



Revista Facultad de Ingeniería

ISSN: 0121-1129

ISSN: 2357-5328

Universidad Pedagógica y Tecnológica de Colombia

Otalora-Luna, Jorge-Enrique; Valero-Bustos, Helver-Augusto; Callejas-Cuervo, Mauro
Methodological Proposal to Determine Technology Maturity Levels TRL 4 to TRL 7 for Mobile Applications
Revista Facultad de Ingeniería, vol. 32, no. 64, 3, 2023, April-June
Universidad Pedagógica y Tecnológica de Colombia

DOI: <https://doi.org/10.19053/01211129.v32.n64.2023.15681>

Available in: <https://www.redalyc.org/articulo.oa?id=413975585003>

- How to cite
- Complete issue
- More information about this article
- Journal's webpage in redalyc.org

UPEM  redalyc.org

Scientific Information System Redalyc
Network of Scientific Journals from Latin America and the Caribbean, Spain and
Portugal

Project academic non-profit, developed under the open access initiative

Revista Facultad de Ingeniería

Journal Homepage: <https://revistas.uptc.edu.co/index.php/ingenieria>



Methodological Proposal to Determine Technology Maturity Levels TRL 4 to TRL 7 for Mobile Applications

Jorge-Enrique Otalora-Luna¹

Helver-Augusto Valero-Bustos²

Mauro Callejas-Cuervo³

Received: February 02, 2023

Accepted: May 01, 2023

Published: May 09, 2023

Citation: J.-E. Otalora-Luna, H.-A. Valero-Bustos, M. Callejas-Cuervo, "Methodological Proposal to Determine Technology Maturity Levels TRL 4 to TRL 7 for Mobile Applications," *Revista Facultad de Ingeniería*, vol. 32, no. 64, e15681 2023.
<https://doi.org/10.19053/01211129.v32.n64.2023.15681>

Abstract

Industries undergoing transformation strategies for improvement can use mobile applications (Apps) to enhance production efficiency, ensure greater coverage, and optimize costs and times. These crucial aspects are imperative to foster confidence

¹ Ph. D. Universidad Pedagógica y Tecnológica de Colombia (Tunja-Boyacá, Colombia). jorge.otalora@uptc.edu.co. ORCID: [0000-0001-5824-1753](https://orcid.org/0000-0001-5824-1753)

² M. Sc. Universidad Pedagógica y Tecnológica de Colombia (Tunja-Boyacá, Colombia). helver.valero@uptc.edu.co. ORCID: [0000-0002-9749-2165](https://orcid.org/0000-0002-9749-2165)

³ Ph. D. Universidad Pedagógica y Tecnológica de Colombia (Tunja-Boyacá, Colombia). mauro.callejas@uptc.edu.co. ORCID: [0000-0001-9894-8737](https://orcid.org/0000-0001-9894-8737)

among stakeholders. To satisfy this need, maturity assessment models such as the Technology Readiness Levels (TRL) have been developed. This article proposes a methodology to ease the determination of mobile app maturity by mapping them to TRL levels 4 to 7. A review of TRL's application to software products, including mobile applications, indicates that research on it has been conducted but confirmed the absence of a methodology to evaluate their development maturity. With this input in mind, we reviewed different methodologies employed for technological assessment in Apps, selected the most suitable, and designed a set of activities and artifacts that constitute the tool. The methodology was validated through the evaluation of a technological product, and its ability to assess technological maturity at TRL levels 4 to 7 was confirmed. Consequently, we conclude that having tools such as the one presented here is of paramount importance to support research and innovation processes, ensure technological product quality, and comply with the TRL model.

Keywords: backend; frontend; mobile applications; technology maturity levels; user interfaces.

Propuesta metodológica para determinar los niveles de madurez tecnológica TRL 4 a TRL 7 para aplicaciones móviles

Resumen

Las industrias dentro de sus estrategias de transformación para la mejora, pueden apoyarse en el uso de aplicaciones móviles (Apps), cuya calidad, es fundamental para la disminución de errores en producción, garantizar mayor cobertura y optimización de costos y tiempos; aspectos importantes para la generación de confianza en los involucrados; a partir de esta necesidad, surgen modelos de evaluación de la madurez, como, por ejemplo, Technology Readiness Levels (TRL), que ha sido acogido por entidades como el Ministerio de Ciencias, Tecnología e Innovación en Colombia, con el fin de identificar el alcance de las actividades asociadas a la investigación, el desarrollo tecnológico y la Innovación (I+D+i) de los proyectos que le son presentados. Cada desarrollo tecnológico tiene sus

particularidades y las aplicaciones móviles no son la excepción, razón por la cual es deseable, contar con elementos que permitan la evaluación de madurez de las Apps, basadas en el modelo TRL. Por esta razón se plantea la construcción de una metodología que busca facilitar la determinación de la madurez de una App, mediante el mapeo en los Niveles de Madurez Tecnológica (TRL4 al TRL7). Para lograr esta meta, se realizó una revisión sistemática de la adopción de TRL a los productos de software incluyendo las aplicaciones móviles, encontrando algunas investigaciones que se tomaron de base, pero reafirmando la ausencia de una metodología, que abordara de forma amplia el uso de aplicaciones móviles; una vez se contó con estos insumos, se procedió a revisar los diferentes métodos, técnicas y herramientas usadas en la evaluación tecnológica de software aplicables a móviles, seleccionando las más apropiadas, para luego diseñar una serie de actividades y artefactos que componen la herramienta que se validó a través de la evaluación de un producto tecnológico dentro de un proyecto de convocatoria de Minciencias, dando como resultado el poder realizar la valoración de la madurez tecnológica en los niveles del 4 al 7 dentro del modelo TRL, y presentando a la comunidad académica y científica un producto replicable, aplicable y adaptable a productos tecnológicos similares. Finalmente se puede concluir que es muy importante contar con herramientas como la presentada aquí, para apoyar los procesos de investigación e innovación, asegurando la calidad de los productos tecnológicos y cumplir lo planteado en el modelo TRL.

