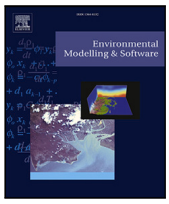
Environmental Modelling and Software 194 (2025) 106692

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: www.elsevier.com/locate/envsoft



# A framework for the acceptance testing of geospatial search engines

Dagoberto José Herrera-Murillo [a,*], Javier Nogueras-Iso [a], Paloma Abad-Power [b], Miguel Á. Latre [a], Francisco J. Lopez-Pellicer [a]

[a] Aragon Institute of Engineering Research (I3A), Universidad de Zaragoza, Spain
[b] Centro Nacional de Informacion Geografica, Spain

ARTICLE INFO

*Keywords:*
Geospatial search engine
User interface
Usability testing
Functional testing
Relevance evaluation

ABSTRACT

Geospatial search engines are an essential component of spatial data infrastructures and enable a broad spectrum of environmental applications. The back-end implementation of these search engines has evolved from traditional text-based information retrieval systems into more specialised search engines. However, to assess the actual improvement brought by this evolution, thorough testing is needed. The aim of this work is to propose a framework for the acceptance testing of geospatial search engines that assesses their functionality, effectiveness, and user-friendliness. For each quality attribute, the framework proposes different testing design techniques and guidelines for their practical implementation. To demonstrate its feasibility, it has been applied to the evaluation of a geospatial semantic search engine of the Spanish National Geographic Institute. The evaluated search engine showed a sufficient level of functionality and effectiveness. However, the usability results were barely satisfactory due to perceived problems associated with complexity, inconsistency, and low learnability.

## 1. Introduction

The advances in geographic information systems (GIS), remote sensing platforms or location-aware devices, among other examples, have motivated the spread on the Web of an enormous volume of geographic information resources in various formats and representations. In order to deal with this volume of data, spatial data infrastructure (SDI) initiatives were launched since the end of the nineties at different administrative levels (regional, national or global) and with the collaboration of both public and private institutions. SDIs can be defined as a cohesive framework encompassing technologies, institutional structures, and policies designed to improve the availability and accessibility of spatial data (Nebert, 2004). They are structured as a hierarchical network of nodes, with key technological components including spatial data, metadata, middleware services (enabling functions such as data location, visualisation, and download), and end-user applications at each node (Martin-Segura et al., 2022). Among these components, metadata, catalogue services and geospatial search engines are instrumental for discovering geographic information resources (Lacasta et al., 2022; Corti et al., 2018). Geospatial resources play a key role in the environmental domain for activities such as ecosystem monitoring, climate analysis, and resource management (Xu et al., 2024; Latre et al., 2013; Wiemann et al., 2016). Many sources of environmental data remain underutilised due to interfaces that require highly specialised technical expertise (Sun et al., 2019). Therefore, it is relevant to connect these sources through SDIs and employ search engines for the discovery of resources.

Traditional geospatial search engines provide a front-end interface where users can enter thematic/spatial keywords, constrain resources on a specific spatial area of interest, and displaying the resources (either directly downloadable datasets or services for visualising and accessing these datasets) in a user-friendly format. However, these search engines are limited in their ability to provide useful results for complex queries as the 'bag-of-words' models used in traditional text retrieval systems cannot palliate the inherent ambiguity and heterogeneity between the language employed in user queries and the terms contained in metadata records. Bone et al. (2016) highlight the importance of enabling explicit spatial search capabilities, moving beyond keyword-based filtering of geographic locations, and advocate the broadening of geospatial data discovery. To address this issue, semantic search engines have been developed on the basis of semantic models such as knowledge graphs to provide a meaningful representation of the annotation of resources in terms of shared and formalised ontologies (Renteria-Agualimpia et al., 2010; Kassim and Rahmany, 2019; Jiang et al., 2017; Janowicz et al., 2019). The experiments conducted by Ferrari et al. (2024) suggest that the semantic augmentation of geospatial search engines has a

* Corresponding author.
*E-mail address:* dherrera@unizar.es (D.J. Herrera-Murillo).

https://doi.org/10.1016/j.envsoft.2025.106692
Received 26 May 2025; Received in revised form 25 August 2025; Accepted 12 September 2025
Available online 18 September 2025

measurable positive effect on the discoverability of relevant geospatial datasets for a given query.

As the goal of semantic search is to improve traditional search methods and provide more useful results for SDI users, it is essential that these software products are thoroughly tested. According to the IEEE Standard Glossary of Software Engineering Terminology (IEEE, 1990), software testing is the process of evaluating a system to verify that it meets specified requirements or to identify any discrepancies between actual and expected results. Testing activities are usually distinguished in different levels according to the party that takes the leading responsibility in each level: on the one hand, the supplying party (the development team) is in charge of the development test level (unit and integration testing) and the system test level in order to assure that the delivered system complies with the expected system requirements, technical specifications and technical design; on the other hand, the accepting party (the contractor) is in charge of the acceptance test level to assure that the received system is the one really expected by contractors, i.e., it meets the user needs and is ready for operational use.

Geospatial search engines demand rigorous testing methodologies that address the complexity of these artifacts in an integrated manner. While existing research has explored individual testing aspects for software products in the geo-information domain, such as functionality or usability (Galimova, 2020; Puspitasari et al., 2023; Kalantari et al., 2021; Horbiński et al., 2021), there is limited work on frameworks specifically tailored for geospatial search engines handling large-scale datasets. This paper addresses this gap by proposing a testing framework for the acceptance testing level of geospatial search engines that combines the principles of the Test Management Approach (TMAP) (Koomen et al., 2007) methodology for the structured software testing process together with domain-specific adaptations and selection of testing tools.

Independently of the design architecture or the semantic model behind search engines, the user interface plays an essential role in the success of a geospatial search engine product and this is typically the core object of analysis in acceptance testing. The proposed framework for acceptance testing aims at evaluating three main quality attributes of geospatial search engines: functionality (level of confidence in the ability of the system to accurately and comprehensively process information), effectiveness (the capacity of the system to deliver a desired output), and user-friendliness (the ease with which end-users use the system). In the case of functionality testing, we propose the application of branch testing and scenario testing as testing design techniques. In the case of evaluating the effectiveness, we propose the measurement of evaluation relevance metrics. Last, in the case of the evaluation of user-friendliness, we propose the application of usability testing techniques considering both the inclusion of users (usability tests) or not (heuristics checking, cognitive walkthroughs).

To demonstrate the feasibility of the proposed framework, it has been applied to the semantic searcher developed as a result of the Linked Cartography project of the Spanish National Geographic Institute (IGN). This project aims to make the cartographic resources of the institution more accessible to the public by means of the development of a Knowledge Graph that integrates various sources in the IGN geospatial data ecosystem (Bucher et al., 2021): the database of datasets in multiple GIS formats available through the IGN Download Centre[1]; the catalogue of the IGN Map Library,[2] which contains historical cartography assets; and the catalogue used as back-end at the Online Shop of the National Centre for Geographic Information for selling IGN products in hardcopy format.[3] Many of these information resources are directly related to natural environmental systems — such as satellite and aerial images, land use data, or digital elevation models — which are of interest to both experts and the general public. The geospatial searcher of the Linked Cartography project employs as back-end a semantic search engine that exploits the knowledge graph populated with the aforementioned sources and contains more than 2 million geographic resources semantically represented in more than 150 million triples. The objective is to make profit from a better annotation of resources to provide users with more precise results accompanied with contextual information and recommendations.

The rest of this paper is structured as follows. Section 2 presents an overview of related work. In Section 3, we delineate the proposed framework adapting TMAP for the acceptance testing of a geospatial search engine. Section 4 describes how to instantiate this framework in a real case study: the newly developed geospatial search engine of the Spanish National Geographic Institute (IGN). Finally, Section 5 offers conclusions and suggests future research directions.

## 2. Related work

This section reviews the state of the art of work related to the role of testing in the development of geographic information software products and the three types of testing relevant to our case study of geospatial search engines: functional testing, effectiveness evaluation and usability testing. In addition, the concept and role of test process methodologies are also presented.

Well-developed testing improves the quality, competitiveness, and demand for geographic information software products (Galimova, 2020; Puspitasari et al., 2023). Galimova (2020) draws a distinction between manual testing, valued for its flexibility and ability to closely replicate user actions, and automated testing, appreciated for its ease of reuse. Through her research, which focused on three classes of geographic information systems (mobile, server, desktop), the author concludes that, for all the GIS classes under consideration, semi-automated testing emerges as the preferred approach.

With respect to **functional testing** for acceptance tests, there is a line of work in software engineering known as *early testing* (Gutiérrez et al., 2006) that aims at facilitating the automation of user acceptance tests based on the definition of requirements. There are several works in this area oriented towards the generation of test cases based on the semi-formal representation of use cases (Escalona et al., 2011), and some tools of increasing use such as Cucumber[4] have popularised the automation of user tests starting from a minimally controlled plain text describing functional requirements of the system under test (Pucciani, 2022).

There are also several approaches for the design of functional test cases derived from the user interface definition. In fact, as one of the few elements on which an agreement is reached between client and developer during the analysis phase of an application are the user interface prototypes and the way in which they will be navigated, the automation tools that try to involve the accepting party in the testing process are based on the use of diagrams that model the behaviour and interaction with the application user. For instance, tools such as Testar (Vos et al., 2021) or NDT-Suite (García-García et al., 2012) are based on this approach.

