	● Y ○ N
Vehicle deceleration	● Y ○ N
Steering angle	● Y ○ N

Figure 17. CAN signal availability questionnaire for ABS ECU in ProTSA online.



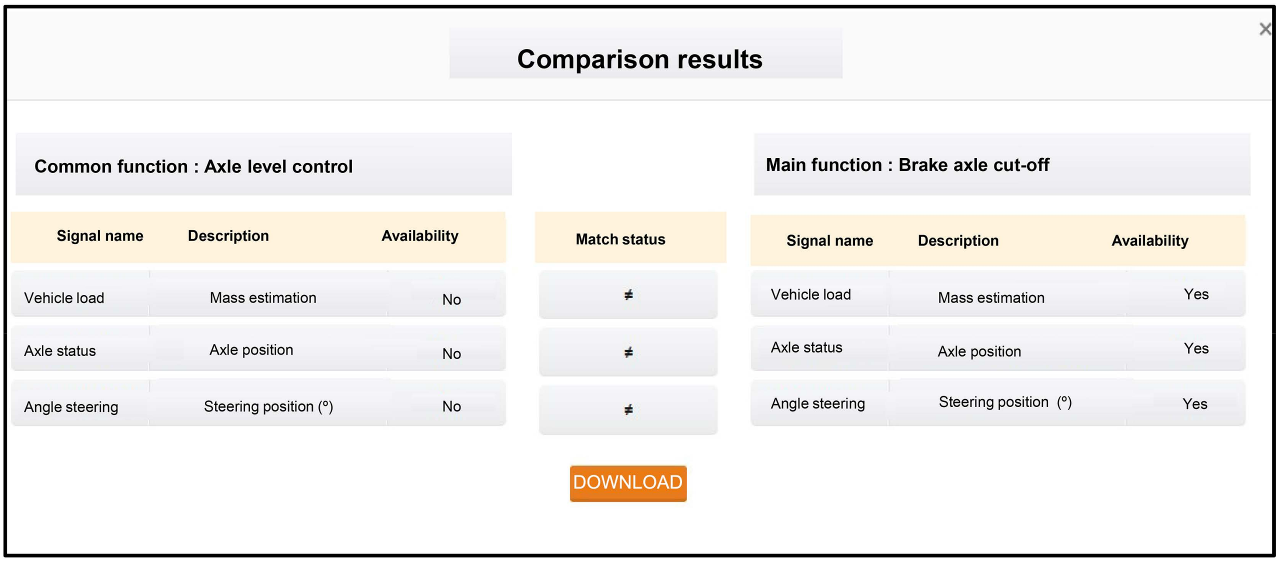
Questionnaire of signal CAN availability:

●●●●●●○

Signal name	Is it available?
Vehicle load	○ Y ● N
Axle status	○ Y ● N
Steering status	○ Y ● N

Figure 18. CAN signal availability questionnaire for ECU Chassis in ProTSA online.

In the first result of ProTSA's CAN signal integration module, users can check which CAN signals of the evaluated ECUs are not compatible for correct integration. This is done by requesting the website to check the presence of CAN signals in the two ECUs and their compatibility based on their status availability. Compatibility results are indicated with a "=" sign, while non-compatibility results are indicated with a ≠ sign, requiring discussions to achieve compatibility for correct integration. Figure 19 illustrates an example of the output from ProTSA's CAN signal integration module, demonstrating compatibility and non-compatibility results for CAN signals between evaluated ECUs.



Comparison results

Common function : Axle level control

Main function : Brake axle cut-off

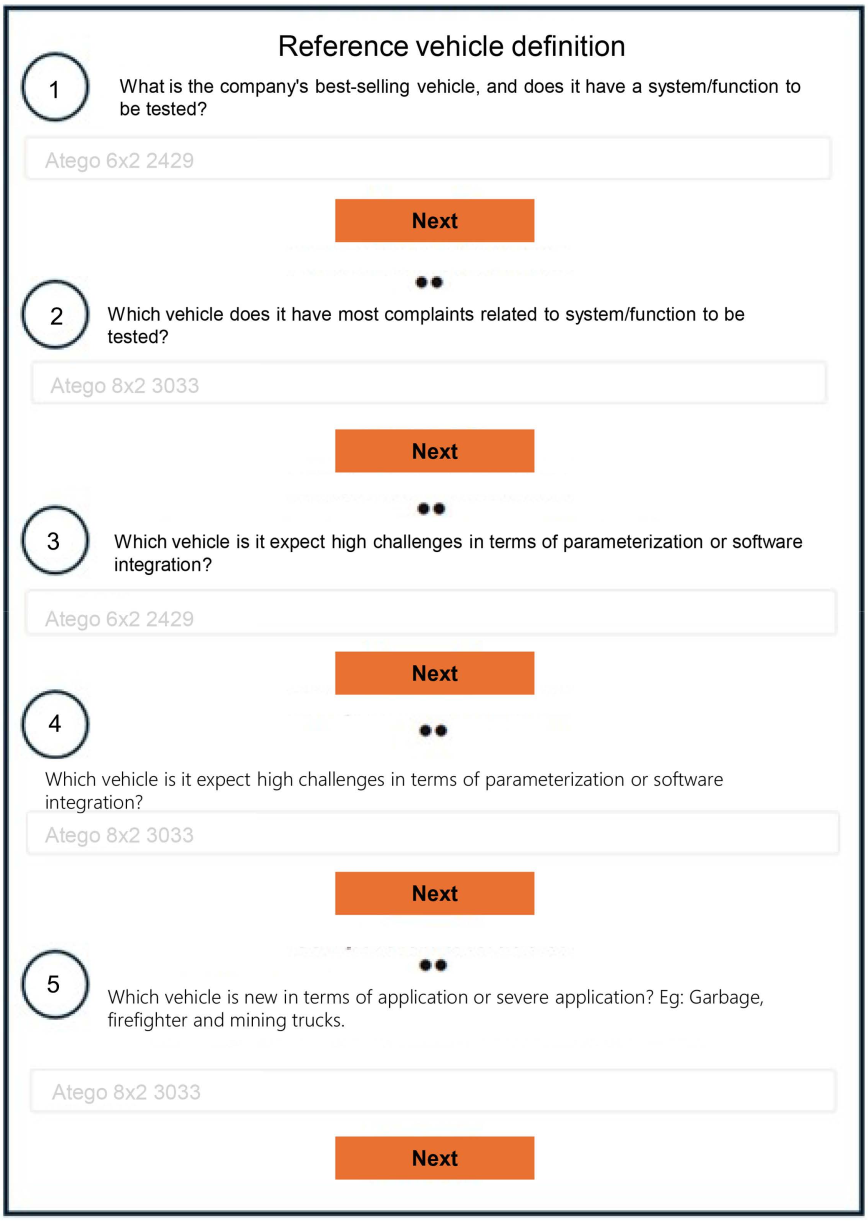
Signal name	Description	Availability	Match status	Signal name	Description	Availability
Vehicle load	Mass estimation	No	≠	Vehicle load	Mass estimation	Yes
Axle status	Axle position	No	≠	Axle status	Axle position	Yes
Angle steering	Steering position (°)	No	≠	Angle steering	Steering position (°)	Yes

Figure 19. Comparison of CAN signals in ProTSA online.

4.8. Parameter Integrity—Module 4

The next module 4—parameter integrity is structured with steps aimed at ensuring the manufacture of vehicle variants with parameter sets for each ECU and, in some cases, with a single software, but with different parameter sets, making it possible for a single type of car to be sold in distinct categories. In this way, a polo vehicle can have distinct categories and target audiences, the polo track version of the Volkswagen vehicle is equipped with an ABS and ESP safety system; while the same vehicle in the Highline category is equipped with the same ABS, ESP, and automatic braking systems [22]. In several automakers, the same type of vehicle is sold in different classes by offering different functionalities, made available by a set of parameters of a software. In this line of reasoning, a single type of vehicle can be sold in various categories and values by configuring a set of parameters in the same software. Therefore, this module addresses the importance of ensuring that the various sets of parameters are tested to avoid failures in the manufacture of the product and in the operation by the customer.

In this module, the first step that was conceptualized was the definition of priority of prototype vehicles for parameterization tests. At this step, prototype vehicle variants are selected for testing, since assessing all variants of an automaker's vehicles becomes unfeasible and, according to [8], a set of software tests tends to infinity. **Figure 20** lists a group of questions defined with software testing experts in the automotive industry and are being used in ProTSA to rank the top vehicles for functional software testing. At this stage, the portfolio of vehicles used to exemplify was from the truck manufacturer Mercedes-Benz do Brasil, available on its website [23].



Reference vehicle definition

1 What is the company's best-selling vehicle, and does it have a system/function to be tested?

Atego 6x2 2429

2 Which vehicle does it have most complaints related to system/function to be tested?

Atego 8x2 3033

3 Which vehicle is it expect high challenges in terms of parameterization or software integration?

Atego 6x2 2429

4 Which vehicle is it expect high challenges in terms of parameterization or software integration?

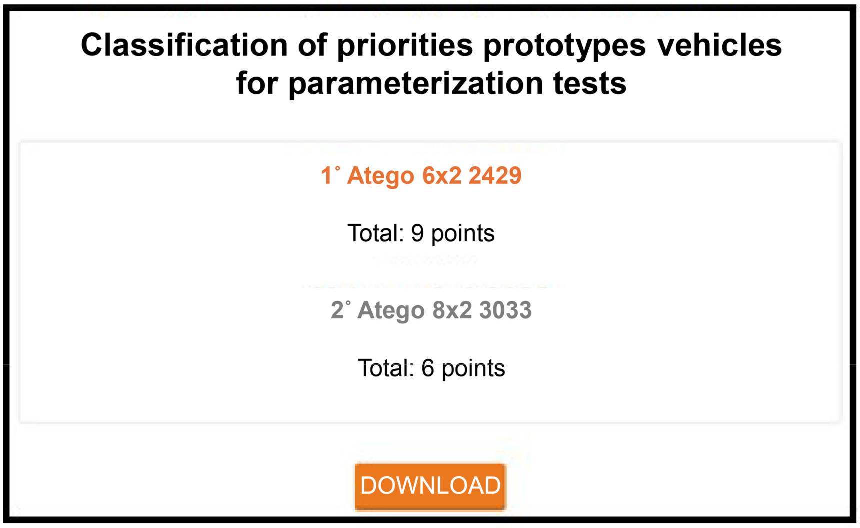
Atego 8x2 3033

5 Which vehicle is new in terms of application or severe application? Eg: Garbage, firefighter and mining trucks.

Atego 8x2 3033

Figure 20. Questions for defining and prioritizing prototype vehicles for testing in ProTSA online.