Palabras clave: aplicaciones móviles; interfaces de usuario; interfaz; niveles de madurez de la tecnología; servidor.

Proposta metodológica para determinar os níveis de maturidade tecnológica TRL 4 a TRL 7 para aplicativos móveis

Resumo

As indústrias dentro de suas estratégias de transformação para a melhor, podem apoiar-se no uso de aplicativos móveis (Apps), cuja qualidade, é fundamental para a diminuição de erros na produção, garantir maior cobertura e otimização de custos

e tempos; aspectos importantes para a geração de confiança nos envolvidos; a partir desta necessidade, surgiram modelos de avaliação da madurez, como, por exemplo, Technology Readiness Levels (TRL), que foi aprovado por entidades como o Ministério de Ciências, Tecnologia e Inovação da Colômbia, com o objetivo de identificar o alcance das atividades associadas à investigação, ao desenvolvimento tecnológico e à Inovação (I+D+i) dos projetos que são apresentados. Cada desenvolvimento tecnológico tem suas particularidades e os aplicativos móveis não são a exceção, razão pela qual é desejável, contar com elementos que permitem a avaliação de madurez dos aplicativos, baseados no modelo TRL. Por esta razão se planeja a construção de uma metodologia que busca facilitar a determinação da maturidade de um App, mediante o mapa nos Níveis de Madurez Tecnológica (TRL4 a TRL7). Para lograr esta meta, realiza-se uma revisão sistemática da adoção de TRL aos produtos de software incluindo os aplicativos móveis, encontrando algumas investigações que se tomam de base, mas reafirmando a ausência de uma metodologia, que aborda de forma ampla o uso de aplicativos móveis; uma vez que se contó com esses insumos, procede-se a revisar os diferentes métodos, técnicas e ferramentas usadas na avaliação tecnológica de software aplicável a dispositivos móveis, selecionando os mais apropriados, para depois projetar uma série de atividades e artefatos que compõem a ferramenta que se validou a través da avaliação de um produto tecnológico dentro de um projeto de convocatória de Minciências, dando como resultado o poder realizar a valorização da madurez tecnológica nos níveis de 4 a 7 dentro do modelo TRL, e apresentando à comunidade acadêmica e um produto científico replicável, aplicável e adaptável a produtos tecnológicos semelhantes. Finalmente se pode concluir que é muito importante contar com herramientas como a apresentada aqui, para apoyar los procesos de investigación e innovación, asegurando a qualidade dos produtos tecnológicos e cumplir lo plantado no modelo TRL.

Palabras-chave: aplicativos móveis; interfaces de usuário; interfaz; níveis de madurez de la tecnología; servidor.

I. INTRODUCTION

Technology Readiness Levels (TRLs) represent a widely used metric for assessing the maturity of a given technology [1-3]. Each project is evaluated against specific parameters for each level, and a TRL rating is assigned based on the project's progress. The scale ranges from TRL 1—the lowest level of maturity—to TRL 9—the highest level. This concept originated from NASA (National Aeronautics and Space Administration) but has since been used in the development of technological projects in various industries [4-8].

The literature has documented the development of several projects concerning the evaluation of technological maturity in diverse fields. For example, [9] proposes a methodology to assess different technologies in Artificial Intelligence by mapping them to different TRL levels and evaluating them through representative examples of AI technologies, from autonomous cars to virtual assistants. In [10], the TRL of a technology is determined by expert evaluation using the Delphi method, leveraging the knowledge and experience of professionals to assess the degree to which quality criteria are met. The primary objective of that work was to propose a technique and research methodology to support the TRL to evaluate technological maturity in the design and development phase of the technological life cycle.

The Turkish defense industry has implemented a validation mechanism to verify the TRL maturity of new technologies being developed in laboratories and the national industry [11]. They designed an algorithm to evaluate the experience of experts from the procurement office, which classifies questions about the technology being evaluated into critical and non-critical for each level so that the questions have different influences on the evaluation. The result was software to calculate TRL levels for systems engineering and a technology management tool. This tool allows technology developers to upload evidence such as proof of results, drawings, photos, or other documents related to the achieved level.

The application of TRL to software products is reviewed. It is noted that the problem lies in the original concept of TRL, which was originally focused on—or at least expressed in terms of—hardware [12]. As large, complex systems become more

dependent on software for performance-critical functionality, the application of TRL to software becomes more important. Both NASA and the Department of Defense have developed descriptions of software TRL, but both versions tend to describe the maturity of a specific product rather than the overall development of a technology. They review existing definitions of hardware and software TRL and propose an expanded definition of application to software technology, as opposed to software products.

II. METHODOLOGY

This methodology seeks for a way to evaluate the TRLs regarding the progress in the development of a mobile application by addressing the aspects considered relevant for the industry as well as for research, technological development, and innovation. In this sense, first, the development of mobile applications is framed within the discipline of software engineering, so its precepts are applicable and therefore what is proposed here could be adapted to other types of products such as web applications or local applications (desktops). The economic evaluation models are not part of the scope of this methodology.

This section outlines a pattern to implement the Software Technology Product Maturity Definition process in relation to TRL levels. To this end, the following steps are taken:

- First, the definition is deconstructed into phrases describing each relevant concept to determine the maturity of the product. These descriptions are referred to as "characteristics."
- Second, each identified characteristic is classified based on its nature; that is, whether it serves as an input for review or is a product derived from it. In other words, it is determined whether a given characteristic is a necessary element for carrying out the evaluation or an artifact that will be generated because of product testing.

The characteristics identified for the TRL classification must be aligned with those of the technological product under evaluation. In this particular case, as the subject is

a mobile application, it must be positioned within the context of software engineering practices and definitions [13-14]. It is worth noting that such an app typically consists of both front-end and back-end web applications, both should be considered in the adaptation process. Each feature may have several dimensions, and multiple features may be aligned with a single dimension; these intricacies must be considered to ensure the completeness of the mapping process.