With respect to the **assessment of the effectiveness**, it must be noted that the most common approach to measure the satisfaction of users in information retrieval systems is to compile metrics related to the evaluation of the relevance in the list of results returned by these systems for a list of information needs under control (Manning et al., 2008). Therefore, since a geospatial search engine can be also considered as an information retrieval system, the evaluation of the relevance should be an appropriate indicator to assess the effectiveness of the system to retrieve relevant results. The effectiveness of a search

[1] https://centrodedescargas.cnig.es/CentroDescargas/home
[2] https://www.ign.es/web/catalogo-cartoteca/
[3] https://www.cnig.es/locale?lang=en
[4] https://cucumber.io/

engine providing a ranked list of results should also take into account the ability of a system to return first those results that are relevant. For that purpose, the 11-point interpolated average precision or Mean Average Precision (MAP) measures are typically computed. However, precision–recall curves or MAP consider the precision at all recall levels and this is quite unfeasible in the case of search engines indexing millions of records: relevance judgments should be available for all the documents in the collection (Zhou et al., 2012). Therefore, many search engines use as evaluation measure the precision at fixed low levels of retrieved results, i.e., *Precision at K* (Sivarajkumar et al., 2024; Ghosh et al., 2022; Cortes et al., 2022).

Regarding the **usability assessment** of search user interfaces, Hearst (2009) provides a comprehensive overview of research on the evaluation of this kind of interface. Search interfaces should be evaluated in terms of efficiency, effectiveness, and satisfaction. In particular, the subjective reaction of participants to the interface is a critical factor in determining the likelihood of use. The evolving nature of interface development requires evaluations with varying levels of complexity and detail. In the early stages of comparing candidate designs, designs are shown to participants, and their responses are recorded to identify areas for improvement. These are often referred to as informal usability studies. Later, formal usability studies through controlled experiments allow us to understand how the target users use the interface and determine whether the design concepts work as expected (Shneiderman and Plaisant, 2004). Finally, for operational interfaces, it is important to conduct studies in which participants use the search platform in their daily routines and environments over a significant period of time (Shneiderman and Plaisant, 2006).

User-unfriendly interfaces and poor GUI design are identified as some of the main problems in the current geospatial software ecosystem (Vandewalle et al., 2021). Related to the collaboration of users in the evaluation of user interfaces, it is worth noting the work of Popelka et al. (2019) and Kalantari et al. (2021). Popelka et al. (2019) employed an eye-tracking method to evaluate the user-friendliness of map-based visual analytics tools, and their conclusions encourage a stronger use of mixed research designs that combine the advantages of quantitative and qualitative methods. Such designs include think-aloud protocols, which provide deeper insights into user reasoning and the causes of errors when interacting with interactive maps. Kalantari et al. (2021) evaluated spatial metadata systems by conducting think-aloud usability testing and semi-structured interviews with users.

Last, it is also worth mentioning **test process methodologies** that could be applied to define a workflow of activities for the testing of geospatial search engines. van Veenendaal (2022) has asserted that the failure to implement test process methodologies is a fundamental factor contributing to system releases falling short of expectations in terms of quality, cost, and timely delivery. The same author declares that while testing theory advocates complete adherence to structured testing as the optimal and most effective solution, in real-world situations, a professional tester often is capable of choosing a minimal set of testing practices from a structured testing approach. He defines this approach as ‘good enough testing’. Its success depends on a clear definition of testing priorities and appropriate risk assessment. Similarly, Vukovic et al. (2018) emphasise the importance of predefining the test process methodology rather than conducting it ad hoc. Keeping this in mind, organisations can choose from existing models or customise one to suit their needs. The same research acknowledges the challenges faced by small and medium-sized companies in formalising their testing procedures, attributed to constraints such as limited time and human resources. There is a recognised necessity to simplify the complexity of available testing models to enhance their feasibility and implementation in such organisations.

When referring to general software test process methodologies, we can cite the Test Management Approach (TMAP) (Koomen et al., 2007). It is a well-known structured process methodology for software testing, that proposes a life cycle model to structure all the activities required for the management, preparation and execution of test processes. Van Banerveld et al. (2016), who employed TMAP to assess the efficacy of a natural language processing tool, acknowledge the distinctive challenge presented by query systems dealing with massive and complex data. The case study carried out by the latter author focuses on the TMAP notion of quality attribute.

With a more specific focus on specific test processes for semantic search engines, it is worth noting the existence of frameworks like the Large-Scale Semantic Evaluation (SEALS) project (Wrigley et al., 2010). The core of the SEALS project is a two-phase process. The automated phase involves collecting non-interactive metrics such as execution success, number of results returned, execution time, and system load. The user-in-the-loop phase requires real users to perform specific search objectives on the search engine. In this process, a number of user-centric metrics are collected, such as the time taken to obtain a successful response and the user impression of the tool through questionnaires. Another test process specifically designed for search engines is the proposal of Zhou et al. (2015). It proposes the application of metamorphic testing as an essential technique to evaluate search engines by comparing relationships between inputs and outputs of different search engines.

The framework proposed in the following section is derived from the TMAP methodology. In addition, it is based on core principles outlined in each of the three types of testing relevant for geospatial search engines: automating test cases based on representative cases from the functional testing domain, *Precision at K* from the assessment of effectiveness and the use of realistic scenarios from usability tests.

It is worth noting that other widely adopted software test process models (Vukovic et al., 2018), such as ISO 29119-2 (ISO/IEC/IEEE, 2021a) and Test Maturity Model Integration (TMMi) (TMMi Foundation, 2022), also provide structured methodologies for software testing. While these frameworks share the common goal of establishing systematic testing processes, the selection of TMAP as the main reference framework for this proposal is due to the alignment of its characteristics with the dynamic nature of geospatial search engines. One of its key advantages is its flexibility, agility, and lightweight structure, making it well-suited for environments where search relevance, indexing mechanisms, and knowledge graph updates are continuously evolving. Additionally, its strong support for automated testing facilitates the evaluation of search UI interactions. Finally, its emphasis on quality attributes aligns with the evaluation needs of modern geospatial search engines.

## 3. Testing framework

In our proposed acceptance testing framework, we have adapted the Test Management Approach (TMAP) to the case of this type of geospatial search engines. For each test level, TMAP delineates a life cycle model for organising the test activities across seven phases: planning (activities are detailed later); control, which encompasses monitoring and readjusting of the planning; setting up and maintaining the infrastructure; preparation of test cases; specification of test scripts; execution of tests and reporting of results; and completion, which consists of the evaluation of the test process and preservation of the testware for future test processes.

Fig. 1 presents an activity diagram showing the life cycle model of an acceptance test level and the logical order of these phases. It can be observed that control and infrastructure phases are transversal activities executed in parallel to the activities devoted to preparation, execution and completion. In addition, it can be observed that the planning phase has been subdivided in various activities and that we have highlighted in blue face the test products that are relevant for our acceptance test process.

During the planning phase, after understanding the mission of an acceptance test level and identifying potential sources for a test basis, we must analyse the product risks to identify and prioritise the test

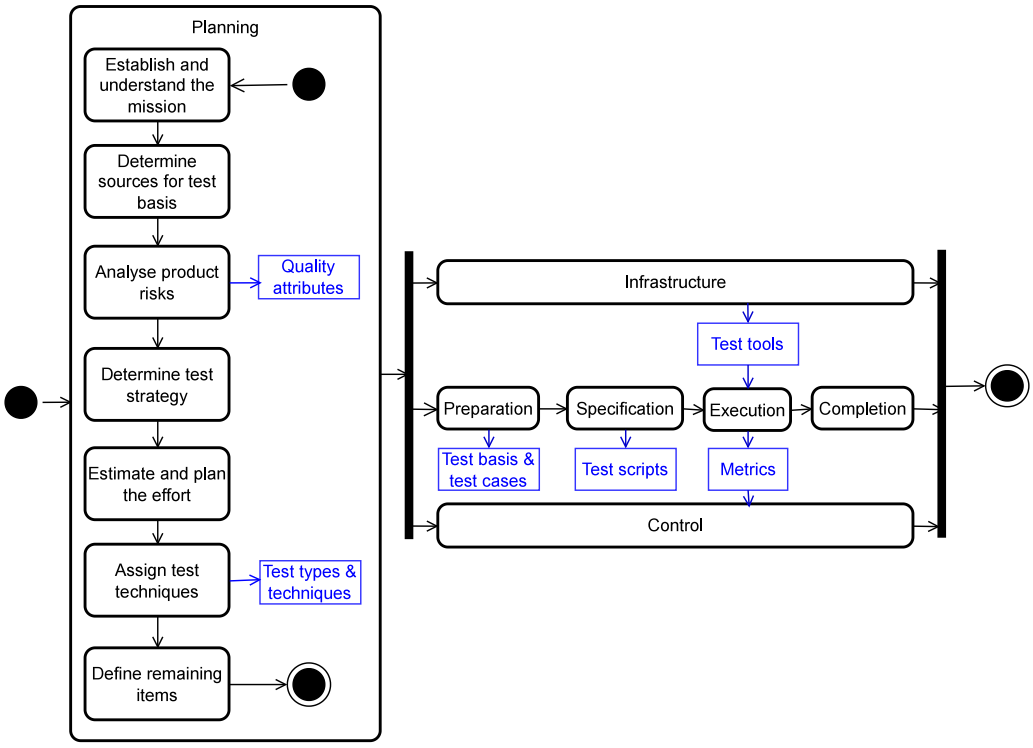



**Fig. 1.** Activity diagram describing the life cycle model of the acceptance test level. Relevant test products generated along the test activities are highlighted in blue face.

objectives, i.e., determine the 'quality attributes' of the system that are critical for a geospatial search engine. Then, we need to determine a test strategy that defines the intensity of testing for every quality attribute. Later, we need to estimate and schedule the required human resource efforts. The following step is devoted to identify the 'test types' and 'techniques' that are more appropriate for designing the test cases associated with each quality attribute. The planning finishes with the definition of other remaining planning details related to the list of deliverables, organisation, infrastructure or management. Outside the planning phase, it must be noted that the preparation phase includes the compilation of a specific 'test basis' (information defining the required behaviour of the system) for the definition of test cases. In addition, the main output of the specification phase are the test scripts: automatic programs or manual procedures for the execution of test cases.