After answering the questions, the online environment lists the vehicles from the highest score to the lowest according to users for each answer. Each question has different weights defined in meetings with test specialists from a vehicle assembly company and can also be changed in the administrator environment. For this development, question 1 is equivalent to 4 points, question 3 is equivalent to 3 points, question 3 is equivalent to 5 points because it is a question of approvals, that is, legislation requirements, while question 4 is equivalent to 1 point and question 5 is equivalent to 2 points. If a single vehicle is selected for all questions, its maximum score will be 15 points. The example in Figure 21 shows the classification for the portfolio in the ProTSA online environment.


Classification of priorities prototypes vehicles for parameterization tests

1˚ Atego 6x2 2429

Total: 9 points

2˚ Atego 8x2 3033

Total: 6 points

Figure 21. Classification of vehicle priorities for parameterization tests in the online ProTSA.

In the next step, for parameterization tests, ProTSA recommends using parameterization data that worked properly in previous projects to start defining the proper parameterization set for the current project.

With this information, ProTSA recommends that users hold a workshop to align which of the contents should be released for series-produced vehicles. After the workshop, it is expected that the results of the tests will be sufficient to perform the parameterization of the software for serial production, however, if failures in the parameterization are detected, ProTSA recommends that the sequence of steps described for module 4—parameter integrity be repeated.

4.9. Maturity of ECUs—Module 5

The ECU maturity module in ProTSA is designed to verify information among stakeholders engaged in software testing activities within the automotive domain. ProTSA introduces a structured information exchange process where the requester can approve or reject the responses received. Approved responses signify an advancement in ECU maturity, which is visually represented through a progress percentage bar.

Therefore, this module is organized into six sequential steps for information exchange during testing: Software Information, Unit or Concept Tests, HILL Tests at the supplier or assembler, Application Tests, Durability Tests, and Homologation Tests. Figure 22 illustrates the structured organization of the six steps within the online environment as part of ProTSA.

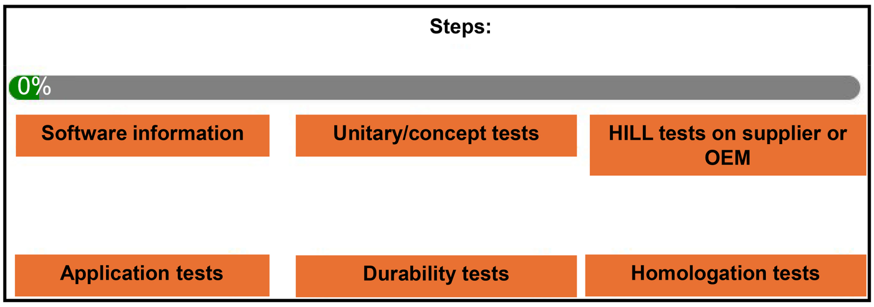


Figure 22. Steps to maturity of ECUs in ProTSA online.

After accessing the ProTSA step for ECU maturity, users will select their project and initiate the maturity process by requesting software information from the supplier.

Following this, the requester will have the choice to either approve or reject the information provided by the supplier. In the event of a rejection, the user must request clarification from the supplier regarding the reasons for rejection and any specific requirements for subsequent approval.

Following the rejection of information, the supplier is required to provide updated data for re-evaluation by the applicant. However, upon approval of the information by the applicant, it signifies that the ECUs possess the essential information to commence the testing phases. Therefore, approval at this stage, as determined by expert input, corresponds to 10% maturity out of the total 100%.

Figure 23 displays the percentage bar, incrementing by 10% following the requester's approval.

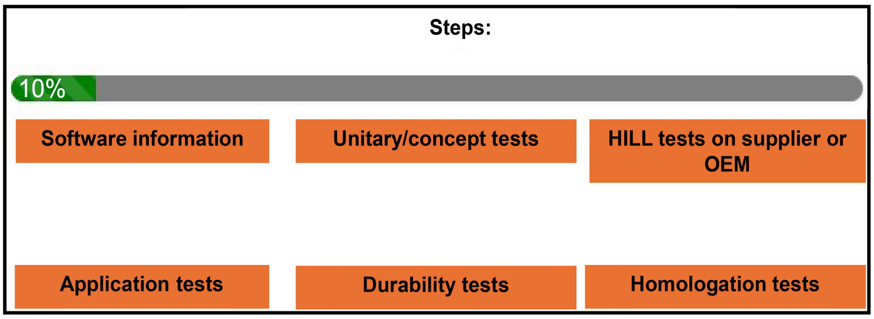
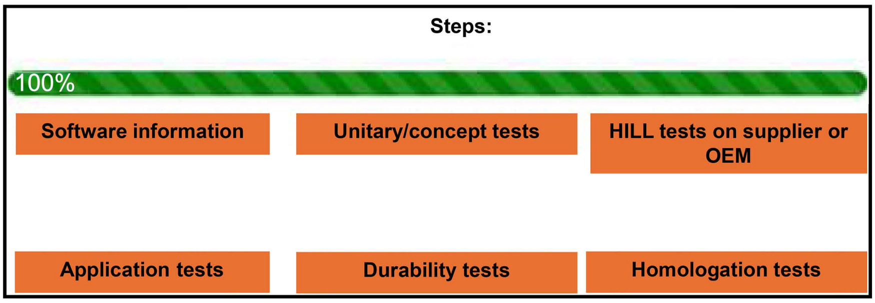


Figure 23. Software information approved.

The subsequent steps follow a similar logic of requesting information for each phase of the tests within the ProTSA module. Approval of information in any of these phases results in an increase in the percentage bar, indicating an increase in the ECU's maturity. The objective of this stage in ProTSA is to reach 100%

maturity, as illustrated in Figure 24. This signifies that all activities necessary to finalize the V & V process of the ECU/software/system have been accomplished, and stakeholders concur that it is sufficiently mature for production release. Consequently, this phase of ProTSA will also conclude.


Steps:

100%

Software information

Unitary/concept tests

HILL tests on supplier or OEM

Application tests

Durability tests

Homologation tests

Figure 24. ECU maturity completed in ProTSA online.

4.10. Fail Safe Test—Module 6

The last module—Fail Safe Test was incorporated into ProTSA at the suggestion of the experts interviewed, because during years of experience with V & V, fault monitoring is used on a regular basis, discussing each one of them, to achieve success during the development phase and V & V for any ECU/software. These experts stressed the significance of fault monitoring and discussion of each one of them throughout the development and V & V phases to ensure success for any ECU/software.

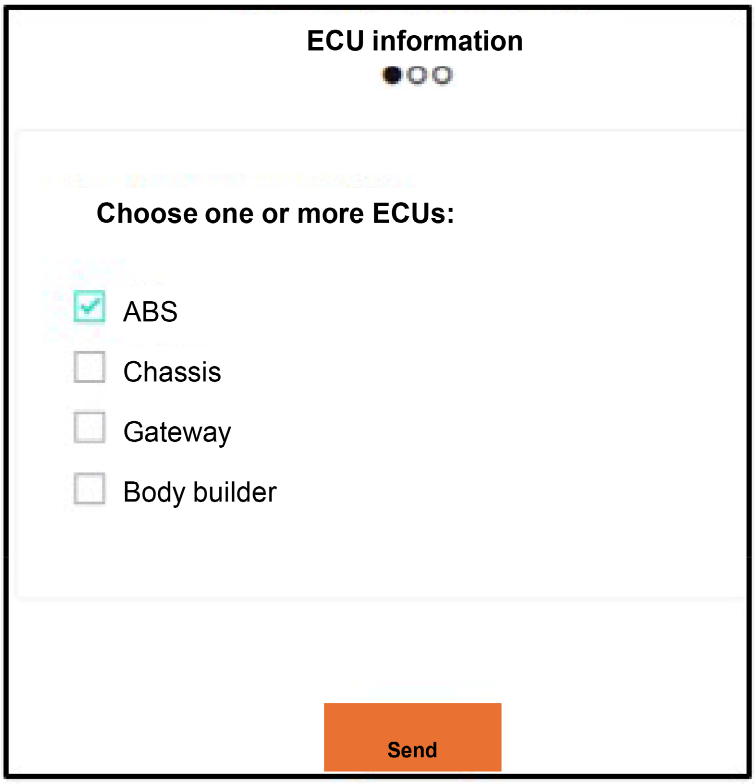
The Fail Safe Test module within ProTSA was developed with the objective of proactively identifying failures during the development phase, thereby preventing these failures from propagating to serial software or vehicles for end customers. It is structured to map faults by ECU or vehicle configuration, generate fault code reports, display solution progress graphs, and organize regular workshops to prioritize resources based on the number of failures per ECU or vehicle system.

The initial step of the Fail Safe Test module within ProTSA requires users to access the Fail Safe Test menu, choose the ECU for assessment (Figure 25), and upload the list of fault codes gathered from prototype vehicles (Figure 26).

Subsequently, the fault codes of the previously selected ABS ECU will be displayed on the pre-selected vehicles, along with the option to download this information for future alignments and stakeholder meetings (Figure 27).