Each resulting characteristic should be accompanied by the identified processes that enables assessing maturity, including the expected deliverables and, of course, the description of the minimum content, as follows: Activity, Type (input, output), Deliverable, and Characteristics.

For each activity, different metrics, scales, and weightings are defined according to the criteria implied by each characteristic evaluated within the level and also a quantifiable threshold of acceptance of the product within the TRL. The appropriate measurement activities are applied to the product and result in a set of recommendations and an evaluation of the level.

The recommendations must be reviewed by the product development team, who must accept or reject them and justify their decision. An appropriate time is given to correct the findings, and once this has been done, the assessment is repeated.

If the evaluation meets the previously defined threshold, the maturity level is granted, and the evaluation process can proceed to the next level. The activities proposed for each TRL to verify the maturity level are detailed below.

III. TRL 4 - VALIDATION OF COMPONENTS/SUB-SYSTEMS IN LABORATORY TESTS

As defined by [1], TRL 4 represents a crucial stage in the technology's development process. It is characterized by the identification of the individual components that constitute a technology, and the determination of whether these components are capable of functioning in an integrated manner as a system. At this stage, a prototype unit is constructed within a laboratory and a controlled environment. The resulting operations provide data to assess the potential for scaling up, as initial life

cycle and economic assessment models are pre-validated through the product design process.

Considering that the technological product refers to a mobile application with its respective content manager from a web application, the scope of TRL 4 is defined as:

- Identify the components that make up the solution.
- Determine the individual capabilities of these components.
- Measure the integrated performance of the components in the system.
- Identify the potential for expansion according to the software life cycle.

The review is performed at the laboratory based on the provided project documentation. This document is addressed to the project management team. Table 1 describes the activities proposed for the maturity review.

Table 1. Activities to verify the maturity level TRL 4.

Item	TRL 4 - Validation of components/sub-systems in laboratory tests	
1	Activity	Request the inventory of components (Architecture) that integrate the technology used for the construction of the mobile application (Backend + app).
	Type	Input
	Deliverable	Component inventory document
	Description	✓ Architecture Document ✓ Component Diagram ✓ Other design artifacts
2	Activity	Perform analysis and evaluation of the relationship between the identified components in terms of their capabilities and integration between them.
	Type	Output
	Deliverable	Document with compatibility and component capabilities analysis results.
	Description	✓ Analysis Document ✓ Record of results ✓ Component integration diagram ✓ Conclusions and recommendations
3	Activity	Request the prototype of the application along with the environment and documentation needed to get it up and running it.
	Type	Input
	Deliverable	Application prototype (Backend, app)
	Description	✓ Deployment Diagram ✓ Deployment document ✓ Software (apk, Backend distribution)
4	Activity	Install and configure the environment for running the delivered prototype (backend, application) in a local environment.
	Type	Output

Item	TRL 4 - Validation of components/sub-systems in laboratory tests	
	Deliverable	Document with findings from the process of replicating the prototype execution environments and recommendations.
	Description	✓ Platform used in the local environment. ✓ Local URL (controlled lab environment)
5	Activity	Identify the lifecycle scaling potential of the platform.
	Type	Output
	Deliverable	Technical concept on component scalability.
	Description	✓ Document of quantification and qualification of the system's growth potential.
6	Activity	Socialization of Phase results.
	Type	Output
	Deliverable	Formal presentation of results phase.
	Description	✓ Presentation and Discussion of Results ✓ Minutes and commitments

A. Identification of System Components

The identification of the system components is derived from the analysis of the information repository made available by the development team. This information repository encompasses various aspects of the system, including the architecture of the backend and frontend application, the process view, and the navigability tree of the administration module. Through this process, both the physical and logical components that constitute the system are identified, which enable creating the Component Inventory Document.

1) Physical Component. The hardware artifact that houses the logic components, such as servers and mobile devices, is the subject of discussion. In a laboratory setting, machines with lower performance can be used to simulate the actual deployment environment and conducting concept and functional testing [15].

2) Logical Component. The software artifact is an essential component that provides the processing capability required for the system to function. It comprises algorithms, data, documentation, and other related items. General-purpose components—such as those that enable hardware management—can be employed, as well as specific-purpose components—such as those that constitute an application [16].

B. Scalability Analysis

Outlines the outcome of an examination conducted on the architecture of a system

and considers both its logical and physical scalability. Specifically, software scalability is assessed, i.e., the system's capacity to accommodate a growing workload and increasing amounts of data. Basically, software scalability is the ability of a system to meet its functional and non-functional requirements as it expands [17].

C. Interoperability Analysis

Interoperability Analysis is the technical capacity of two or more systems or components to exchange information and employ it. When used in the software context, interoperability refers to the ability of different programs to exchange data using standardized interchange formats, read and write identical file formats, and use the same protocols [18].

IV. TRL 5 - VALIDATION OF SYSTEMS, SUBSYSTEMS, OR COMPONENTS IN A RELEVANT ENVIRONMENT

In accordance with [1], TRL 5 is defined as "The basic components of a given technology are integrated in such a way that the final configuration resembles its ultimate application, and is, therefore, ready to be employed in simulating a real environment. At this stage, both technical and economic models for the initial design have been refined, and considerations such as safety, environmental, and/or regulatory constraints have been additionally identified. However, the functionality of the system and technologies is still at the laboratory level." The key difference between TRL levels 4 and 5 is the increased faithfulness of the system and its environment to the final application.

In the case of a technological product involving a mobile application with its corresponding content manager from a web application, the TRL 5 scope encompasses the following:

- Assessing the performance of the solution in a laboratory environment.
- Identifying relevant aspects related to usability, security, environmental and regulatory concerns.

The maturity review will be conducted at the laboratory based on the project

documentation. This document is intended for the project management team; therefore, it does not delve into the technical details of the review. Table 2 outlines the proposed activities for the maturity review.