Table 1 summarises the test products that are relevant in our proposed acceptance test process for geospatial search engines. As can be observed, we have identified three main quality attributes that are critical for the development of these geospatial search engines: functionality (level of confidence in the ability of the system to accurately and comprehensively process information), effectiveness (the capacity of the system to deliver a desired output), and user-friendliness (the ease with which end-users use the system). For agreeing on these quality attributes we used as main source the definition of these attributes by TMAP, but they are consistent with other well-known sources for software engineering like the ISO/IEC 25010 standard (ISO/IEC, 2011), which also highlights functionality (the degree to which the system facilitates and cover the realisation of all specified tasks and objectives), effectiveness (the degree to which a system achieves the required level of precision in delivering results) and usability (the degree to which the system facilitates and cover the realisation of all specified tasks and objectives) as main attributes to describe quality in the product or usage model of software and systems.

For each quality attribute, we provide recommendations on the most appropriate test types and test design techniques: functionality testing for assessing the functionality quality attribute; relevance evaluation for assessing the effectiveness of this type of system; and usability to assess user-friendliness. Moreover, each test design technique is aligned with a specific test basis, serving as the main source of information needed for both test case definition and specification of test scripts. Sections 3.1, 3.2 and 3.3 describe how the three different quality attributes have been assessed by selecting the most appropriate test types, test design techniques, and test tools. In addition, each design technique produces a metric for a quantitative assessment of the quality attribute.

### 3.1. Functionality quality attribute

For assessing the functionality quality attribute, we propose the application of two different test design techniques employed in functionality testing: branch testing and scenario testing. The following subsections describe the use of these techniques.

#### 3.1.1. Scenario testing

Scenario testing is a specification-based technique in which testers design cases to evaluate how the software will function when end-users interact with it for specific purposes (ISO/IEC/IEEE, 2021b). This technique involves developing a model of the interaction sequences between the test item and users to test the usage flows that involve the test item (Desikan and Ramesh, 2006).

We employ a form of scenario testing called use case testing (Hass, 2008). This method involves a use case model of the test item that outlines how it interacts with one or more actors. Since use cases are employed to express requirements in the early stages of development, they serve as an excellent basis for acceptance testing. From the description of use cases, we can define features and associated test scenarios using the Gherkin language (Pucciani, 2022). This language allows to specify the expected behaviours of the software in a human-readable way: it is a minimally controlled language containing just plain text and a reduced set of reserved words. We decided to use this language

**Table 1**
Test strategy for the acceptance test level.

| Quality attribute | Test type | Testing design technique | | Test basis | Test tools | Metrics |
|---|---|---|---|---|---|---|
| Functionality | Functional | Scenario testing | | Use cases | Behave + Selenium | Number of failing scenarios |
| | | Branch testing | | Navigation map (UI workflow) | Behave + Selenium | Number of failing paths |
| Effectiveness | Relevance evaluation | Ranking evaluation measures | | Relevance evaluation benchmark | Ranking evaluation program | precision@10 |
| User-friendliness | Usability | Without users | Heuristic evaluation | User interface | Manual inspection | Number of violations |
| | | | Cognitive walkthrough | Tasks proposed by experts | Behave + Selenium | Number of results + Execution time |
| | | With users | Usability test | Usability test scenario | Sesion recordings + questionnaire | SUS score |

because it is designed to write acceptance tests that can be implemented using Cucumber, a platform widely used for functional testing and behaviour driven development. The ultimate goal of Gherkin is to facilitate the understanding of software testing or behaviour by technical and non-technical team members.

With respect to the infrastructure needed for the automation of these tests, we propose the use of Behave (Rice et al., 2023), a Python implementation of Cucumber that converts Gherkin scenarios into Python test scripts. In addition, to interact with the web interface of a geospatial search engine, we propose the integration of Selenium (García et al., 2020). Selenium is an open-source automation testing framework for web-based applications, which offers a Selenium WebDriver that allows interaction with most modern web browsers. That is to say, test scripts written in Python (and also other programming languages) can integrate a specialised library to interact with the Selenium WebDriver to trigger different User Interface (UI) events (e.g. open/close web pages, typing text or mouse events) in an automated way.

#### 3.1.2. Branch testing

Branch testing is a structure-based technique that evaluates the system by following possible logical branches in its functional flow (ISO/IEC/IEEE, 2022).

We conducted branch testing considering the decision points and flow restrictions outlined in Fig. 4. In particular, we derived test cases by applying the test depth level N technique (Koomen et al., 2007). According to this technique, achieving test depth at a certain level $N$ implies that all the combinations of $N$ consecutive branches are covered. For instance, test depth level 1 is equivalent to achieve full branch coverage as stated in part 4 of the ISO 29119 standard (ISO/IEC/IEEE, 2021b). With test depth level 2, all combinations of branches going in and out of each decision point are covered, or, equivalently, all subpaths of two consecutive branches starting at each decision point. Test depth level 3 covers all subpaths of three consecutive branches starting at each decision point and so on.

With respect to the infrastructure for the automation of these test cases, we propose to express these paths as Gherkin scenarios because Gherkin allows expressing a sequence of actions in an almost human-readable way. In addition, the translation of these steps into interactions with the geospatial search engine can be implemented in the same way as proposed for scenario testing, i.e. using Behave and Selenium WebDriver.

### 3.2. Effectiveness quality attribute

As already introduced in Section 2, the effectiveness of search engines and information retrieval systems is usually assessed in terms of the evaluation of the relevance in the list of results returned by these systems.

To measure the performance of an information retrieval system in terms of relevance evaluation, we need a relevance evaluation benchmark, also known as test collection, comprising three components: a document collection; queries expressing information needs; and a set of relevance judgments (usually a binary assessment) for each query-document pair. The most common measures for information retrieval effectiveness without taking into account the ranking of results are precision (the proportion of retrieved documents that are considered relevant), recall (the proportion of relevant documents retrieved) and the F-measure (a weighted harmonic mean of the previous measures). However, the effectiveness of a search engine providing a ranked list of results should also take into account the ability of a system to return first those results that are relevant. For that purpose, the 11-point interpolated average precision or Mean Average Precision (MAP) measures are typically computed: on the one hand, the 11-point interpolated average precision is a precision–recall curve consisting of the average interpolated precision of the considered information needs at 11 fixed recall points; on the other hand, MAP computes the arithmetic mean of the average precision of each considered information need, which averages the precisions whenever a relevant document is retrieved.

The problem of precision–recall curves or MAP is that they consider the precision at all recall levels. However, this is quite unfeasible in the case of search engines indexing millions of records: despite using a small test suite of information needs for relevance evaluation, it is not possible to have relevance judgments for all the documents in the collection. On the other hand, this may not be relevant to final users of search engines because they are usually interested only in the first page of results. Therefore, for the purpose of relevance evaluation in the proposed framework of this work, we propose to measure the precision at fixed low levels of retrieved results. This is referred to *Precision at K* or *Precision@k*, i.e., the precision computed when the top $k$ documents are retrieved. This measure is widely used for the evaluation of web search engines and offers sufficient conditions for acceptance testing, where testers do not have direct access to the full list of resources ranked by relevance.

### 3.3. User-friendliness quality attribute

In the case of the evaluation of the user-friendliness quality attribute, we propose the application of usability testing techniques considering both the inclusion of users (usability tests) or not (heuristics checking, cognitive walkthroughs). The customisation of these techniques for the case of geospatial search engines is explained in the following subsections.

#### 3.3.1. Tests without users: Heuristics checking

Heuristic evaluation entails expert raters applying a usability checklist to a user interface in order to spot potential usability issues that could prevent users from carrying out their intended tasks (Nielsen,

1993). In a classic heuristic evaluation, the user interface is checked to verify whether specific design criteria are followed to enhance the user experience.

In particular, we propose the use of the ten heuristics established by Nielsen (1992) for evaluating user interfaces: visibility of system status (1); match between system and the real world (2); user control and freedom (3); consistency and standards (4); error prevention (5); recognition rather than recall (6); flexibility and efficiency of use (7); aesthetic and minimalist design (8); help users recognise, diagnose, and recover from errors (9); and help and documentation (10). An expert evaluator executes typical browser search tasks, annotates violations in a standardised worksheet, and assigns severity ratings to the violated heuristics.

#### 3.3.2. Tests without users: Cognitive walkthroughs

Cognitive walkthrough is a usability evaluation technique that connects the interface review to a cognitive model (Mahatody et al., 2010). The evaluator simulates the experience of a typical user by performing tasks on the interface. The process compares the user expectations with the actual steps required by the interface to complete the tasks.

The accepting party must collaborate in the identification of specific examples of information needs required by new users of the geospatial search engine. After defining these information needs, it is necessary to think about the actions that should be performed in the search engine to accomplish these information needs. Then, these information needs are expressed as a sequence of steps in a Gherkin scenario. On the one hand, Gherkin facilitates a well-known language to express these actions. On the other hand, these actions can be automated in the same way as proposed for scenario testing and branch testing.