The next step in the ProTSA Fail Safe Test module enables users to generate graphs illustrating the failure status of ECUs, categorizing faults as not relevant, under repair, or resolved. The status attributed to each fault must align with the individual responsible for this aspect in ProTSA. To mark a fault as resolved, evidence such as measurements with the proposed solution, specification documents, or user feedback is necessary.

The graph in Figure 28 represents the fault status of the ABS ECU in selected prototype vehicles over a specific period.



ECU information

Choose one or more ECUs:

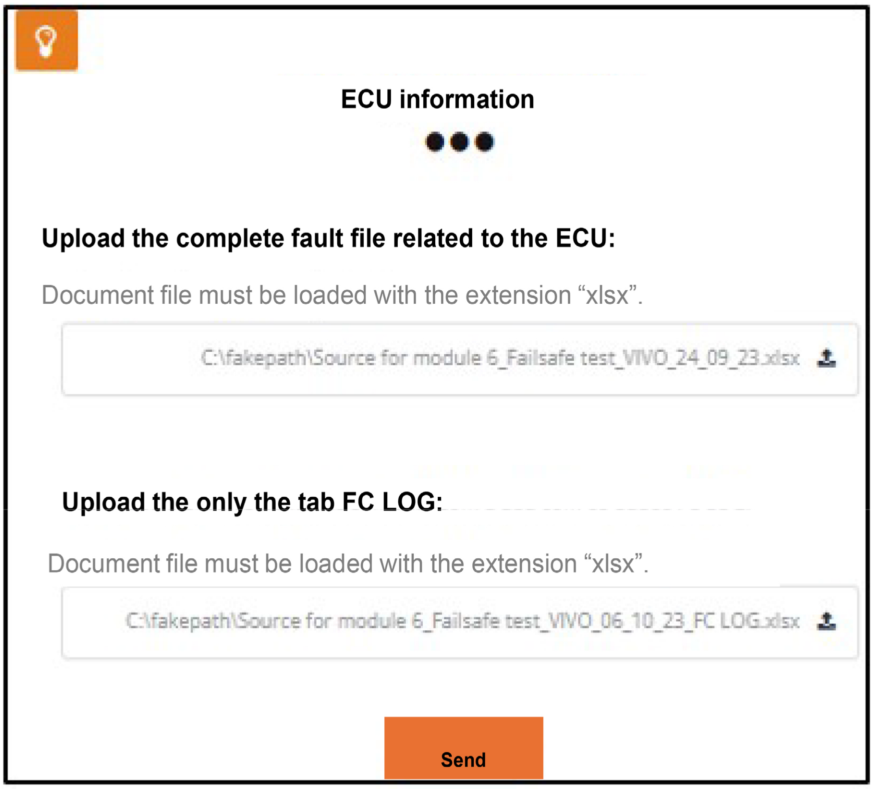
☑ ABS

☐ Chassis

☐ Gateway

☐ Body builder

Figure 25. ECU selection in ProTSA online.



ECU information

Upload the complete fault file related to the ECU:

Document file must be loaded with the extension "xlsx".

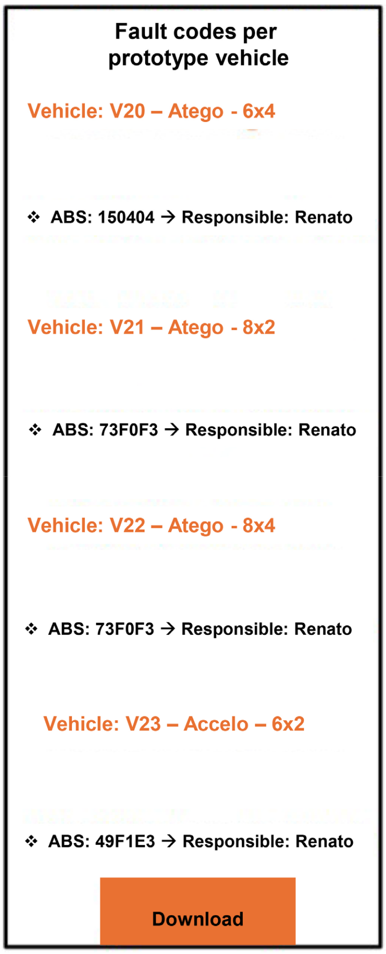
C:\fakepath\Source for module 6_Failsafe test_VIVO_24_09_23.xlsx

Upload the only the tab FC LOG:

Document file must be loaded with the extension "xlsx".

C:\fakepath\Source for module 6_Failsafe test_VIVO_06_10_23_FC LOG.xlsx

Figure 26. ECU selection in ProTSA online. Prototype Vehicle Failure spreadsheet in ProTSA online.



Fault codes per prototype vehicle

Vehicle: V20 – Atego - 6x4

- ABS: 150404 → Responsible: Renato

Vehicle: V21 – Atego - 8x2

- ABS: 73F0F3 → Responsible: Renato

Vehicle: V22 – Atego - 8x4

- ABS: 73F0F3 → Responsible: Renato

Vehicle: V23 – Accelo – 6x2

- ABS: 49F1E3 → Responsible: Renato

Figure 27. Reporting failures of one ECU per vehicle in ProTSA online.

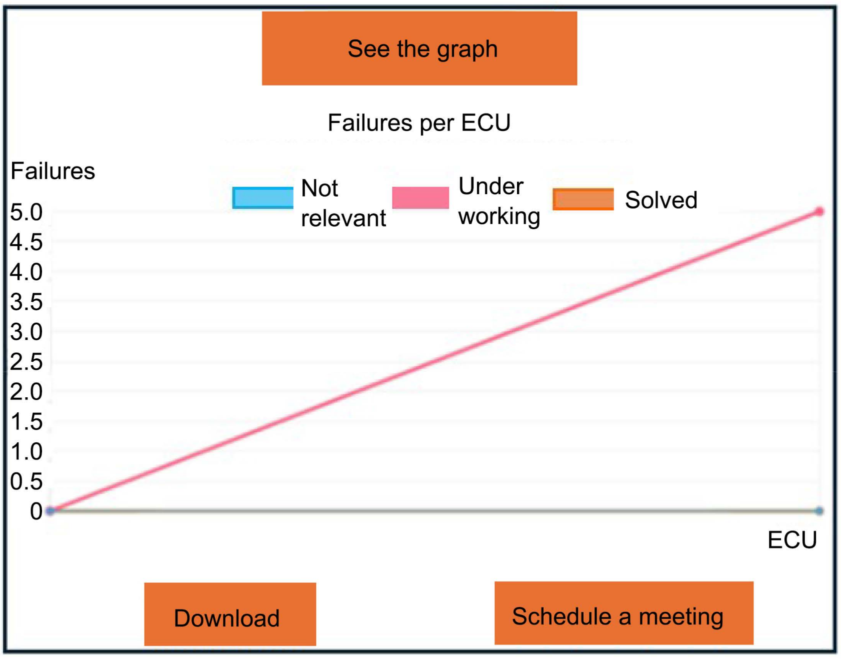


Figure 28. Solution status for ECU in ProTSA online.

In the final stage of the Fail Safe Test module, it is advised to conduct weekly workshops to monitor the progress of fault resolution by ECUs/software and determine resource priorities. Users can schedule these meetings on the ProTSA platform by utilizing the "schedule a meeting" button and sending invitations along with generated reports to participants, allowing them to prepare in advance for the workshop and take necessary actions.

5. Results & Analysis

This section outlines the professional profile of fourteen voluntary professionals from a commercial vehicle manufacturing company who participated in a survey on the software testing process in the automotive domain. At selection process of this OEM involves availability, easy contacts and wiliness of software testers to participate as contributor to this study and also intended to have an improvement in their software V & V activities, therefore the authors defined this specific company as a relevant sample to perform this study. The survey was conducted between November and December 2023. Out of a total of eighteen professionals invited, fourteen accepted and actively participated in the survey. These participants also completed a test case as a prerequisite for answering the survey questions. The sample of volunteers for this research comprises professionals experienced in V & V of automotive vehicle safety systems, powertrain, and mechatronics integration software. They were informed about the research's confidentiality and accepted the Informed Consent Form during the survey.

5.1. Participants

According to Figure 29, the survey participants' experience level in software development or testing is distributed as follows: 35.7% have 0 - 3 years of experience, 28.6% have 3 - 5 years of experience, 14.3% have 5 - 10 years of experience, and 21.4% have more than 10 years of experience in the field.

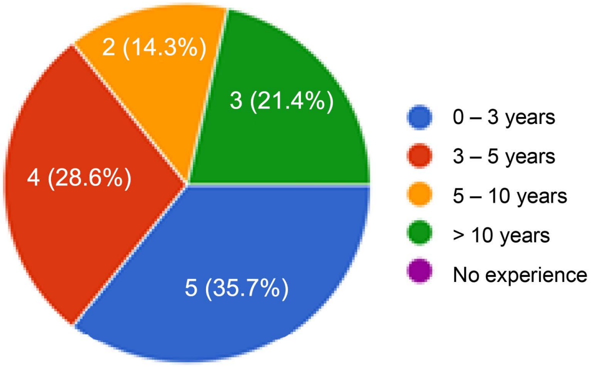


Figure 29. Years of experience in software development or testing.

The respondents' roles are distributed as follows: 64.3% are engineers, 14.3% are technicians, another 14.3% are analysts, and 7.1% are interns, as illustrated in Figure 30.

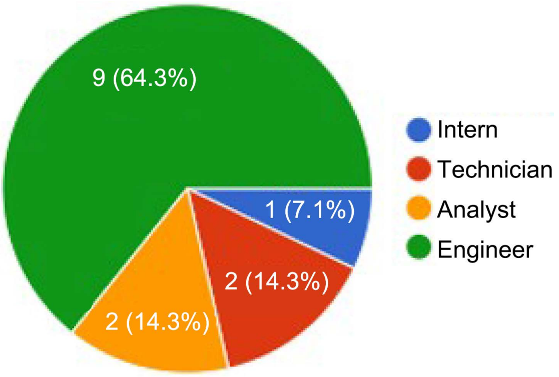


Figure 30. Role of survey participants.