Table 2. TRL 5 maturity level verification activities.

Item	TRL 5 - Validation of systems, subsystems, or components in a relevant environment	
1	Activity	Request prototype version update with recommendations from the previous phase.
	Type	Input
	Deliverable	Updated prototype version
	Description	✓ Architecture Document ✓ Component Diagram ✓ Deployment elements ✓ Server Provisioning ✓ Other
2	Activity	Install and configure the environment to execute the delivered prototype (Backend, app) in a cloud environment.
	Type	Output
	Deliverable	Document with findings on the process of replicating the prototype execution environments and recommendations.
	Description	✓ Platform deployed in the cloud environment ✓ A public URL (controlled lab environment)
3	Activity	Ask for System Requirements
	Type	Input
	Deliverable	System requirements document
	Description	✓ Functional Requirements ✓ Non-Functional Requirements ✓ User Stories ✓ Other Requirements
4	Activity	Functional test
	Type	Output
	Deliverable	Verification document of the application functionalities according to the requirements specification.
	Description	✓ Functional requirements compliance checklist by module. ✓ Test cases by use scenarios.
5	Activity	Performance tests
	Type	Output
	Deliverable	Server and Network performance report.
	Description	✓ Load testing ✓ Identification of critical points
6	Activity	Memory tests
	Type	Output
	Deliverable	Report optimized memory usage in the application.
	Description	✓ Memory consumption ✓ Consumption peaks
7	Activity	Interruption tests
	Type	Output
	Deliverable	Report behavioral interruptions while the application is running.

Item	TRL 5 - Validation of systems, subsystems, or components in a relevant environment	
	Description	✓ Incoming calls or SMS ✓ Low memory warning ✓ Low battery warning ✓ Internet flicker detection
8	Activity	Setup tests
	Type	Output
	Deliverable	Setup Process Verification Report
	Description	✓ Ease of setup process ✓ Problems during setup, including upgrading and uninstallation
9	Activity	Usability tests
	Type	Output
	Deliverable	Preliminary usability report.
	Description	✓ Efficiency ✓ Effectiveness
10	Activity	
	Type	Security tests
	Deliverable	Application Safety Report
	Description	✓ Validate application resistance to attacks by malicious users. ✓ List of vulnerabilities ✓ Elastic testing ✓ Dynamic testing ✓ Forensic testing
11	Activity	Review of standards and regulations.
	Type	Output
	Deliverable	ICT application regulations report.
	Description	✓ App regulations ✓ Data processing ✓ Others
12	Activity	Socialization of phase results.
	Type	Output
	Deliverable	Formal presentation of results phase.
	Description	✓ Presentation and discussion of results ✓ Minutes and commitments

To meet the requirements of TRL 5, in addition to functionality, aspects such as usability, security, and regulations will be examined.

A. System Test Strategy

The system test strategy entails designing tests tailored to each system module to ascertain their functional performance and timely response as per the provided requirements. Subsequently, an automated test environment is established, and corresponding scripts are created.

B. Test Design

After applying tests to the system, an Incident Consolidation document is created to compile the findings. The review dimensions are configured based on the application's navigation map and incidents are categorized as follows:

- Usability: measures the effectiveness, efficiency, and satisfaction of user interactions with the product.
- Information Integrity: refers to the accuracy and reliability of data, which must be complete and unchanged from the original.
- Performance: quantifies the amount of work performed by a computer system in relation to the time and hardware resources used.
- Regulations: refers to the official rules that must be followed in specific contexts. For computer applications, consumer regulations, personal data protection, electronic commerce, and intellectual property are reviewed.
- Security: encompasses application-level security measures designed to prevent data or code theft or hijacking within the application. It covers security considerations that should be considered during the development and design of applications, as well as systems and approaches for to protect them after deployment.

The report can be organized according to the application's described modules, and incidents are prioritized as high, medium, or low.

V. TRL 6 - VALIDATION OF THE SYSTEM, SUBSYSTEM, MODEL, OR PROTOTYPE UNDER NEAR-REAL CONDITIONS

In accordance with [1], TRL 6 is defined as the stage at which pilot prototypes can execute all the necessary functions within a given system, having passed feasibility tests under real operating conditions. Components and processes may have been scaled up to demonstrate their industrial potential in real systems. While available documentation may be limited, it can begin with the prototype, which has been tested under conditions very close to the expected, identified, and modeled at full

commercial scale, refining the life cycle assessment and economic evaluation. This stage is also known as the "beta test demonstration." The scope for TRL 6 includes:

- Installation and configuration of the environment for running the delivered prototype (backend, application) in a beta test environment.
- Configuration of test strategy.
- Review of the base prototype by end users in terms of standards, security, and usability.

Table 3. Activities to verify the maturity level TRL 6.

Item	TRL 6 - Validation of the system, subsystem, model, or prototype under near-real conditions	
1	Activity	Install and configure the environment to execute the delivered prototype (Backend, app) in a Beta test environment.
	Type	Input
	Deliverable	Beta versions installed
	Description	✓ Beta for Android ✓ Beta for IOS
2	Activity	Characterization of users
	Type	Output
	Deliverable	User Profiles document
	Description	✓ User Roles ✓ User types ✓ Sample determination
3	Activity	User selection
	Type	Output
	Deliverable	List of users.
	Description	✓ User data ✓ Role ✓ Type of user
4	Activity	Test strategy configuration
	Type	Output
	Deliverable	Test strategy
	Description	✓ Free Testing ✓ Guided testing ✓ Test cases
5	Activity	Incident log platform configuration
	Type	Output
	Deliverable	Incident repository
	Description	✓ Forms creation ✓ Storage ✓ Management
6	Activity	Test Application
	Type	Output
	Deliverable	Repository of performed tests
	Description	✓ Logs and evidence
7	Activity	Incident analysis

Item	TRL 6 - Validation of the system, subsystem, model, or prototype under near-real conditions	
8	Type	Output
	Deliverable	Beta test report
	Description	✓ Usability ✓ Security ✓ Performance ✓ Others
	Activity	Socialization of Phase results.
8	Type	Output
	Deliverable	Formal presentation of phase results.
	Description	✓ Presentation and discussion of results ✓ Minutes and commitments
	Activity	Socialization of Phase results.