Finally, in order to identify potential deviation of the automated executions of Gherkin scenarios from the expected behaviour, we propose to record the execution time each scenario and the number of results that were returned.

#### 3.3.3. Tests with users: Usability tests

In a usability test, a researcher asks a participant to perform representative tasks using one or more specific user interfaces. During the task completion, the researcher observes the participant behaviour and listens for feedback (Moran, 2019). The primary objectives of usability tests are to identify any issues with the system design and gain insight into the behaviour and preferences of our target users.

For the testing sessions we propose the use of the 'think-aloud strategy' (Elbedweihy et al., 2015), a well-established method for gathering data in user studies where participants are asked to vocalise their thoughts while performing tasks. This provides insight into the user reasoning, perception, and difficulties with the search tasks through the interface.

To understand the patterns of perception and use by novice and specialised users, the recruitment of participants must take into account individual differences in search performance, as studied in the literature (Hearst, 2009). Factors such as knowledge of the task domain, experience as searchers, and cognitive differences are considered to define three types of participants: I. Non-experts, II. Non-familiar experts, III. Familiar experts. The difference between unfamiliar and familiar expert users is that the latter regularly use the platforms and products of the institution whose search engine is going to be tested.

Finally, participants must complete the System Usability Scale (SUS) questionnaire, a widely used survey (Brooke, 1996), after completing the task. This questionnaire consists of 10 items rated on a 5-point Likert scale, and the final score ranges from 0 to 100.

## 4. Case study: the evaluation of the new IGN search engine

This section illustrates how the proposed framework can instantiated in a real case study such as the geospatial search engined developed by IGN. Fig. 2 shows the main web page of the search engine interface. This interface allows final users to have access to different functionalities related to the search process as illustrated in the use case diagram of Fig. 3. In addition, in order to describe better the overall search process enabled by the geospatial search engine, Fig. 4 provides an activity diagram of the search workflow. The process begins with the selection of a search method, either textual or explicitly geospatial. The search results are then displayed and can be refined using faceted filters or various types of views. Finally, the user can perform actions such as view, download, purchase, and locate on a specific resource.

The following subsections provide practical guidelines for the implementation of the testing design techniques in our proposed framework. In some cases, such as scenario testing, branch testing or cognitive walkthroughs, test scripts can be fully automated. For the rest of the techniques, we provide full details for the preparation of test cases and their manual execution. In all cases, we describe the obtained results after the execution of tests. There is also a code repository[5] with the implemented scripts for automated tests and some Python notebooks for the analysis of results.

### 4.1. Functional testing results

#### 4.1.1. Scenario testing

The use cases employed for our testing were provided by the search engine development team, who had already specified and justified them in detail at the beginning of the project in the technical documentation. The resulting 7 use cases are the ones already shown in Fig. 3: Search, Display results, Filter results, View metadata, Locate resource, Download resource and Buy resource.

The implementation of the Gherkin scenarios associated to the 7 features (use cases) are provided in the code repository (see *scenario_testing.feature* and the implementation of steps in these scenarios). The scenarios associated to the search feature (shown in Table 2) are the most complex because they include a variety of inputs. During the execution of the test cases, we did not encounter any incidents. Based on our testing results, we can confirm that the search engine of the platform satisfies the functional requirements for which it was designed. In addition, the implementation and execution of these test cases helped us to make the implementation of the Gherkin steps as much generic as possible. This was important because the implementation of test cases derived from the application of branch testing and cognitive walkthrough testing design techniques were also expressed in Gherkin language.

#### 4.1.2. Branch testing

Fig. 5 presents a simplified version of the diagram shown in Fig. 4, that we use to derive the test situations according to the test depth level $N$ technique. The only nodes the graph depicts, besides the initial and final ones, correspond to the decisions present in Fig. 4, while the edges are the complete branches or paths that go from a decision or starting node to another decision or final node in the original diagram. For instance, edge labelled '2' represents the complete branch where the user selects 'free search' in the first decision node, types free text, performs the search and gets the results displayed, ending in the decision node where the user must choose among different types of filters.

In our case, test depth level 2 provides a total of 46 test situations (that is, 46 pairs of consecutive branches in the graph), like '1-2' or '2-9'. These 46 test situations can be exercised in 24 execution paths or test

[5] https://github.com/IAAA-Lab/Acceptance-testing-of-geospatial-semantic-search-engines-ODECO-CNIG

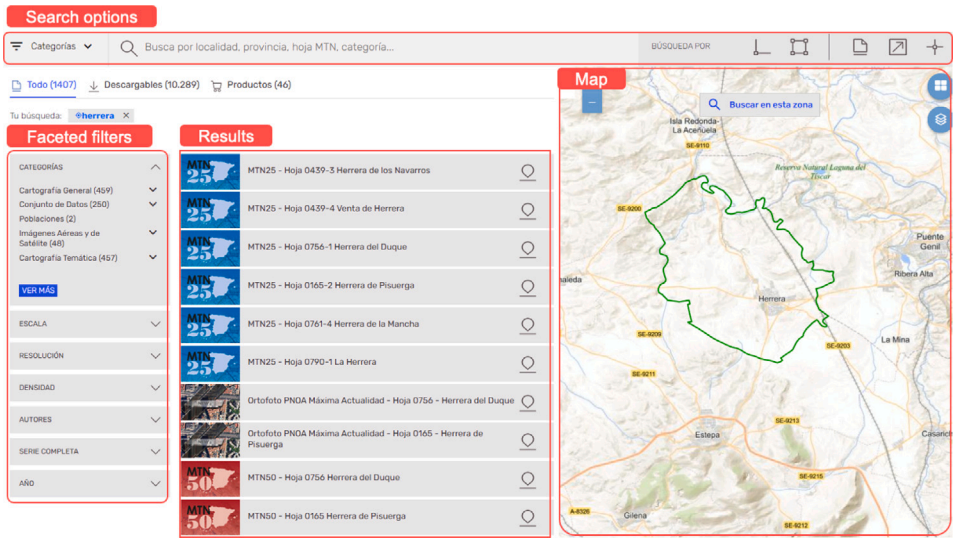



**Fig. 2.** Geospatial search engine interface.

**Table 2**
Test scenarios for the search feature written in Gherkin.

| Test scenario | Status |
|---|---|
| **Scenario:** the user is able to search for resources typing text<br>**Given** the user is on the home page of the search engine<br>**When** the user performs a textual search for *"Madrid"*<br>**Then** search results are displayed | Passed |
| **Scenario:** the user is able to search for resources selecting a point on the map<br>**Given** the user is on the home page of the search engine<br>**When** the user searches for a point in the centre of the map<br>**Then** search results are displayed | Passed |
| **Scenario:** the user is able to search for resources drawing a geometry on the map<br>**Given** the user is on the home page of the search engine<br>**When** the user searches for a geometry in the centre of the map<br>**Then** search results are displayed | Passed |
| **Scenario:** the user is able to search for resources uploading a geometry file<br>**Given** the user is on the home page of the search engine<br>**When** the user loads the file *"BTT0101_vivar_del_cid-burgos.gpx"*<br>**Then** search results are displayed | Passed |
| **Scenario:** the user is able to search for resources typing a set of coordinates<br>**Given** the user is on the home page of the search engine<br>**When** the user types coordinate *"3.40" "40.30"*<br>**Then** search results are displayed | Passed |
| **Scenario:** the user is able search for resources typing a cadastral reference<br>**Given** the user is on the home page of the search engine<br>**When** the user enters the cadastral reference *"9977715VK3797F"*<br>**Then** search results are displayed | Passed |

cases. For instance, '1-3-4-9-11-17' is one of these paths and represents a test case where the user starts the search, chooses an explicit spatial search, selects a point in the map, performs the search, gets results displayed, applies a faceted filter, filters the results by products, buys and is displayed payment options.

In order to make the testing more exhaustive, we wanted to make sure that all the possible combinations of the six different types of search with the four possibilities of filtering the results and the five possible actions that can be performed with the results were also tested. Given our graph, the later can also be achieved with the test depth technique, switching to level 3. There are a total of 134 test situations in this case, that can be covered with 90 paths. In order to decrease the number of test cases, while testing all the aforementioned combinations of types of search, filters and actions, we designed again the test cases to cover all the level 2 test situations with those level 3 test situations that do not include the branch labelled with '9', that loops back to the filtering section. In this last case, we obtained a combined total of 99 test situations.

The obtained test cases cover 66% of the test depth level 3 test situations, 100% of the level 2 test situations and also 100% of the branch test situations according to part 4 of the ISO 29119 standard (ISO/IEC/IEEE, 2021b), while keeping the number of test cases in a reasonable level.

To cover the selected 99 test situations we executed a total of 66 paths. The description in Gherkin of these paths is available at *branch_testing.feature* file of the code repository, together with its associated Python implementation.