Regarding which segments of automotive software testing the participants have already worked, considering that they may have worked in more than one of the segments listed, forty-six votes were obtained. The segment in which the participants most worked was V & V with 24%, in the next place, the application and integration tests stand out with 21.7% each. In the next position, prototype tests with 17.4%, while documentation tests appear with an 8.7% performance and homologation with 6.5% (**Figure 31**).

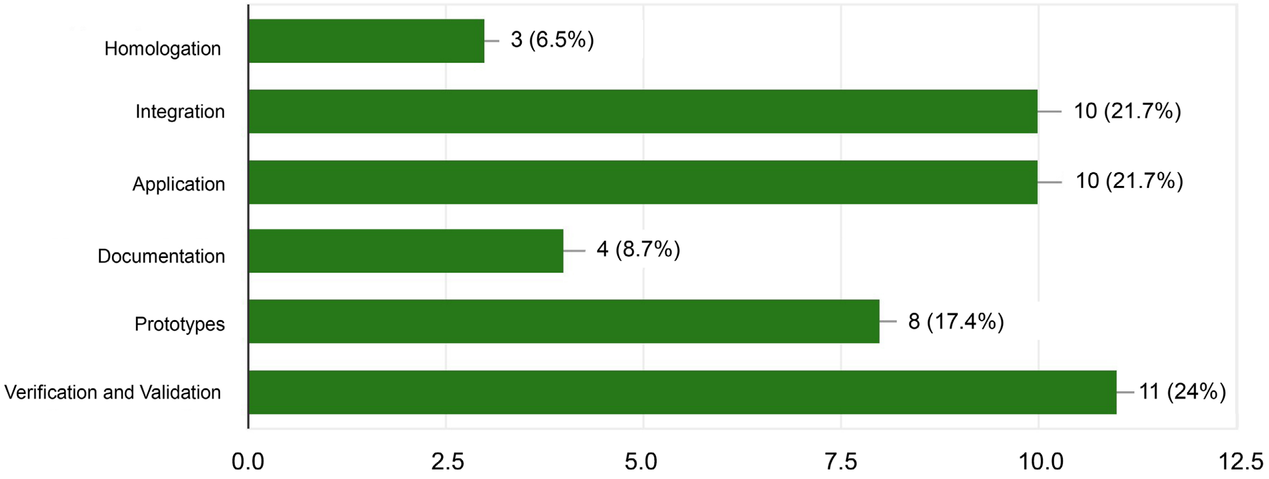


Figure 31. Area of expertise of the survey participants.

The survey also examined the number of participants who have previously utilized a standardized process for software validation. **Figure 32** reveals that 2 participants (14.3%) have employed a standardized process before, whereas 12 participants (85.7%) have never utilized a standardized process for software validation.

Related the two participants who responded about their experience with standardized processes mentioned using the Fail-Safe Test, which is also integrated into ProTSA. The second process involved a test script for new functions, in alignment with the parent company, focusing on software validation for Diesel engine ECUs.

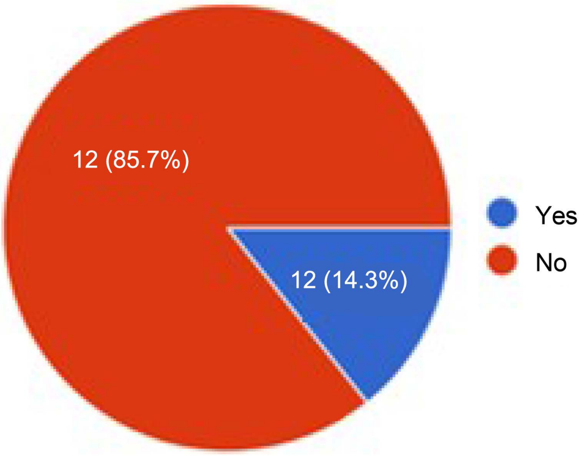


Figure 32. Participants who have already used a software validation process.

The participants unanimously expressed a positive predisposition towards using a standardized V & V (Verification and Validation) process in their daily work for tackling testing stages. This inclination was reflected in all fourteen respondents, as illustrated in **Figure 33**.

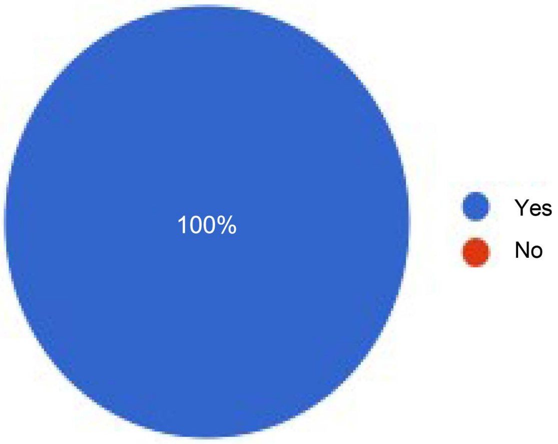
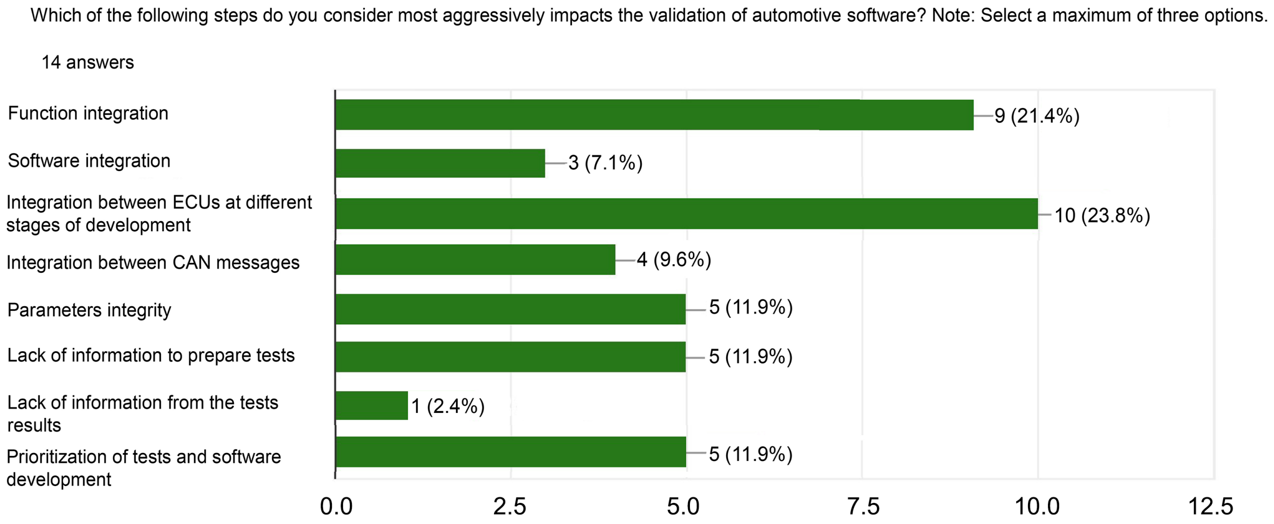


Figure 33. Participants predisposed to use a standard process for V & V.

In the next question, the participants were asked about critical activities or actions that could influence the outcome of a V & V process based on their experience. This inquiry sought to identify key aspects raised by participants that ProTSA could subsequently assist in streamlining their daily work.

Among the alternatives listed, the one that obtained the most votes was the integration activity between ECUs at distinct stages of development, with 10 votes, which represents 23.8% of the total of 42 presented. The second most favored option, function integration activity, was selected by 21.4% of respondents. In the third position, there was a tie between three options: parameters integrity activity, lack of information for test preparation, and prioritization of tests and software development, each receiving 11.9% of the votes.

In the fourth position, the integration activity between CAN messages received 9.6% of the votes, followed by software integration with 7.1%. Additionally, one participant reported a lack of information from test results as impacting the V & V process, representing 2.4% of the responses, illustrating in **Figure 34**.


Which of the following steps do you consider most aggressively impacts the validation of automotive software? Note: Select a maximum of three options.

14 answers

Figure 34. Experts opinion regarding most relevants activites that impact the V & V in the automotive domain.

5.2. ProTSA Assessment Methodology

A test case was presented to the experts in ProTSA, involving a V & V process between the ABS and Chassis ECUs. This case aimed to integrate two functions: the axle brake cut-off function in the ABS ECU and the axle lifting control in the Chassis ECU. The validation and integration were planned for a 6 × 2 commercial vehicle model currently in production. The participants were given a tutorial outlining the steps to execute this process.

Methodology Evaluation of Questions with Likert Scale

When evaluating ProTSA through a survey, questions were rated using the Likert scale with a 5-point range. A rating of 1 indicated "Strongly disagree" with the statement, while a rating of 5 indicated "Strongly agree." To quantitatively analyse the collected data, mean and standard deviation calculations were conducted.

The Likert scale question aimed to gauge experts' opinions on whether ProTSA could enhance resource optimization and boost productivity during V & V phases. The survey yielded a mean score of 4.64 with a standard deviation of 0.47 (**Figure 35**). The average score above 4, coupled with responses mostly falling between 4 and 5, indicates a positive impact attributed to ProTSA in terms of resource optimization and productivity gains. This aligns with the intended objectives of ProTSA, which emphasizes structured testing processes to enhance productivity, efficiency, resource optimization, and overall quality, as noted by Vieira [24].