To fulfill the criteria of TRL 6, beyond assessing functionality, various aspects such as usability, security, and regulations will also be scrutinized.

A. System Test Strategy

Tests are designed according to each system module to verify that they will perform their function and that the response will be made in a time according to the usability standards.

- Test cases are written according to the requirements provided as input.
- Test users were selected for manual testing according to the methodology proposed below.

Proposed testing methodology: According to Nielsen [19], the following types of tests can be performed with users, a scheme can be seen in Figure 1.

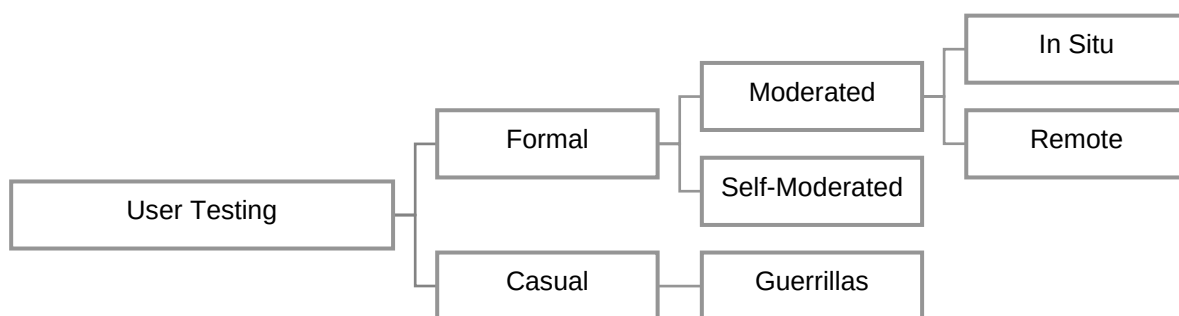


Fig. 1. Types of user testing (adapted from Nielsen).

The following is a brief description of each type of testing:

Guerrilla: quick and informal tests that are conducted with little planning, usually in public places such as bars or coffee shops, where people are asked to test a prototype, website, or application for a few minutes [20-21].

Moderated In Situ: Moderated user tests are formal tests conducted in a usability laboratory or in the context of real user use, these require prior planning. During the tests, the moderator will accompany the user and will indicate the tasks to be performed, another person from the team observes the level of effectiveness and efficiency with which they manage to perform the tasks.

Moderated Remote: like the previous test, except that the users and the moderator are not in the same location and the moderation is done via video call.

Self-moderated: There is no moderator intervention, the user receives instructions on the tasks to be performed through the interface of the application in which the test is developed.

Characterization of the product under evaluation: The product to be evaluated consists of two pieces of software with a user interface (UI):

Web app: A web application that allows the management of the backend by managing the users and the content of the mobile application.

Mobile app: A mobile application available on the Android or iOS platform that allows business customers to access and manage each offered feature.

B. User Characterization

The product users exhibit diverse characteristics that must be considered in testing; then, it is necessary to determine the most relevant ones. User groups are described below.

- **Users of the web application:** This group has limited access with a deep understanding of business operations and the potential for adequate training.
- **Users of the mobile application:** This group has a medium potential volume of users (1000) with different profiles. Testing will be conducted based on the following criteria:

- ✓ Technological platform used: it will enable an evaluation of the impact of differences on the user interface across different platforms. Users with iOS devices and users with Android devices will be tested separately.
- ✓ Characteristics of the device: variations in device size may impact the UI behavior and usability. Testing will be performed on iPhone and tablet devices.
- ✓ Company user profile: the account type determines the services and products available to users. The type of user (age, gender, education level, special) will also be considered.
- ✓ Previous knowledge of the company: testing will be performed on users who are already familiar with the company's operations (i.e., current customers) and users who have no knowledge of the business.

C. Users Sample Definition for Usability Tests

To conduct usability tests, it is essential to establish a representative sample of users based on their characteristics, as detailed below:

- **Web Application Users:** As the application targets a restricted group of users with comprehensive knowledge of the business operations, the usability testing will apply a moderated remote test to two randomly selected users, one of whom will have an administrator profile and the other an operator profile. Additionally, a moderated face-to-face test will be conducted with one user having an operator profile and no prior knowledge of the business but knowing about usability.
- **Mobile Application Users:** For users of the mobile application, moderated remote tests will be conducted as per the distribution illustrated in Figure 2.

Methodological Proposal to Determine Technology Maturity Levels TRL 4 to TRL 7 for Mobile Applications

Characteristic		User												
Type of test	Detail	1	2	3	4	5	6	7	8	9	10	11	12	13
Platform	AIOs device	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☒	☐	☐
	Android device	☐	☐	☐	☐	☐	☐	☐	☐	☐	☒	☐	☐	☐
Device	Iphone	☐	☐	☐	☐	☐	☐	☐	☐	☒	☐	☐	☐	☐
	Tablet over 7"	☐	☐	☐	☐	☐	☐	☐	☒	☐	☐	☐	☐	☐
Previous knowledge of the business	Know the business	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☒	☐
	Does not know the business	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☒
Account profile	Free account	☐	☐	☐	☐	☐	☒	☐	☐	☐	☐	☐	☐	☐
	Account paid	☐	☐	☐	☐	☐	☐	☒	☐	☐	☐	☐	☐	☐
Generation	Generation Z	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
	Millennials	☐	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
	Generation X	☐	☐	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
	Baby Boomers	☐	☐	☐	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐
	Silent generation	☐	☐	☐	☐	☒	☐	☐	☐	☐	☐	☐	☐	☐
Gender	Male	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
	Female	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Fig. 2. Distribution of user characteristics to be selected.