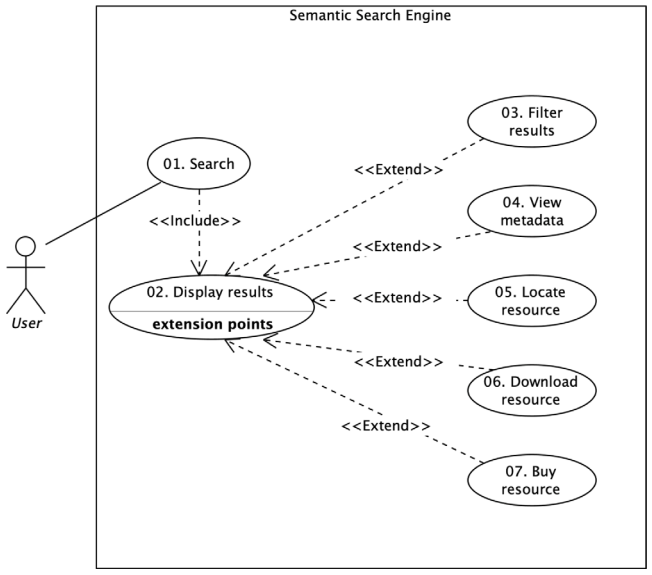



**Fig. 3.** Use case diagram illustrating all the functionalities related to the search process.

About the results of the execution of Gherkin scenarios, during the first iteration we found that 54 (82%) of the paths executed without any problems. The remaining 12 tests failed, all of which are associated with search mechanisms based on the selection of points or polygons on the side map. The instructions to interact with Selenium WebDriver were programmed to perform the selection in the centre of the map, which by default shows the entire Iberian Peninsula with a portion of the ocean, and whose centre falls in Portugal. We discovered that the execution failure occurred because the search engine did not have any resources indexed for that area in the download or purchase category. Therefore, any successive operations to view, locate, download or purchase resources simply could not be executed. To fix this issue, we moved the selection zone to Spain, which allowed us to confirm that the filters were working correctly. This exercise enabled us to confirm that users would eventually be able to perform the search sequences without encountering errors or crashes.

### 4.2. Relevance evaluation results

According to the proposed methodology, we calculated the *Precision@10* of the geospatial search engine with some information needs. But taking into account the existence of other search engines at IGN, we also compared the measure with the one obtained by the three current geospatial search engines existing at IGN for discovering separately resources on the Download Centre, the Map Library and the Online Shop.

Previous to the computation of *Precision@10*, we had to prepare the evaluation benchmark. For that purpose, we proposed first five information needs, which are shown *information need* column in Table 3. Second, we compiled the first ten results returned by the four search engines with the search terms associated with these five information needs. Third, we had to annotate the relevancy of all the results with respect to the information needs. This task was performed by three experts (the judges). Finally, those results having a majority of relevant votes were considered relevant for the computation of *Precision@10*. In addition, to assess the agreement between the judges, we computed the Fleiss' kappa measure, which is an extension of Cohen's kappa measure to evaluate the agreement between two or more judges (Latha, 2017). The results indicated a substantial agreement among the judges with $\kappa = .70$.

Table 3 shows the *Precision@10* measure obtained by each search engine. Overall, the geospatial search engine outperformed its counterparts in both average precision and individual searches, with the exception of one case. The Library Map had a decent performance, but its relevant results were limited to historical cartography. In contrast, the Download Centre and the Online Shop had notably poor performance, with some queries yielding no relevant results.

### 4.3. Usability testing results

#### 4.3.1. Tests without users: Heuristics checking

Usability testing without users started with the verification of the Nielsen heuristics. Table 4 contains the list of violations associated with each of the ten heuristics and the assigned severity. Violations were categorised into low, medium and high severity. Low severity violations are those that may slow down search tasks, but would not necessarily prevent the user from finding the necessary information. Medium-severity violations are those that may create significant obstacles in search tasks, which may cause confusion or delay, but do not completely prevent the user from finding or correctly interpreting the necessary information. High-severity violations are those that may prevent the user from finding or correctly interpreting useful resources that are actually contained in the system.

Upon reviewing the detected violations, we identified two distinct categories. The first category comprises violations related to search and navigation structures, such as the search bar, filters, and icons. The second category is associated with the quality of content. In addition to these specific violations, we also observed transversal deficiencies in navigation and content, such as the systematic use of jargon. Of all the heuristics used to detect violations, the one that targets the internal and external consistency of the search engine reported the highest number of violations.

#### 4.3.2. Tests without users: Cognitive walkthroughs

As indicated in Section 3.3.2, the experts of the accepting party proposed five user tasks (searching of information needs) for applying the technique of cognitive walkthroughs. The 'Test case' column of Table 5 shows the sequence of actions for these five tasks as Gherkin scenarios. The implementation of these Gherkin scenarios is provided in the code repository, see *cognitive_walkthrough.feature* and the implementation of steps in these scenarios.

The columns *Passed*, *# results* and *Ex. Time* in Table 5 show the results after the execution of the test scripts. All tests met the defined criteria within execution times of less than one minute. When reviewing manually the execution of the walkthroughs contained in the test scripts, no deviations from the expected results were observed, and the results were found to be consistent with the search criteria.

#### 4.3.3. Tests with users: Usability tests

For the usability test, we selected a search task that represented the projected use of the platform and was easily understood by novice and expert users - planning a trip to a popular tourist destination in Spain:

'As you plan your visit to the Sierra Nevada National Park, some information about the area is required. Use the search engine to find information, download files, or add products to the cart that could be useful for your trip'.

The usability test adhered to the Ethics Appraisal Procedure established by the European Union Horizon 2020 Programme.[6] This adherence included procedures to enlist appropriate participants and an informed consent protocol. The demographics of the recruited participants for the usability test are shown in Table 6. Overall, there are no significant disparities in the composition of gender, age, and education among the three groups of participants. At the start of each session, a

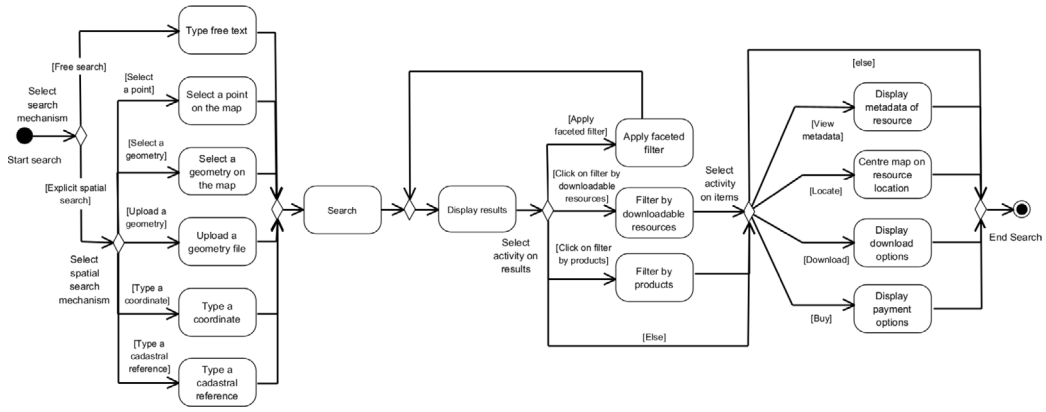



**Fig. 4.** Activity diagram illustrating the search workflow.

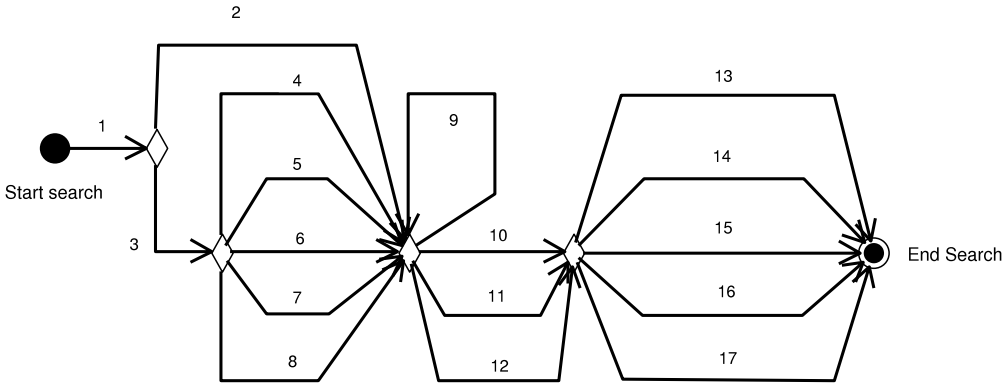



**Fig. 5.** Schematic graph with complete branches as single edges.

**Table 3**
Precision@10.

| Information need | Semantic Search Engine | Map Library | Online Shop | Download Centre* |
|---|---|---|---|---|
| Discover cartographic resources of the autonomous community of **"Asturias"** | 1.0 | 1.0 | 0.5 | 0.1 |
| Download a trail file related to the search for the **"Way of El Cid"** | 0.4 | 0.0 | 0.0 | 0.0 |
| Buy the current map of the city of **"Toledo"** | 1.0 | 0.8 | 0.0 | 0.4 |
| Discover general cartographic resources of the region of **"Murcia"** | 1.0 | 1.0 | 0.4 | 0.0 |
| View the area of the **"Sierra Nevada"** National Park on the side map | 0.4 | 0.9 | 0.6 | 0.5 |
| Average precision@10 | 0.8 | 0.7 | 0.3 | 0.2 |

*It is not strictly a free-text search mechanism. The user types in the search and must necessarily choose one of the suggested terms and results are not displayed in one single list.

pre-test questionnaire was administered to confirm the classification of participants into their respective user categories.

Due to security restrictions, the version of the browser used for testing could not be accessed by external users, so the exercise was designed in such a way that the user had to verbally indicate their actions to the moderator. During the tests, recordings were made and the moderator of the sessions took notes of the user feedback. This content was then reviewed by the accepting party, who identified three major areas for improvement that were consistently mentioned by users: (a) the role of the side map needs to be rethought to make it a truly interactive visualiser that is closely linked to the search results, (b) the faceted filters and categories need to be redesigned to make their meaning and operation more intuitive, and (c) a better guidance must be provided explaining how to use the retrieved geographic resources. Expert users also provided specific suggestions about how to improve the information architecture of the search engine by providing examples from other platforms and previous experiences. Novice user comments were generally more limited in detail and expressiveness. These comments were also passed on to the accepting party for consideration.