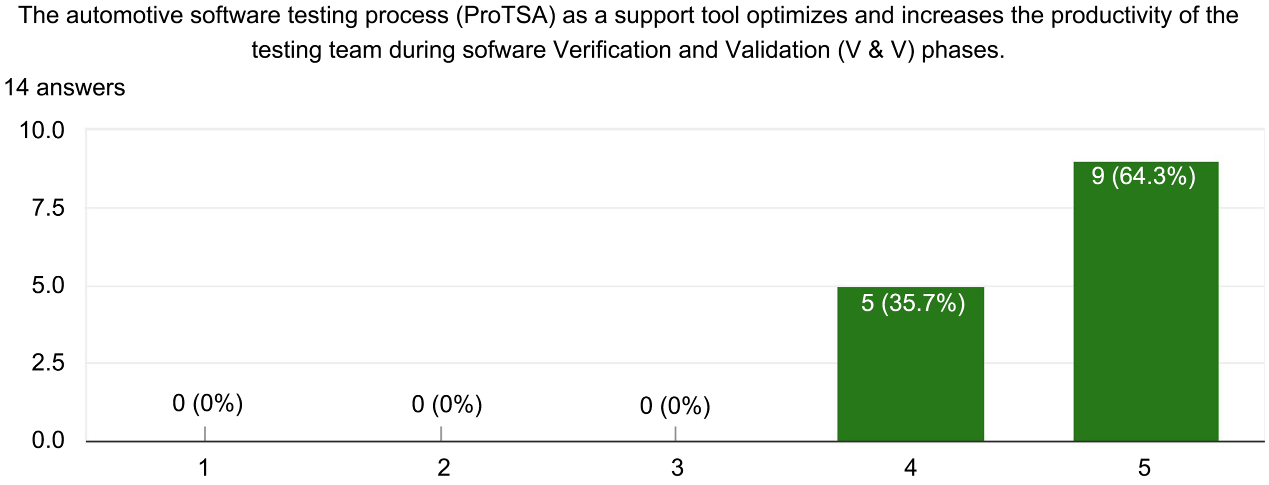


Figure 35. Expert assessment for impact of increasing productivy due the application of ProTSA

The second question sought participants' opinions on whether introducing ProTSA with defined steps would positively impact the quality requirements in V & V processes for automotive software. The survey results, with a mean of 4.78 and a standard deviation of 0.41, indicate a generally positive outlook. Most responses falling between 4 and 5 suggest that most evaluators believe ProTSA's introduction would enhance the quality of V & V processes (**Figure 36**). This aligns with the notion in Rocha [25] that software quality is enhanced through proactive measures like early detection and prevention of faults, which ProTSA's Fail-Safe Test module is designed to facilitate.

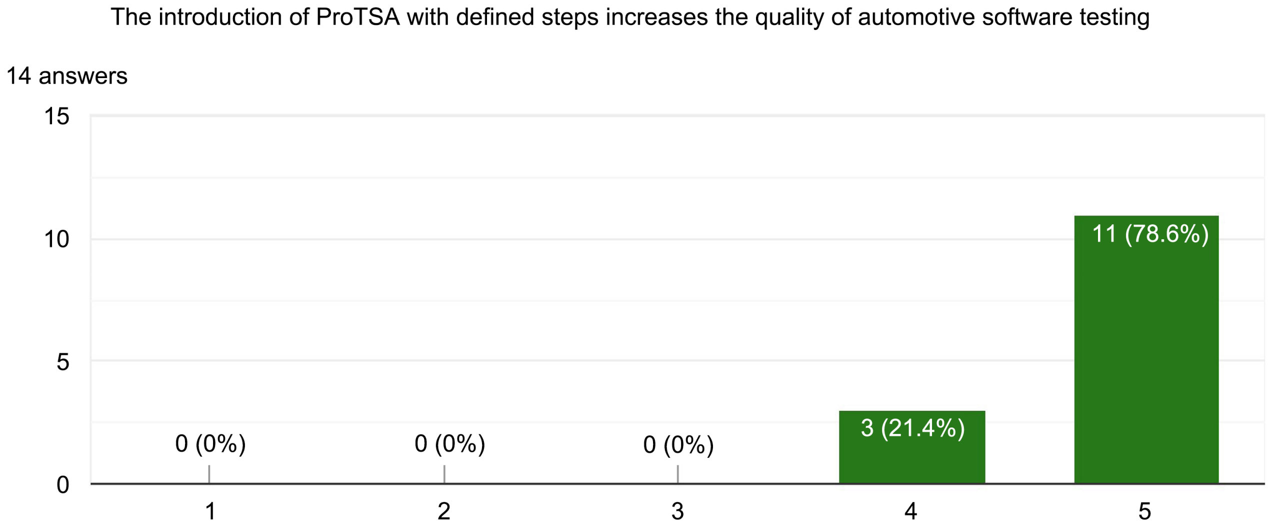


Figure 36. Expert assessment related increasement of quality software testing processes in the automotive environment by introducing ProTSA.

In the third question, the objective was to receive feedback from the experts regarding the increase in efficacy if ProTSA is introduced in V & V processes. The mean of 4.42 and standard deviation of 0.49 suggest, in a positive way, gains in terms of efficacy. However, in **Figure 37**, it is possible to detect more responses at value 4, which suggests the need for some points of improvement in ProTSA to help with the efficacy requirement. The importance of efficacy is relevant and was considered in ProTSA, because, according to Galdino [26], there is a growing demand for quality certificates by customers and how a V & V program can tend to infinite test cases. Therefore, the selected test cases need to be effective to find more defects in less time and avoid high costs for software error corrections. Thus, this requirement was included for the evaluation of the experts.

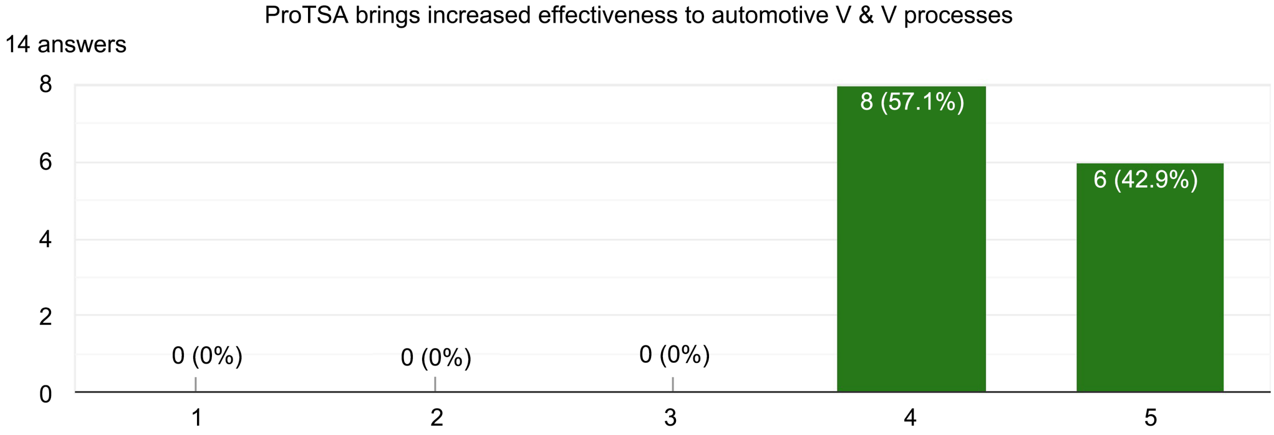


Figure 37. Evaluation of the positive impact of the introduction of ProTSA on the efficacy requirement of tests.

The subsequent question sought feedback from experts concerning the clarity and ease of using ProTSA. Overall, there was positive feedback with a mean score of 4.28 and a standard deviation of 0.58. However, the slight variance indicated by the standard deviation and some respondents choosing value 1 suggest differing opinions among experts regarding this aspect, as depicted in **Figure 38**. To enhance this requirement, some participants proposed conducting training sessions on ProTSA to enhance clarity and adaptability with the process.

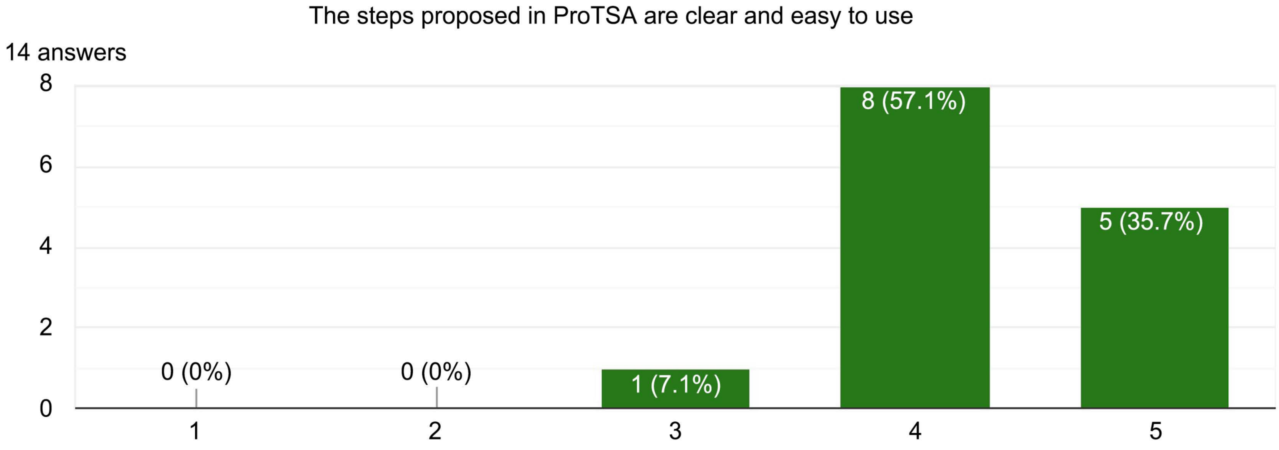


Figure 38. Expert Feedback on ProTSA's Clarity and ease of use.

The subsequent evaluation aims to assess the relevance of the steps incorporated in ProTSA. The mean score of 4.78 and standard deviation of 0.41 suggest a high level of relevance of the steps outlined in ProTSA for a V & V process in the automotive environment. The predominance of responses at value 5 further supports the significance of these steps, as depicted in **Figure 39**.