Since the characteristics are not mutually exclusive, the sample requires at least one user who meets one of them. It means that the minimum number of users is 6 and the maximum is 14, for a sample size of 10 users. Complementary tests are also performed, and the total number of user tests is 15. Self-moderated tests should be performed by a user with good usability knowledge.

VI. TRL 7 - DEMONSTRATION OF A VALIDATED SYSTEM OR PROTOTYPE IN A REAL-WORLD OPERATIONAL ENVIRONMENT

TRL 7 is defined as "The system is in or close to the pre-commercial operation. It is possible to carry out the phase of identification of manufacturing aspects, life cycle assessment, and economic evaluation of technologies, with most of the functionalities available for testing. The available documentation may be limited, but the technology has been demonstrated to work and operate at a pre-commercial scale, the life cycle assessment and economic development have been refined. At this stage, the first pilot run and actual final testing are underway" [1].

A pilot test enables testing a new system by applying it to only one business process. This allows the company to determine whether the system is intuitive and whether users can adapt well. If the system is deemed effective, it can be rolled out to other processes within the company. Conversely, if the software is deemed slow or not

optimized, the company will need to evaluate whether it is worth implementing the software for the rest of the organization. Bearing this definition in mind, the pilot test is conducted on the prototype baseline, considering what is presented in Table 4.

Table 4. TRL 7 maturity level verification activities.

Item	TRL 7 - Demonstration of a validated system or prototype in a real-world operational environment	
1	Activity	Install and configure the environment for running the prototype with the corrections from the previous beta test in a beta test environment.
	Type	Input
	Deliverable	Installed beta versions
	Description	✓ Android Beta ✓ iOS Beta
2	Activity	Preparation of free tests
	Type	Output
	Deliverable	Test plan
	Description	✓ Free tests ✓ Testing period ✓ User selection outside beta testers.
3	Activity	Test execution
	Type	Output
	Deliverable	Incident repository
	Description	✓ Forms creation ✓ Incident storage ✓ Incident Management
4	Activity	Incident analysis
	Type	Output
	Deliverable	Pilot Test Report
	Description	✓ User Experience ✓ Product Security ✓ Performance ✓ Other issues
5	Activity	Socialization of phase results.
	Type	Output
	Deliverable	Formal presentation of the phase results.
	Description	✓ Presentation and Discussion of Results. ✓ Minutes and commitments

Test users were selected based on the methodology proposed for manual testing, as described above. These users were required to have a means of documenting incidents and providing feedback regarding the system's functionality. Furthermore, they were expected to report on whether they were able to successfully execute the application's functions. Upon completion of the testing phase, a consolidation document was created to evaluate the technological product's usability through a

quantitative analysis.

VII. RESULTS

The results obtained by implementing the proposed methodology are described below.

A. TRL 4 - Validation of Components/Sub-systems in Laboratory Tests

After performing each of the activities outlined in the methodology guiding this research, the authors obtained the following items.

1) Scalability Analysis. The criteria used to determine the maturity level of TRL 4 are evaluated qualitatively using the values: Excellent, Good, Fair, and Poor. Tables 5 and 6 present the criteria and item descriptions for scalability analysis of the backend, while Tables 7 and 8 show the same for the frontend.

Table 5. Backend Scalability Analysis Criteria.

Criterion 1	General Architecture Agility
Description	General agility is the ability to respond quickly to an ever-changing environment.
Criterion 2	Ease of deployment
Description	The ability to introduce changes without overestimating the development effort.
Criterion 3	Ease of Test Deployment
Description	Ability to apply test cases to each component independently.
Criterion 4	Scalability
Description	Ability to easily scale the application without losing speed and responsiveness.
Criterion 5	Growth and/or expansion
Description	Ability to add or extend the logical capabilities of the system.
Criterion 6	Ease of development
Description	Speed of implementation and adaptation of functional requirements.

Table 6. Criteria for scalability analysis of the physical components of the Backend

Criterion 1	Vertical Growth
Description	The ease of increasing the capacity of infrastructure resources.
Criterion 2	Horizontal Growth
Description	The ease of increasing infrastructure resources through instance replication.

Table 7. Criteria for Frontend Scalability Analysis

Criterion 1	General Architecture Agility
Description	General agility is the ability to respond quickly to an ever-changing environment.
Criterion 2	Ease of deployment
Description	The ability to introduce changes without overestimating the development effort.

Criterion 3	Ease of Test Deployment
Description	Ability to apply test cases to each component independently.
Criterion 4	Scalability
Description	Ability to easily scale the application without losing speed and responsiveness.
Criterion 5	Growth and/or expansion
Description	Ability to add or extend the logical capabilities of the system.
Criterion 6	Ease of development
Description	Speed of implementation and adaptation of functional requirements.

Table 8. Criteria for scalability analysis of Frontend physical components

Criterion 1	Adaptability to Hosting Infrastructure
Description	Ability to host on a variety of mobile devices
Criterion 2	Physical resource consumption
Description	Level of resource requirements (storage, processing, and connectivity).

2) Interoperability Analysis. Tables 9 and 10 show the criteria and item descriptions for the backend interoperability analysis, and Tables 11 and 12 show the criteria and item descriptions for the frontend interoperability analysis.

Table 9. Criteria for Backend Logical Component Interoperability Analysis.

Criterion 1	Data Exchange Formats
Description	Formats are files used by applications to transport large amounts of information between components. The structure of each format is different, and these differences determine the advantages or disadvantages of one format over another.
Criterion 1	Data Interchange Formats
Description	Formats are files that applications use to transport large amounts of information between components. Each format has a different structure, and these differences determine the advantages or disadvantages of one over the other.
Criterion 3	Distributed development
Description	The ability to build components by development teams in remote locations.