[6] https://ec.europa.eu/research/participants/docs/h2020-funding-guide/cross-cutting-issues/ethics_en.htm

**Table 4**
Nielsen heuristics.

| # | Heuristic | Notes | Severity | | |
|---|---|---|---|---|---|
| | | | Low | Medium | High |
| 1 | Visibility of System Status | Updates of the shopping cart or download area is not noticeable | X | | |
| 2 | Match between the System and the Real World | The search engine contains a wealth of jargon unknown to the general public and even to experts outside the institute | | | X |
| 3 | User Control and Freedom | The user can remove previous searches, but cannot edit them directly to refine them | X | | |
| 4 | Consistency and Standards | Different applications of the same institute handle different icons for the same functions | | X | |
| | | The descriptions of some of the resources do not correspond appropriately to the files or products deployed | | | X |
| | | Sometimes the result list does not intuitively represent the demarcated area on the side map | | | X |
| | | The number of resources shown by the filters does not correspond to the result list | | X | |
| | | Sometimes the result list does not intuitively represent the demarcated area on the side map | X | | |
| 5 | Error Prevention | Some search mechanisms that require the input of parameters, such as coordinates, lack default values to help the user identify the expected format | | X | |
| | | Some of the resources point to empty metadata records | X | | |
| 6 | Recognitition Rather Than Recall | Most of the icons lack tooltips | | X | |
| 7 | Flexibility and Efficiency of Use | Several search mechanisms cannot be combined efficiently and even combining them worsens the result | | | X |
| 8 | Aesthetic and Minimalist Design | The thematic category filters have dozens of options that are displayed simultaneously | X | | |
| 9 | Recognise, Diagnose and Recover from Errors | No suggestions are shown when searches are unsuccessful or when resources are empty | X | | |
| | | The error messages are not informative | X | | |
| 10 | Help and Documentation | The files lack sufficient description to know their contents without first downloading them | | X | |

The System Usability Scale (SUS) applied at the end of each of the 30 usability tests reports a value of $\alpha = .85$ for Cronbach's alpha, which means that the reliability of the questionnaire is high. Table 7 shows the results for item scores and overall SUS scores. A Kruskall–Wallis test was used to determine if there were significant differences between testing groups. Only items 7 and 8 showed significant differences. Post-hoc tests using the Bonferroni correction showed that non-exerts and familiar experts are different for item 7, while non-familiar experts and familiar experts are different for item 8 at statistical significance at level $\alpha = .05$.

While no statistically significant differences were found in overall SUS scores, Fig. 6 presents box plots for each testing group and situates them within the adjective ratings and acceptability ranges proposed by Bangor et al. (2009). This complementary system helps to clarify how numerical scores correspond to qualitative judgements of usability: scores below 50 fall in the 'NOT ACCEPTABLE' range and are typically described as 'Poor' or even 'Worst Imaginable'; scores between 50 and 70 correspond to a 'MARGINAL' range (often labelled 'OK') that reflects usability which is just passable but not yet satisfactory; and scores above 70 are generally regarded as 'ACCEPTABLE', aligning with adjectives such as 'Good', 'Excellent', or 'Best Imaginable'. The overall SUS mean of 67.5 is positioned at the upper edge of the marginal range (barely acceptable, but approaching the boundary of what is considered 'Good').

## 5. Conclusions

This paper has presented a framework for the acceptance testing of geospatial search engines to assess their functionality, effectiveness, and user-friendliness. For each quality attribute the framework proposes the applicability of different testing design techniques and provides guidelines for their practical implementation taking into account that the search engine provides a web-based interface. Whenever possible, test scripts implementing test cases (functional tests and cognitive walkthroughs) have been automated. In addition, to put this framework into practice, we evaluated a new semantic search engine developed for discovering resources at the Spanish National Geographic Institute.

While this study does not include a direct empirical comparison of our framework with other testing frameworks, we discuss the similarities and differences at a conceptual level to highlight its contributions. The acceptance testing methodology behind our framework is derived from TMAP. This distinguishes our approach from the reviewed related work that tests geographic information software products without explicitly elucidating a test process framework (Galimova, 2020;

**Table 5**
Cognitive walkthroughs written in Gherkin.

| Test case | Status | # results* | Ex. Time |
|---|---|---|---|
| **Scenario**: Discover cartographic resources of the autonomous community of *"Asturias"* | | | |
| **Given** the user is on the home page of the search engine | | | |
| **When** the user performs a textual search for *"Asturias"* | *Passed* | *T: 12,359* *D: 112,308* *P: 141* | *39.433s* |
| **Then** resources related to *"Asturias"* are displayed in the *"All"* view | | | |
| **When** the user selects one of the available resources | | | |
| **Then** a full metadata record describing the resource is displayed in a new tab | | | |
| **Scenario**: Download a trail file related to the search for the *"Way of El Cid"* | | | |
| **Given** the user is on the home page of the search engine | | | |
| **When** the user performs a textual search for *"Way of El Cid"* | | *T: 862* | |
| **Then** resources related to *"Way of El Cid"* are displayed in the *"All"* view | *Passed* | *D: 12,165* *P: 35* | *50.610s* |
| **When** the user selects one of the available resources | | | |
| **Then** a full metadata sheet describing the resource is displayed in a new tab | | | |
| **When** the user downloads one of the files available in the metadata record | | | |
| **Then** the file is downloaded locally | | | |
| **Scenario**: Buy the current map of the city of *"Toledo"* | | | |
| **Given** the user is on the home page of the search engine | | | |
| **When** the user performs a textual search for *"Toledo"* | | *T: 21,720* | |
| **Then** resources related to *"Toledo"* are displayed in the *"All"* view | *Passed* | *D: 116,708* *P: 192* | *48.223s* |
| **When** the user selects one of the available resources | | | |
| **Then** a full metadata record describing the resource is displayed in a new tab | | | |
| **When** the user buys one of the available resources in the metadata record | | | |
| **Then** the selected product is added to the shopping cart | | | |
| **Scenario**: Discover general cartographic resources of the region of *"Murcia"* | | *T: 17,911* | |
| **Given** the user is on the home page of the search engine | | *D: 109,854* | |
| **When** the user performs a textual search for *"Murcia"* | | *P: 139* | |
| **Then** resources related to *"Murcia"* are displayed in the *"All"* view | *Passed* | *After filtering* | *44.140s* |
| **When** the user selects the filter of *"General Cartography"* | | *T: 456* | |
| **Then** only the resources related to *"General Cartography"* are displayed | | *D: 3,133* | |
| | | *P: 136* | |
| **Scenario**: View the area of the *"Sierra Nevada"* National Park on the side map | | | |
| **Given** the user is on the home page of the search engine | | *T: 1,759* | |
| **When** the user performs a textual search for *"Sierra Nevada"* | *Passed* | *D: 22,611* | *38.939s* |
| **Then** resources related to *"Sierra Nevada"* are displayed in the *"All"* view | | *P: 117* | |
| **When** the user locates one of the available resources | | | |
| **Then** the location of the resource is shown in the side map | | | |

*T (Total results), D (Downloads), P (Products)

**Table 6**
Demographics of study participants (%).

| | All | I. Non-experts | II. Non-familiar experts | III. Familiar experts |
|---|---|---|---|---|
| **Gender** | | | | |
| Male | 17 *(57%)* | 5 *(50%)* | 6 *(60%)* | 6 *(60%)* |
| Female | 13 *(43%)* | 5 *(50%)* | 4 *(40%)* | 4 *(40%)* |
| **Age** | | | | |
| 18-24 | 2 *(7%)* | 1 *(10%)* | 1 *(10%)* | - *(0%)* |
| 25-34 | 6 *(20%)* | 2 *(20%)* | 2 *(20%)* | 2 *(20%)* |
| 35-44 | 6 *(20%)* | - *(0%)* | 1 *(10%)* | 5 *(50%)* |
| 45-54 | 13 *(43%)* | 5 *(50%)* | 5 *(50%)* | 3 *(30%)* |
| 54-65 | 2 *(7%)* | 2 *(20%)* | - *(0%)* | - *(0%)* |
| +65 | 1 *(3%)* | - *(0%)* | 1 *(10%)* | - *(0%)* |
| **Education** | | | | |
| High School | 1 *(3%)* | 1 *(10%)* | - *(0%)* | - *(0%)* |
| Graduate | 18 *(60%)* | 6 *(60%)* | 5 *(50%)* | 7 *(70%)* |
| Postgraduate | 11 *(37%)* | 3 *(30%)* | 5 *(50%)* | 3 *(30%)* |
| **Total** | 30 | 10 | 10 | 10 |

Puspitasari et al., 2023). Our effort could facilitate its potential for future adoption among practitioners engaged in testing products with similar characteristics, providing a more systematic and comprehensive view of the testing process as authors such as van Veenendaal (2022) or Vukovic et al. (2018) suggest. As in the case study developed by Van Banerveld et al. (2016), the determination of quality attributes that occurs in the TMAP product risk analysis was of great importance in the acceptance test of the geospatial search engine because clear quality attributes help to maximise the impact of scarce testing resources. Therefore, the active involvement of the accepting party in the prioritisation of these attributes should receive a great deal of attention from the testing team.