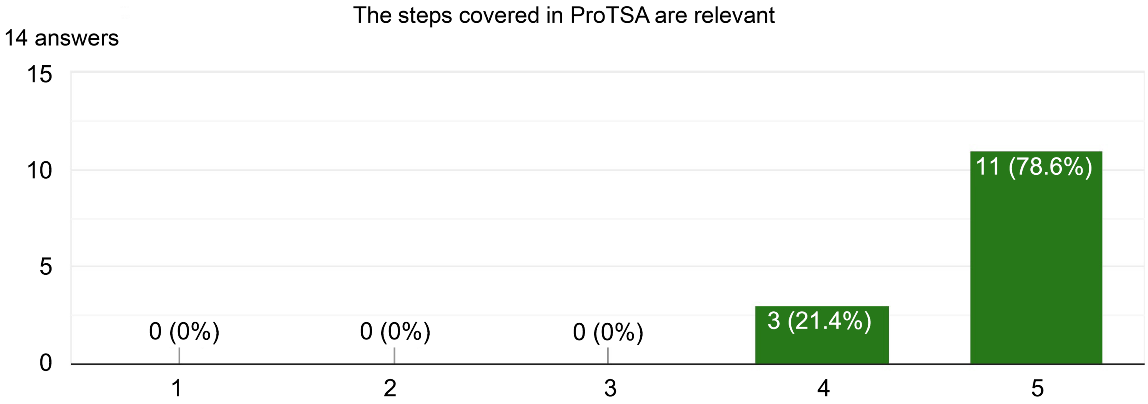


Figure 39. Expert assessment of the relevance of the steps covered in the ProTSA.

In the next question, the study sought to verify the ease of applicability of the proposed process in the automotive environment. With a mean of 4.21 and a standard deviation of 0.67, the answers with a mean higher than 4 positively suggest the ease of applicability of the process. However, the relatively high upper standard deviation indicates that there are some participants who have different opinions regarding the mean and the easy applicability factor. According to Marques [27], there are some success factors in the implementation of a project such as adequate budget, support from top management, provision for training, project sponsor, etc. These points raised by the study are factors that some experts may have seen as difficult factors for applicability quickly and easily. Therefore, there were 2 evaluations on scale 3, as shown in **Figure 40**. Suggestions for improvements pointed out by the participants will be presented in the subsequent sections.

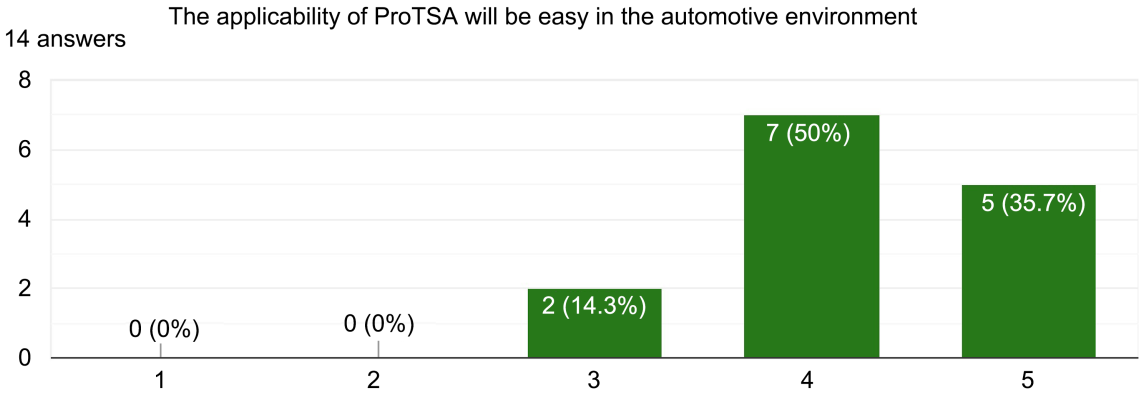


Figure 40. Expert assessment of ProTSA's ease of applicability in the automotive environment requirement.

The penultimate question asked the experts to evaluate the presence of inconsistencies in the proposed process. A score of 1 indicated few inconsistencies, while 5 indicated many inconsistencies. The average score of 1.42, close to 1, suggests that participants generally perceived few inconsistencies in the process. However, the standard deviation of 0.62 and some responses at value 1 indicate slight variation in the experts' opinions, as illustrated in **Figure 41**.

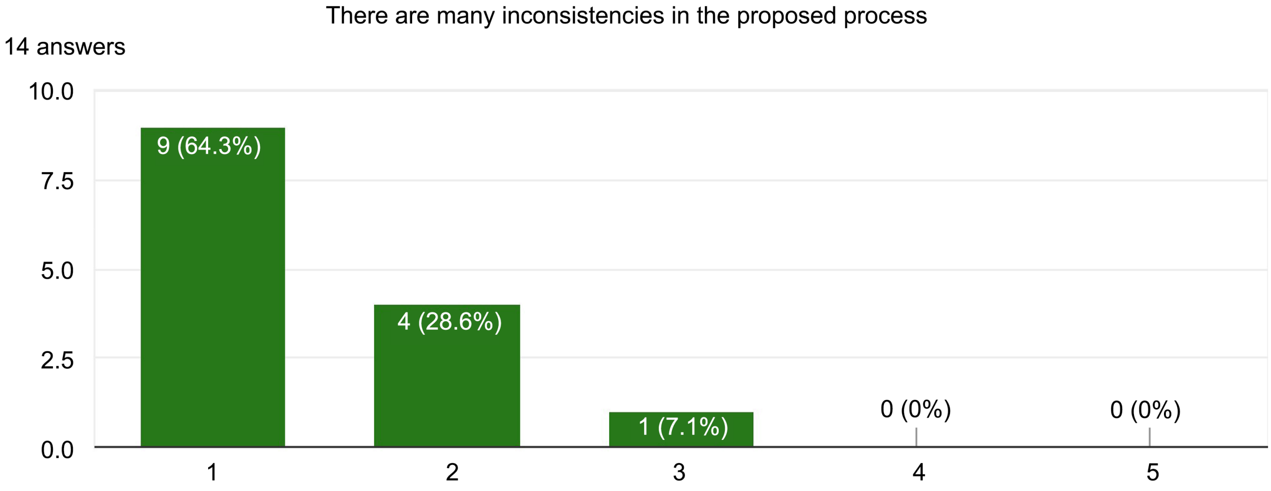


Figure 41. Expert assessment of ProTSA's for inconsistencies

In the final question using a Likert scale, participants were asked to rate the completeness of the ProTSA steps. The responses yielded an average score of 4.28 and a standard deviation of 0.69. This indicates that most opinions fell within the range of 4 or higher, reflecting a positive trend regarding this evaluation criterion. However, two responses rated the completeness at 3, as depicted in **Figure 42**, signaling room for improvement and suggesting areas that participants highlighted for enhancement, which will be detailed in subsequent sections.

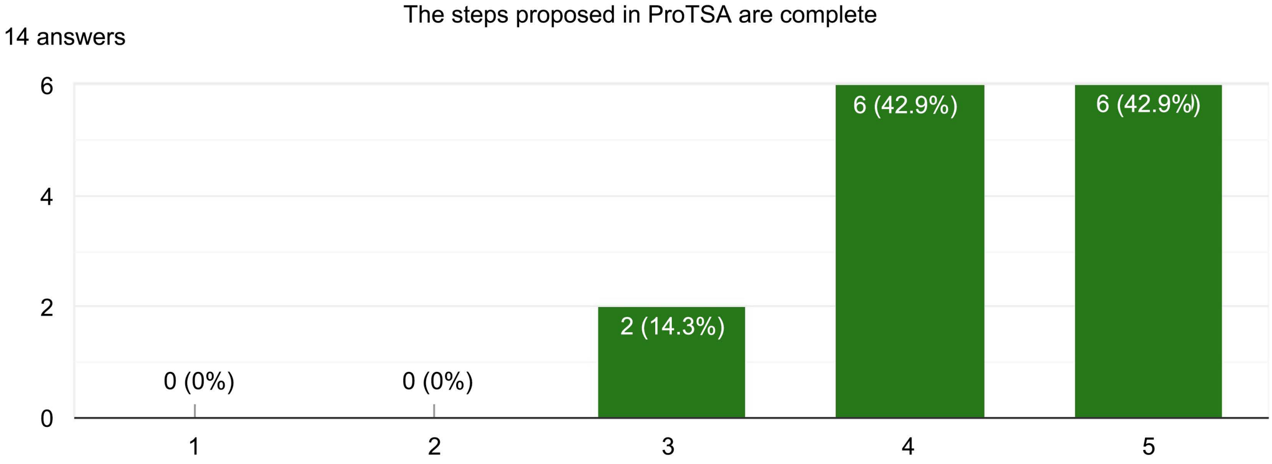


Figure 42. Expert assessment of whether the ProTSA steps are complete.

5.3. Suggestions Made by Survey Participants

The final question of the survey asked participants to suggest areas for improvement in ProTSA. In this aspect, and in a more qualitative way, the following points were listed:

1) Adds an exclusive step for SiL

One of the participants suggested including the SiL stage in ProTSA in an exclusive module, as it is a more agile type of test and is already used as part of the approval process for security systems for the specific OEM, such as the Electronic Stability Program (ESP).

2) Training for the use of ProTSA

A suggestion was made for a hands-on training format with an instructor in person to clarify doubts that can occur during the use of the process.

3) Incorporation of ProTSA into the company's development and testing strategy

One response received was the support from middle and senior management for the effective integration of the ProTSA process into R & D teams was suggested as an important factor for successful implementation.

As it is a new tool, there is an initial phase of some adaptation, rejection, or challenges to operate without errors and management support can be a large part of the success of the implementation.

4) Automation of the online interface

Some professionals have suggested that the steps in the current version of ProTSA could be more automated or integrate them into the company's existing processes.

5) Maturity of the prototype vehicle

Some professionals mentioned the need to incorporate a stage and verification of the maturity of the vehicle, as many functional software tests have a direct impact on the maturity of the vehicle.

6) Approach not only to testing, but to hardware and software obsolescence risks.