Table 10. Criteria for interoperability analysis of Physical Backend components

Criterion 1	Application server configuration
Description	Hardware devices that can support the application server and database server.
Criterion 2	Data traffic
Description	Bandwidth supports concurrent sessions and traffic flow.

Table 11. Criteria for interoperability analysis of Frontend Logic components

Criterion 1	Interoperability between dependencies and plugins
Description	Dependencies on libraries, Apis, or other tools can cause interoperability problems.
Criterion 2	Performance issues
Description	Speed of response to application functionality.
Criterion 3	Application size
Description	The size of the executable application during installation, loading, and operation.

Table 12. Criteria for interoperability analysis of Frontend Physical components

Criterion 1	Device architecture adaptability
Description	Ability to run on multiple mobile device technologies
Criterion 2	Physical resource consumption
Description	Level of resource requirements (storage, processing, and connectivity)

B. TRL 5 - Validation of Systems, Subsystems, or Components in a Relevant Environment

A laboratory environment is established, and the backend is deployed on a web server with the specifications described in the input documents. This is done to simulate the minimum conditions necessary to execute the Content Manager. A test environment is also created with emulators of various Android mobile device types, and the application is installed on physical devices of different sizes and ranges to observe its behavior in these settings.

After the environment is set up, the development team's requirements and the navigability map are reviewed. Based on these, test cases are written and executed on each selected device. The results are documented, and the relevant evidence is attached. This process is carried out with at least two testers who perform the tests independently. Any duplicate incidents are combined, and the resulting list is consolidated into a single document.

C. TRL 6 - Validation of System, Subsystem, Model, or Prototype Under Near-real Conditions

The following is a description of the design we plan to use:

- Informed consent is required.
- Test modality: thinking by speaking.

The user will be asked to use the application and to verbally state everything he/she thinks while using it; the user's behavior will be recorded.

- Test time: 20 minutes.
- Mode of communication: Video conference.
- Activities to be performed by the user: The user will be told which activities

or functionalities to use.

- Mechanism for recording results: Recording of the test.

D. TRL 7 - Demonstration of a Validated System or Prototype in a Real-world Operational Environment

The following is a description of the proposed design:

- Informed consent is required.
- Test modality: thinking by speaking.

The user is asked to use the application and to verbally state everything he/she thinks while using it; the user's behavior is recorded.

- Test time: 20 minutes.
- Mode of communication: Video conference.
- Activities to be performed by the user: The user will be told which activities or functionalities to use.
- Mechanism for recording results: Recording of the test.

VIII. DISCUSSION

In accordance with the methodology described by Martinez-Plumed [9], various Artificial Intelligence (AI) technologies are categorized and evaluated by mapping them by Technology Readiness Levels (TRL). This assessment is intended to determine the feasibility of implementing these technologies in practical applications while considering factors such as data availability, algorithm accuracy and effectiveness, scalability, and ease of real-world implementation. A methodology is a valuable tool for companies and organizations looking to implement AI technologies in their business and evaluate the maturity of available solutions in the market. This research seeks to identify general dimensions that can represent different layers of technical breadth, making the methodology adaptable to any type of software evaluation.

In their study, Sarfaraz et al. [10] suggest using the Delphi method to evaluate the maturity of a technology in relation to its TRL. This method involves assembling a

panel of subject matter experts who are asked a series of questions. They respond anonymously and their answers are analyzed to provide feedback. The process is repeated until a consensus on the answers is reached. The Delphi method seeks to assess the maturity of a technology in comparison to its TRL.

The feedback provided by experts can encompass various factors that influence the maturity of a technology, such as data availability, accuracy, scalability, and ease of real-world implementation. The Delphi method enables thorough discussion and feedback among experts, which can aid in identifying areas of consensus and disagreement and improving understanding of the technology. The research proposes a methodology that utilizes a comprehensive documentary baseline to equip evaluators with sufficient reference elements. This framework reduces the subjectivity associated with the Delphi method by incorporating metrics and rating scales. As a result, it does not require a panel of experts, thereby reducing the cost of implementation and mitigating the effects of bias that may arise.

The TRL assessment process can be a complex and multifaceted evaluation involving several factors, and different industries may have specific requirements and considerations when assessing the maturity of a technology. The Turkish defense industry has provided a valuable tool for TRL assessment based on the process experience of Taner Altunok [11]. However, this model is closed and not configurable, which limits its adaptability to other domains.

While the tool proposed by Taner Altunok may provide an accurate and detailed assessment of the maturity of technology within the Turkish defense industry, it is not possible to change the critical questions in the model. This lack of flexibility may weaken the objectivity of the results. To address the issue, this research proposes a methodology that leaves its processes open, allowing for the redesign of guidelines according to the evaluators' needs.

Armstrong [7] provides a conceptual framework to adopt TRL in the evaluation of hardware maturity. Over time, it has been adapted to other areas, such as software and information technology in general. The methodology considers these changes in the application along with the guidelines provided by James R. Its objective is to

facilitate the assessment of the maturity of the technology in relation to its readiness for implementation in software or other areas.

IX. CONCLUSIONS

The present study proposes a methodology to determine the maturity of a software product by mapping it to the Technological Maturity Levels (TRL4 to TRL7), which includes the definition, scope, activities, and detail of the deliverables. The aim of this methodology is to improve coverage and reduce costs and time, thereby increasing stakeholder confidence in determining the maturity of a software product within the TRL parameters. The TRL methodology was provided by NASA and adopted in Colombia by the National System of Science, Technology, and Innovation – SNCTeI.

The adoption of TRL for evaluating mobile applications can be useful to determine the maturity of an application at a given point in its life cycle. It provides a systematic approach and is widely used in industry and research to evaluate products and technologies. In the context of mobile applications, TRL levels can be used to assess the maturity of the technology used, its degree of innovation and integration with other technologies, the stability of the product, and the responsiveness to user requirements.