Our framework also shares several similarities with the SEALS approach (Wrigley et al., 2010), particularly in the use of automatic and user-in-the-loop elements. However, while SEALS primarily focuses on identifying metrics, our approach emphasises the integration of various test design techniques. In addition, our proposal aligns with think-aloud protocols identified in the summary of major methods for evaluating user interfaces by Hearst (2009). The same think-aloud approach is reflected in other evaluations of interfaces in the geospatial domain Kalantari et al. (2021). However, our proposal places a greater emphasis on visualising the geospatial search engine in the context of a software development project. We believe that this approach can

**Table 7**
Median System Usability Scale (SUS) scores for each item and testing group.

| SUS items | All | I | II | III | p |
|---|---|---|---|---|---|
| | Median score contribution (0-4) | | | | |
| 1. I think that I would like to use this system frequently | 3 | 3 | 3 | 3 | *0.55* |
| 2. I found the system unnecessarily complex | 2 | 3 | 2 | 1.5 | *0.06* |
| 3. I thought the system was easy to use | 3 | 3 | 3.5 | 3 | *0.19* |
| 4. I think that I would need the support of a technical person to be able to use this system | 3 | 3 | 4 | 3 | *0.07* |
| 5. I found the various functions in this system were well integrated | 3 | 3 | 3 | 2 | *0.27* |
| 6. I thought there was too much inconsistency in this system | 2 | 2 | 2.5 | 2.5 | *0.67* |
| 7. I would imagine that most people would learn to use this system very quickly | 2 | 3 | 2.5 | 1 | *0.01** |
| 8. I found the system very cumbersome to use | 3 | 3 | 3.5 | 2 | *0.04** |
| 9. I felt very confident using the system | 3 | 3 | 3 | 2 | *0.51* |
| 10. I needed to learn a lot of things before I could get going with this system | 3 | 3 | 4 | 3 | *0.25* |
| SUS Score (0-100) | 67.5 | 70 | 75 | 58.8 | *0.17* |

I. Non-experts, II. Non-familiar experts, III. Expert Familiar Users
* indicates the statistical significance at level a=.05

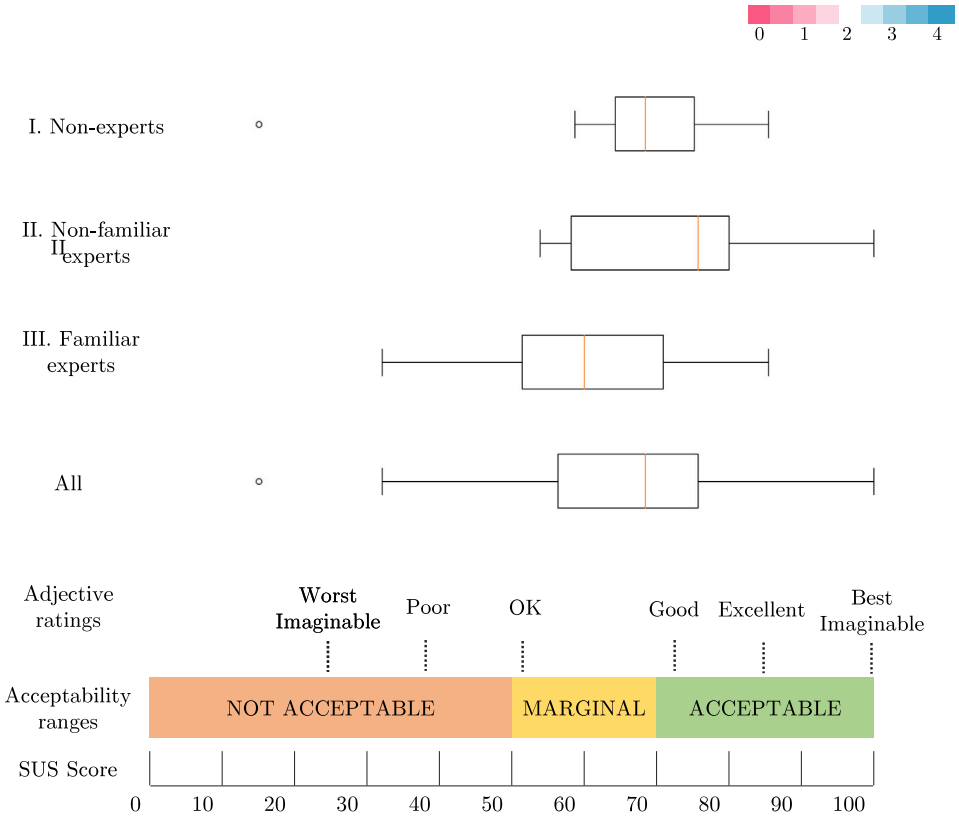



**Fig. 6.** Box plots of System Usability Scale (SUS) scores for each testing group.

facilitate its adoption by integrating well-known software engineering testing techniques such as branch testing, scenario testing, and cognitive walkthroughs.

A key consideration for the future development of this framework is its adaptability to different organisational contexts. From a scalability perspective, larger implementations would benefit from automated testing pipelines capable of managing high volumes of queries, benchmarking performance across heterogeneous datasets, and ensuring reproducible results under diverse semantic configurations. One example would be its application to cross-border infrastructures involving large-scale geospatial data. At the same time, the adoption by smaller organisations may be constrained by limited technical capacity, financial resources, and maintenance capabilities. To address these challenges, the framework could be provided with simplified test suites that minimise the need for specialist expertise while still ensuring essential coverage. In addition, modular testing procedures would allow organisations to deploy only those components most relevant to their operational context.

Regarding the results of the geospatial search engine case, the platform performed well in terms of functionality, fulfilling all specifications for exploring geographic resources through various search mechanisms. The relevance evaluation suggests that the new search engine offers an improvement over the base platforms belonging to the institute. However, usability testing revealed areas for improvement, particularly in visual representation on the map, the structure of filters, and guidance for using the resources found. The overall usability rating in tests with users was barely satisfactory, with no significant differences between novice and expert users in the total score or items on the System Usability Scale, except for those items referring to the cumbersomeness of the system and the opinion on whether most people would learn to use this system very quickly. Novice users were more likely to believe that most people would learn to use the search engine quickly compared to expert users.

We identified several potential threats to the validity of our framework and implemented measures to mitigate their effects. First, the limited size of samples used for relevance evaluation (particularly Precision@k) and usability testing can constrain the robustness and generalisability of the results. This underscores the importance of carefully selecting evaluation tasks so that they can reflect the diversity of information needs encountered in real-world use, with the same care applied to choosing representative participants for testing. Second, the constraints introduced by the moderation method based on the verbalisation of actions (which, in this case, had to be employed due to security restrictions) may have introduced a degree of artificiality that limited the validity of some results. In particular, requiring participants to articulate their actions could have slowed task performance, constrained spontaneous exploration, and disrupted the natural flow of interaction. Nevertheless, it is important to emphasise that the framework itself remains independent of such contextual constraints. This limitation is best understood as an example of the kinds of adaptations that testing teams may encounter when applying the framework in their own contexts. At the same time, this unforeseen restriction, arguably an extreme variant of the think-aloud technique, may also have yielded indirect benefits. By focusing attention on verbalising actions, users may have made more explicit some of the advantages typically associated with thinking aloud, such as the expression and capture of intentions, reasoning, and decision-making processes. Lastly, the reliance on a single geospatial search engine developed by IGN represents a limitation in terms of generalisability. Whilst this focus enabled a detailed, end-to-end demonstration of feasibility, it does not in itself confirm applicability across a wider range of systems. Nevertheless, the framework was deliberately grounded in widely adopted and domain-independent testing principles such as the TMAP life-cycle model, Precision@k for relevance evaluation, and the System Usability Scale (SUS) for usability. These principles strengthen its potential portability to other geospatial platforms.

Across all domains, the availability of case studies that provide end-to-end evaluations of new user interfaces, from test conceptualisation to the deployment of results, can greatly assist practitioners in guiding product development (Faizrakhmanov et al., 2025). Compared to other alternatives for testing geospatial search engines, our proposal integrates well-known software engineering testing techniques in a structured approach that could facilitate its potential future adoption. In addition, the development of complex geographic software products with high human input requires a deep understanding of the dynamics of user interaction.

Future work will focus on extending the application of the framework to multiple platforms, including both institutional SDI catalogues and community-driven open-source platforms, in order to confirm its adaptability and robustness across diverse technical and organisational contexts. Moreover, the observed differences in the behaviour of expert and novice users, particularly in relation to feedback expressiveness, suggest that further research is needed to clarify the origins of these differences (whether they stem from cognitive overload, task complexity, or unfamiliarity with geospatial systems), their impact on user experience, and their implications for system design. One promising avenue to pursue this is through the analysis of differences in the mental models of interaction among users with varying levels of experience (Herrera-Murillo et al., 2023). Additional lines of research include the automatic translation of Gherkin test cases into scripts that interact with the web application (Köroğlu and Sen, 2020), the development of field testing of geographic information search systems, which will help to compensate for the limitations of think-aloud testing by capturing user interaction under more naturalistic conditions, and the investigation of how the geographic products retrieved through the system are subsequently used in real-time decision-making.