One survey participant reported the need to broaden the scope of ProTSA and add hardware and software obsolescence risk mapping. One proposal would be to have an exclusive module with roadmaps for each ECU and results of discussions of obsolescence topics to be available to the entire R & D team on a single platform.

7) Application in a larger number of real cases

One user reported the need to apply the process to a larger number of test cases to identify possible additional points for improvement.

8) Positive feedback

Some users have reported that the process is specified, with clear steps, steps well sequenced from each other and robust. In addition, it was also mentioned that, once the process is explained, it is easy to understand the application of the sequence of use for ProTSA.

6. Discussion on Research Questions

RQ1 was defined as "What software testing techniques and strategies in the automotive environment improve software quality?". From the development of this work, the author understands that it was possible to identify types of tests, alignment strategies to define and prioritize the execution of test cases based on evidence of defects, in addition to active leadership is something that supports the increase in the quality of the software. Therefore, according to the research carried out, the increase in the quality of the software is allied to a set of actions or techniques employed through a V & V process.

As a result of identifying this set of activities or steps needed to improve software quality, it was possible to propose a V & V process for application in the automotive domain. The proposed ProTSA process was structured into six modules, the first module—integration between functions has as one of the objectives to make an analysis of the documentation in a group to identify defects in advance. In this module, there is also a proposal to create block diagrams to facilitate the visualization of the integration between the functions of the software and to answer a questionnaire to identify which of the functions needs to be developed with priority and allocate more resources, through the classification of client and service functions.

The second module—software integration, focuses mainly on suggesting the creation of block diagrams to visualize the integration between software and hardware involved. In addition, a mapping of release dates for features for testing is carried out and, finally, it is sought to create regular meetings to monitor the development of this stage of ProTSA.

Regarding the next module—integration of CAN signals, one of the objectives is to map the CAN signals necessary for the correct functioning of the ECUs/software/features involved in the integration. Thus, the expected output of this module is the identification of CAN signal compatibility in advance, to avoid problems only in the stages of dynamic software testing where the cost to correct errors will be higher. In this module, it is also suggested a strong alignment between the team through reports to achieve success in the progress of the tests.

In the fourth module—parameter integrity, the focus is to ensure a parameterization of the software capable of applying to various vehicle variants without any customer fault or error codes. To do this, it is necessary for the team to answer a questionnaire for the purpose of classifying the most critical prototype vehicles for testing the parameters. In the next step, parameter information is cross-referenced with previous projects. In addition, a new spreadsheet is prepared so that the parameters can be evaluated and documented to be released to production.

In the penultimate module—maturity of ECUs, the main objective is to establish a flow of information exchange of test results between testers, developers, automakers, and suppliers. In this way, it will be possible to request and obtain information for the creation of test cases, test results and, if the results are sufficient and by mutual agreement, those involved will be able to classify that test stage as overcome and its maturity will be high. Therefore, through this module, those involved in the V & V process will have clarity and transparency and continuous alignment on the ongoing testing stages and confidence to evolve the maturity of a test.

In the last module, the monitoring of fault codes in a regular and structured way for all ECUs with embedded software is focused. Through the identification of the codes, an action plan is developed to solve the error and a follow-up is focused on improving the quality of the software. In this module, it is also proposed a robust performance of the team leader, to make decisions to allocate resources in software error resolution based on the evidence of the number of mapped fault codes.

Therefore, the increase in the quality of software in the automotive environment can be increased through a set of techniques as proposed in ProTSA and ratified by the survey participants. Another relevant factor refers to an effective communication plan, that is, a continuous alignment between stakeholders in the V & V process. Additionally, the presence of a technical project leader acting strongly in the allocation, prioritization of resources and orchestrating the teams will be a success factor for increasing the quality of the software.

In relation to RQ2, this was defined as "How does the proposed testing process improve the productivity of automotive software testing teams?". From the introduction of a V & V process, in this case, ProTSA, which has structured and sequenced steps with the objective of detecting defects through a specification analysis instead of detecting the same defect only in a dynamic test in a prototype vehicle and, later, which directly impacts the cost of correcting the defect. Productivity can be calculated by means of the following formula: productivity = product produced/total cost, for this question, a product produced with the number of defects detected in each phase of the test will be used for example, and the total cost will be defined according to the phase of the software project. For example, during a documental test (specification analysis), 10 defects were found at a total cost of 10 units of money. If the software were assessed only on a prototype vehicle, the same 10 would also be found, however, at the cost of 1000 units of money. This difference in cost is due to Myers' rule of 10, in which the cost of correcting a defect increases by 10× for each later phase in that the defect is detected, that is, if detected in the initial phase (concept), the cost of correction would be 1 unit of money; If detected at the later stage, its cost would be multiplied by 10, which would be a total of 10 units of money, and so on, as shown in Figure 43.

Therefore, when applying ProTSA, it is recommended to run tests in the initial stages of software design, such as in the specification phase. Considering the use of ProTSA and the beginning of the tests in the specification phase, when calculating the productivity of 10 defects identified in this phase, the productivity

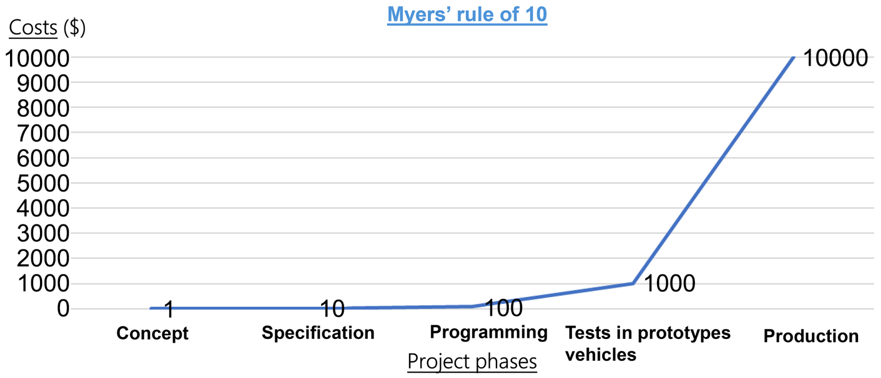


Figure 43. Myers' rule of 10. Adapted from [20].

value is equal to 10/10 = 1. While, when applying the current testing methodology of the company under study, the software will be assessed only in the testing phase in prototype vehicles, and possibly the same number of failures will be detected (10 failures), but late. This will result in a higher cost of correcting the software according to Meyers' rule of 10 and when calculating the productivity, a lower value will be obtained, *i.e.*, equal to 10/1000 = 0.01. Therefore, because ProTSA proposes tests from the beginning of the software project instead of waiting to perform tests on prototype vehicles, it generates a direct impact on the cost of correcting defects and consequent on productivity. This concept also applies to another ProTSA module, called Fail Safe Test, in which fault codes will be mapped and corrected before the software/vehicle goes into production, which reduces the costs of correcting defects and, consequently, increases productivity.

Finally, the survey also addressed whether there would be a positive impact in terms of productivity with the introduction of the ProTSA process. The result was positive with a mean of 4.64 and a standard deviation of 0.47. Thus, the participants demonstrate that they understand that there will be an increase in the productivity of the test teams, which is in line with the demonstration of the relationship between productivity and the anticipation of test tests and Meyer's rule of 10.

In the unfolding of RQ2, RQ2.1 is defined as "What is the saving time?". This question was asked to determine what the time savings would be when we introduced ProTSA in the commercial vehicle company, which is the reference for this study. Through meetings with survey participants, some of them mentioned that they did not see the benefit of static/documentation tests and had never had experience with this type of test with the automaker and emphasized that the main tests performed are the functional ones directly on the prototype vehicle.

Based on the participants' reports and study of the company's current V & V software model, it was also identified, that the average development cycle of a new ECU is about 2 years for the specific company. Regarding testing, normally, the automaker only starts in the testing phase in prototype vehicles, but with ProTSA, if incorporated into the company's strategy, the idea is to start static tests/documentation in the concept and specification stages. For this specific company, if the tests start in the concept phase with the use of ProTSA through block diagrams and CAN matrix compatibility, instead of starting only the test stage in vehicles, an anticipation in the detection of defects would be obtained and named saving 1 in **Figure 44**, totaling a saving time of 6 months. If the tests were started in the specification stage with group discussion, to understand the content of the specification and simulate the operation of the software in a non-dynamic way instead of starting the tests only in the testing phase in prototype vehicles, a saving of 2 would be obtained, totaling 4.5 months of saving time. These time savings through the application of ProTSA are for the specific company case and may vary according to the company to which it applies, the software project development cycle.

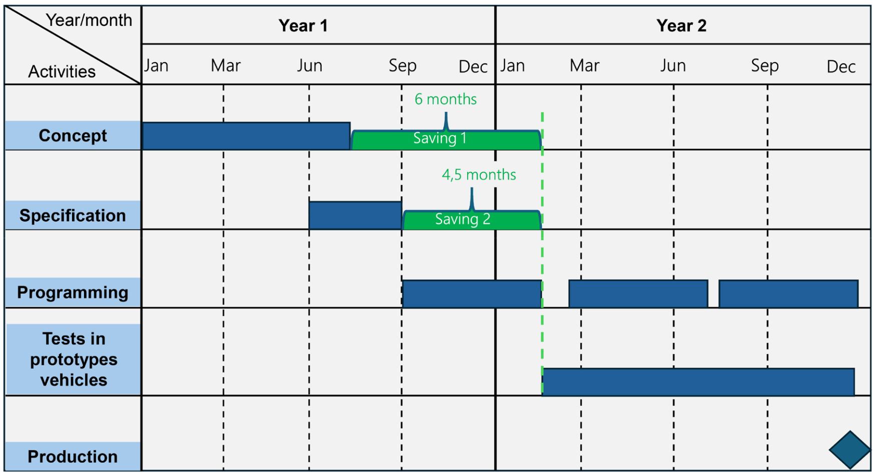


Figure 44. Myers' rule of 10. Adapted from [28]. Expert assessment for saving time in V&V application of ProTSA.