However, TRL levels alone do not provide a complete measure of the quality of a mobile application. Other factors such as user experience, accessibility, security, and usability are also critical to evaluate it. Therefore, they should be used in combination with other methods to obtain a more comprehensive assessment of application quality, as proposed in the methodology resulting from the present research.

AUTHORS' CONTRIBUTION

Jorge-Enrique Otalora-Luna: Conceptualization, Investigation, Methodology, Writing-review and editing.

Helver-Augusto Valero-Bustos: Conceptualization, Investigation, Methodology, Writing-review and editing.

Mauro Callejas-Cuervo: Project Administration, Supervision, Writing-review and editing.

REFERENCES

- [1] J. C. Mankins, *Technology readiness levels*, NASA, Washington D. C., 1995.
- [2] J. C. Mankins, "Technology readiness assessments: A retrospective," *Acta Astronautica*, vol. 65, no. 9-10, pp. 1216-1223, 2009. <https://doi.org/10.1016/j.actaastro.2009.03.058>.
- [3] J. Straub, "Evaluating the Use of Technology Readiness Levels (TRLs) for Cybersecurity Systems," in *IEEE International Systems Conference (SysCon)*, Vancouver, BC, Canada, 2021, pp. 1-6. <https://doi.org/10.1109/SysCon48628.2021.9447130>
- [4] S. M. Saad, R. Bahadori, H. Jafarnejad, M. F. Putra, "Smart Production Planning and Control: Technology Readiness Assessment," *Procedia Computer Science*, vol. 180, pp. 618-627, 2021. <https://doi.org/10.1016/j.procs.2021.01.284>
- [5] M. Richardson, M. Gorley, Y. Wang, G. Aiello, G. Pintsuk, E. Gaganidze, M. Richou, J. Henry, R. Vila, M. Rieth, "Technology readiness assessment of materials for DEMO in-vessel applications," *Journal of Nuclear Materials*, vol. 550, e152906, 2021. <https://doi.org/10.1016/j.jnucmat.2021.152906>
- [6] K. B. Kota, S. Shenbagaraj, P. K. Sharma, A. K. Sharma, P. K. Ghodke, W. Chen, "Biomass torrefaction: An overview of process and technology assessment based on global readiness level," *Fuel*, vol. 324, e124663, 2022. <https://doi.org/10.1016/j.fuel.2022.124663>
- [7] R. Ruiz Seva, A. Li Sin Tan, L. M. Sequerra Tejero, M. L. Dorothy S. Salvacion, "Multi-dimensional readiness assessment of medical devices," *Theoretical Issues in Ergonomics Science*, vol. 24, no. 2, pp. 189-205, 2023. <https://doi.org/10.1080/1463922X.2022.2064934>
- [8] G. T. Jesus, M. F. Chagas Junior, "Using Systems Architecture Views to Assess Integration Readiness Levels," *IEEE Transactions on Engineering Management*, vol. 69, no. 6, pp. 3902-3912, 2022. <https://doi.org/10.1109/TEM.2020.3035492>
- [9] F. Martínez-Plumed, E. Gómez, J. Hernández-Orallo, "Futures of artificial intelligence through technology readiness levels," *Telematics and Informatics*, vol. 58, e101525, 2021. <https://doi.org/10.1016/j.tele.2020.101525>
- [10] M. Sarfaraz, B. J. Sauser, E. W. Bauer, "Using System Architecture Maturity Artifacts to Improve Technology Maturity Assessment," *Procedia computer Science*, vol. 8, pp. 165-170, 2012. <https://doi.org/10.1016/j.procs.2012.01.034>
- [11] T. Altunok, T. Cakmak, "A technology readiness levels (TRLs) calculator software for systems engineering and technology management tool," *Advances in Engineering Software*, vol. 41, no. 5, pp. 769-778, 2010. <https://doi.org/10.1016/j.advensoft.2009.12.018>
- [12] J. R. Armstrong, "Applying Technical Readiness Levels to Software: New Thoughts and Examples," in *INCOSE International Symposium*, 2010, pp. 838-845. <https://doi.org/10.1002/j.2334-5837.2010.tb01108.x>
- [13] R. S. Pressman, B. R. Maxim, *Software Engineering: A Practitioner's Approach*, McGraw-Hill, 2014.

- [14] F. F. Tsui, O. Karam, B. Bernal, *Essentials of Software Engineering*, Jones & Bartlett Learning, 2016.
- [15] P. Olivier, X.-H. Ngo, A. Francillon, "BEERR: Bench of Embedded System Experiments for Reproducible Research," in *IEEE European Symposium on Security and Privacy Workshops*, Genoa, Italy, 2022, pp. 332-339. <https://doi.org/10.1109/EuroSPW55150.2022.00040>
- [16] N. Nazar, Y. Hu, H. Jiang, "Summarizing Software Artifacts: A Literature Review," *Journal of Computer Science and Technology*, vol. 31, pp. 883–909, 2016. <https://doi.org/10.1007/s11390-016-1671-1>
- [17] H. Ibrahim, B. H. Far, A. Eberlein, "Scalability improvement in software evaluation methodologies," in *IEEE International Conference on Information Reuse & Integration*, Las Vegas, NV, USA, 2009, pp. 236-241. <https://doi.org/10.1109/IRI.2009.5211557>
- [18] A. Park, M. Wilson, K. Robson, D. Demetis, J. Kietzmann, *Interoperability: Our exciting and terrifying Web3 future*, Business Horizons, 2022. <https://doi.org/10.1016/j.bushor.2022.10.005>
- [19] J. Nielsen, *Usability Engineering*, Morgan Kaufmann, 1994.
- [20] T. Chassin, J. Ingensand, "E-guerrilla 3D participation: Approach, implementation, and usability study," *Frontiers in Virtual Reality*, vol. 3, e1054252, 2022. <https://doi.org/10.3389/frvir.2022.1054252>
- [21] D. P. Simon, *The art of guerilla usability testing*, 2023. <https://www.uxbooth.com/articles/the-art-of-guerrilla-usability-testing/>.