For RQ3, it was defined as "does the proposed testing process improve the effectiveness in defect detection?".

The implementation of ProTSA aims to make the results of data analysis transparent among those involved by sending reports, holding alignment meetings, and follow-up. With an effective communication strategy and the testing process being visible to all involved, as proposed in articles covered in the SLR, it is expected that the test cases will be better chosen and obtain greater effectiveness in detecting defects. This alignment strategy works on the peer review strategy, in which the test cases developed are reviewed to define which ones have a greater tendency to detect defects quickly.

In addition, ProTSA modules such as function and software integration ask participants to prepare block diagrams to explain to the participants the integrated functioning of the software and functions. Therefore, once the explanation becomes more visual, it will be possible to understand the integrated system more easily and, consequently, define more effective test cases, reducing the group of tests required due to the understanding of the system through the block diagram.

Finally, in the survey, participants were asked if the efficacy requirement is improved with the introduction of ProTSA. The mean results for those agreeing with increased efficacy was 4.42 and standard deviation was 0.42. The results for those who agreed with increased efficacy were 8 participants, while for those who strongly agreed it was 6. Therefore, according to survey participants, it is possible to increase the effectiveness of defect detection, but there is still room for improvement in the ProTSA process, as suggested in Section 5.3.

In addition, the author realized, for future work in Software V & V, in the automotive domain, asking the following questions may be relevant:

- What benefits do you think the proposed solution brings to V & V software?
- Which software V & V technique increases testing effectiveness?
- Do you think the proposed process increases the quality of software V & V results? If so, what would be the percentage?
- Do you think your productivity percentage will be increased? If so, what would be the percentage?

7. Conclusions

RQ1 was defined as "What software testing. The work of this research enabled the development of ProTSA, which contains a sequence of six test modules with defined inputs and outputs.

In the work, a set of knowledge from several areas was applied, such as research methodology, software development models, types of V & V for software, and the perspective of professionals from the automotive industry was integrated through meetings to collect information for the development of the ProTSA and participation in the survey, which the employees agreed to participate voluntarily.

It was found that the use of the Design Science Research methodology worked properly in the development of ProTSA to solve a real difficulty of a commercial vehicle manufacturing company. The development sequences to solve the problem, through the definition of the problem, mapping of the desires and responsibilities, definition of the artifact requirements and validation were extremely valuable, to understand the problem more deeply, develop a prototype and validate it, which demonstrates that the methodology applies to the development of artifacts linked to V & V processes for software.

The proposed objectives of increasing productivity, defining V & V techniques, and increasing efficiency in detecting defects in this project were achieved. With the six modules, the objectives of creating test steps for software validation, increasing the productivity of the test teams, and increasing the effectiveness of the V & V process in the automotive environment were achieved through the tools made available and evaluated by the specialists, in which there was an acceptance in the survey performed, through the activities sequenced in ProTSA, such as creating reports on V & V status, analysing documentation to detect software defects early, providing evidence for prioritizing features and activities, and promoting alignment between development and V & V teams.

The results achieved are relevant for software development and V & V teams in the automotive environment. For V & V teams, it is possible to indicate that the application of ProTSA will structure the sequence of testing activities and promote alignment and evidence-based decision-making.

For future applications of ProTSA, it is necessary to consider that the process is a prototype version, and its online version has been developed on a demonstrative basis to be evaluated by specialists. Therefore, its application in a company that needs a V & V process requires its improvement and some refinements, as suggested by the survey participants.

However, it is important to reinforce the limitations of this research, since the ProTSA was evaluated by a group of specialists from the same company and knowledgeable in the fields of powertrain, chassis, and safety ECUs.

For future studies, an evolution of ProTSA is expected, where it addresses strategies for standards related to cybersecurity, dynamic software updates and autonomous vehicles.

Acknowledgements

We are grateful to anonymous referees for their comments and suggestion, and the valuable contribution of the survey participation. We thank also the UNIFESP to provide the possibility of developing this study.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Mossinger, J. (2010) Software in Automotive Systems. *IEEE Software*, **27**, 92-94. https://doi.org/10.1109/MS.2010.55

[2] Navet, N. and Lion, F.S. (2009) Automotive Embedded Systems Handbook. CRC Press.

[3] Sommerville, I. (2011) Software Engineering. Pearson.

[4] Lochau, M., Lity, S., Lachmann, R., Schaefer, I. and Goltz, U. (2014) Delta-Oriented Model-Based Integration Testing of Large-Scale Systems. *Journal of Systems and Software*, **91**, 63-84. https://doi.org/10.1016/j.jss.2013.11.1096

[5] Warren, T. (2024) Software-Related Recalls and the Auto Industry's Ongoing Evolution. https://www.wardsauto.com/industry-news/software-related-recalls-and-auto-industry-s-ongoing-evolution

[6] Knauf Industries Automotive (2024) Automobile Construction. Cars with a Defined Software Package—A New Phenomenon in the Automotive Industry. https://knaufautomotive.com/pt-br/carros-com-um-pacote-de-Software-definido/

[7] Pretschner, A., Broy, M., Kruger, I.H. and Stauner, T. (2007) Software Engineering for Automotive Systems: A Roadmap. *Proceedings of the Future of Software Engi-*

neering, Minneapolis, 23-25 May 2007, 55-71. https://doi.org/10.1109/FOSE.2007.22

[8] Delamaro, M., *et al.* (2007) Introduction to Software Testing. Campos.

[9] Killian, R. (2021) What Is a TUV Certification? https://blog.1000bulbs.com/home/what-is-tuv-certification#:~:text=A%20TUV%20certification%20means%20a%20sampling%20of%20the,looked%20for%20a%20UL%20wet%20location%20safety%20rating

[10] Institute of Electrical and Electronics Engineers (1990) 610.12-1990-IEEE Standard Glossary of Software Engineering Terminology.

[11] Redmill, F., *et al.* (2022) Sysml Open Source Project: What Is SysML? Who Created SysML? https://sysml.org/?msckid=45dcb1bece6211eca1163fb9a4534962

[12] Bahig, G. and El-Kadi, A. (2017) Formal Verification of Automotive Design in Compliance with ISO 26262 Design Verification Guidelines. *IEEE Access*, **5**, 4505-4516. https://doi.org/10.1109/ACCESS.2017.2683508

[13] Synopsys (2021) What Is ASIL (Automotive Safety Integrity Level)? https://www.synopsys.com/automotive/what-is-asil.html

[14] Arcanjo, R.R., Martins, L.E.G. and Fernandes, D.L.G. (2024) Verification and Validation of Embedded Software in an Automotive Context: A Systematic Literature Review. *Multidisciplinary Scientific Journal Knowledge Core*, **3**, 207-250. https://doi.org/10.32749/nucleodoconhecimento.com.br/computer-science/embedded-software

[15] Rajabli, N., Flammini, F., Nardone, R. and Vittorini, V. (2020) Software Verification and Validation of Safe Autonomous Cars: A Systematic Literature Review. *IEEE Access*, **9**, 4797-4819. https://doi.org/10.1109/ACCESS.2020.3048047

[16] Haghighatkhah, A., Banijamali, A., Pakanen, O.-P., Oivo, M. and Kuvaja, P. (2017) Automotive Software Engineering: A Systematic Mapping Study. *Journal of System and Software*, **128**, 25-55. https://doi.org/10.1016/j.jss.2017.03.005

[17] Dresch, A., Lacerda, D.P. and Antunes, J.A.V. (2015) Design Science Research: A Method for Science and Technology Advancement. Springer.

[18] Jotform Blog (2023) Questions and Examples for Surveys Using the Likert Scale. https://www.jotform.com/pt/blog/exemplo-de-pesquisa-em-escala-likert/

[19] Iqbal, D., Abbas, A., Ali, M., Khan, M.U.S. and Nawaz, R. (2020) Requirement Validation for Embedded Systems in Automotive Industry through Modeling. *IEEE Access*, **8**, 8697-8719. https://doi.org/10.1109/ACCESS.2019.2963774

[20] Kriebel, S., *et al.* (2018) Improving Model-Based Testing in Automotive Software Engineering. *Proceedings of the* 40*th International Conference on Software Engineering: Software Engineering in Practice*, Gothenburg, 27 May-3 June 2018, 172-180. https://doi.org/10.1145/3183519.3183533

[21] Jordan, C.V., Mäurer, F., Löwenberg, S. and Provost, J. (2019) Framework for Flexible, Adaptive Support of Test Management by Means of Software Agents. *IEEE Robotics and Automation Letters*, **4**, 2754-2761. https://doi.org/10.1109/LRA.2019.2918486

[22] Volkswagen (2024) Equipment Levels-Polo. https://www.vw.com.br/pt/configurador.html/__app/polo.app

[23] Mercedes-Benz do Brasil (2023) Product Line. https://www.mercedes-benz-trucks.com.br/caminhoes/

[24] Vieira, H. (2024) Learn about the Importance of the Software Development Life

Cycle (SDLC) and It's Stages. https://www.objective.com.br/insights/sdlc/

[25] Da Rocha, A.R.C., *et al.* (2001) Software Quality: Theory and Practice. Prentice-Hall, 303 p.

[26] Galdino, T.S. (2010) The Importance of Testing in the Software Production Process. Faculty of Technology, Americana, Sao Paulo. https://ric.cps.sp.gov.br/bitstream/123456789/1709/1/20102S_GALDINOThiagoSecomandi_TCCPD1002.pdf

[27] Marques, L.R. (2015) Application of Quality Tools in the Diagnosis of Critical Success Factors in Industrial Projects: A Case Study in an Automotive Company. Monograph: Specialization in Production Engineering-Federal Technological University of Parana. https://portaldeinformacao.utfpr.edu.br/Record/riut-1-18822/Description#tabnav

[28] DevMedia (2021) Software Functional Tests. https://www.devmedia.com.br/testes-funcionais-de-software/23565