## CRediT authorship contribution statement

**Dagoberto José Herrera-Murillo:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Javier Nogueras-Iso:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Paloma Abad-Power:** Writing – review & editing, Writing – original draft, Supervision, Resources, Methodology, Investigation, Formal analysis, Conceptualization. **Miguel Á. Latre:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Francisco J. Lopez-Pellicer:** Writing – review & editing, Writing – original draft, Supervision, Resources, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

## Software and data availability

- Name of software: Scripts for acceptance testing of geospatial semantic search engines
- Developers: Dagoberto José Herrera-Murillo, Miguel Á. Latre, Francisco J. Lopez-Pellicer, Javier Nogueras-Iso
- Contact: dherrera@unizar.es
- Date first available: March 11, 2024
- Software required: Python 3.8 or higher, Jupyter Notebook or a compatible IDE (e.g., VS Code, PyCharm), and the Python libraries listed in the repository.
- Program language: Python
- Source code at: https://github.com/IAAA-Lab/Acceptance-testing-of-geospatial-semantic-search-engines-ODECO-CNIG
- Documentation: This repository contains some python scripts for performing automatic acceptance tests for a geospatial semantic search engine and some python notebooks for the analysis of the data collected in the usability and performance evaluation tests. The folders "Relevance_evaluation" and "SUS_scores" each contain analysis notebooks along with the corresponding datasets. In the "Relevance_evaluation" folder, the "Relevance_evaluation.csv" dataset includes relevance scores assigned by three evaluators to search results obtained from the geospatial search engine and other related platforms. In the "SUS_scores" folder, the "SUS.csv" dataset contains usability scores provided by participants based on their responses to the System Usability Scale (SUS) questionnaire during the usability testing phase.

## Declaration of competing interest

The authors declare that they have not known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This paper is partially supported by the Aragon Regional Government through the project T59_23R. The work of Dagoberto José Herrera-Murillo is supported by the ODECO project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 955569.

## Data availability

We have shared the link to a repository in our manuscript.

## References


Bangor, A., Kortum, P., Miller, J., 2009. Determining what individual SUS scores mean: Adding an adjective rating scale. J. Usability Stud. 4 (3), 114–123.

Bone, C., Ager, A., Bunzel, K., Tierney, L., 2016. A geospatial search engine for discovering multi-format geospatial data across the web. Int. J. Digit. Earth 9 (1), 47–62.

Brooke, J., 1996. Usability Evaluation in Industry. Taylor and Francis, pp. 189–194.

Bucher, B., Folmer, E., Brennan, R., Beek, W., Hbeich, E., Würriehausen, F., Rowland, L., Maturana, R.A., Alvarado, E., R., B., Di Donato, P. (Eds.), 2021. Spatial Linked Data in Europe: Report from Spatial Linked Data Sessions at Knowledge Graph in Action. Official Publication - EuroSDR, pp. 17–18, http://www.eurosdr.net/sites/default/files/uploaded_files/eurosdr_publication_ndeg_73.pdf.

Cortes, E.G., Woloszyn, V., Barone, D., Möller, S., Vieira, R., 2022. A systematic review of question answering systems for non-factoid questions. J. Intell. Inf. Syst. 58 (3), 453–480.

Corti, P., Kralidis, A., Lewis, P., 2018. Enhancing discovery in spatial data infrastructures using a search engine. PeerJ. Comput. Sci. 4.

Desikan, S., Ramesh, G., 2006. Software Testing. Pearson Education India.

Elbedweihy, K., Wrigley, S., Clough, P., Ciravegna, F., 2015. An overview of semantic search evaluation initiatives. J. Web Semant. 30, 82–105.

Escalona, M.J., Gutierrez, J.J., Mejías, M., Aragón, G., Ramos, I., Torres, J., Domínguez, F.J., 2011. An overview on test generation from functional requirements. J. Syst. Softw. 84 (8), 1379–1393.

Faizrakhmanov, R., Bahrami, M.R., Platunov, A., 2025. Prototype, method, and experiment for evaluating usability of smart home user interfaces. Comput. Stand. Interfaces 92, 103903.

Ferrari, E., Striewski, F., Tiefenbacher, F., Bereuter, P., Oesch, D., Di Donato, P., 2024. Search engine for open geospatial consortium web services improving discoverability through natural language processing-based processing and ranking. ISPRS Int. J. Geo-Information 13 (4).

Galimova, E., 2020. Features of software testing in the development of geographic information systems. In: E3S Web of Conferences, vol. 177, EDP Sciences, 02008.

García, B., Gallego, M., Gortázar, F., Munoz-Organero, M., 2020. A survey of the selenium ecosystem. Electronics 9 (7).

García-García, J.A., Ortega, M.A., García-Borgoñon, L., Escalona, M.J., 2012. NDT-Suite: A model-based suite for the application of NDT. In: 12th International Conference on Web Engineering. In: Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 469–472, URL https://link.springer.com/chapter/10.1007/978-3-642-31753-8_46.

Ghosh, S., Razniewski, S., Weikum, G., 2022. Answering count queries with explanatory evidence. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2415–2419.

Gutiérrez, J.J., Escalona, M.J., Mejías, M., Torres, J., 2006. Generation of test cases from functional requirements. A survey. In: Proceedings 4rd Workshop on System Testing and Validation. pp. 1–10.

Hass, A., 2008. Guide to Advanced Software Testing. Artech House.

Hearst, M., 2009. Search User Interfaces. Cambridge University Press.

Herrera-Murillo, D.J., Nogueras-Iso, J., Abad-Power, P., Lopez-Pellicer, F.J., 2023. User interaction mining: Discovering the gap between the conceptual model of a geospatial search engine and its corresponding user mental model. In: Perspectives in Business Informatics Research. Springer, pp. 3–15.

Horbiński, T., Cybulski, P., Medyńska-Gulij, B., 2021. Web map effectiveness in the responsive context of the graphical user interface. ISPRS Int. J. Geo-Information 10 (3), 134.

IEEE, 1990. IEEE 610.12-1990—IEEE standard glossary of software engineering terminology. https://standards.ieee.org/standard/610_12-1990.html.

ISO/IEC, 2011. ISO/IEC 25010:2011 systems and software engineering — systems and software quality requirements and evaluation (SQuaRE) — System and software quality models. https://www.iso.org/standard/35733.html.

ISO/IEC/IEEE, 2021a. ISO/IEC/IEEE 29119-2:2021 software and systems engineering — Software testing — Part 2: Test processes. https://www.iso.org/standard/79430.html.

ISO/IEC/IEEE, 2021b. ISO/IEC/IEEE 29119-4:2021 software and systems engineering — Software testing — Part 4: Test techniques. https://www.iso.org/standard/79430.html.

ISO/IEC/IEEE, 2022. ISO/IEC/IEEE 29119-1:2022 software and systems engineering - software testing - Part 1: General concepts. https://www.iso.org/standard/83636.html.

Janowicz, K., Gao, S., McKenzie, G., Hu, Y., Bhaduri, B., 2019. GeoAI: spatially explicit artificial intelligence techniques for geographic knowledge discovery and beyond. Int. J. Geogr. Inf. Sci. 34 (4), 625–636.

Jiang, Y., Li, Y., Yang, C., Liu, K., Armstrong, E.M., Huang, T., Moroni, D.F., Finch, C.J., 2017. A comprehensive methodology for discovering semantic relationships among geospatial vocabularies using oceanographic data discovery as an example. Int. J. Geogr. Inf. Sci. 31 (11), 2310–2328.

Kalantari, M., Syahrudin, S., Rajabifard, A., Hubbard, H., 2021. Synchronising spatial metadata records and interfaces to improve the usability of metadata systems. ISPRS Int. J. Geo-Information 10 (6), 393.

Kassim, J., Rahmany, M., 2019. Introduction to semantic search engine. In: International Conference on Electrical Engineering and Informatics 2009.

Koomen, T., van der Aalst, L., Broekman, B., Vroon, M., 2007. TMap® Next for Result-Driven Testing, second ed. UTN Publishers, Willem van Oranjelaan 5 5211 CN 's-Hertogenbosch The Netherlands, p. 752, URL http://www.tmap.net/en/tmap-next.

Köroğlu, Y., Sen, A., 2020. Functional test generation from UI test scenarios using reinforcement learning for android applications. Softw. Test. Verif. Reliab. 31.

Lacasta, J., Lopez-Pellicer, F.J., Zarazaga-Soria, J., Béjar, R., Nogueras-Iso, J., 2022. Approaches for the clustering of geographic metadata and the automatic detection of quasi-spatial dataset series. ISPRS Int. J. Geo-Information 11 (2), 87.

Latha, K., 2017. Experiment and Evaluation in Information Retrieval Models. Chapman and Hall/CRC.

Latre, M., Lopez-Pellicer, F.J., Nogueras-Iso, J., Béjar, R., Zarazaga-Soria, F.J., Muro-Medrano, P.R., 2013. Spatial Data Infrastructures for environmental e-government services: The case of water abstractions authorisations. Environ. Model. Softw. 48, 81–92.

Mahatody, T., Sagar, M., Kolski, C., 2010. State of the art on the cognitive walkthrough method, its variants and evolutions. Int. J. Human–Computer Interact. 26 (8), 741–785.

Manning, C.D., Raghavan, P., Schtze, H., 2008. Relevance feedback and query expansion. In: Introduction to Information Retrieval. Cambridge University Press, New York.

Martin-Segura, S., Lopez-Pellicer, F.J., Nogueras-Iso, J., Lacasta, J., Zarazaga-Soria, F.J., 2022. The Problem of Reference Rot in Spatial Metadata Catalogues. ISPRS Int. J. Geo-Information 11 (1), 27.

Moran, K., 2019. Usability testing 101. https://www.nngroup.com/articles/usability-testing-101/.

Nebert, D. (Ed.), 2004. Developing spatial data infrastructures: The SDI cookbook. Global Spatial Data Infrastructure (GSDI), URL http://gsdiassociation.org/images/publications/cookbooks/SDI_Cookbook_GSDI_2004_ver2.pdf.

Nielsen, J., 1992. Finding usability problems through heuristic evaluation. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 373–380.

Nielsen, J., 1993. Usability Engineering. Academic Press Professional.

Popelka, S., Herman, L., Řezník, T., Pařilová, M., Jedlička, K., Bouchal, J., Kepka, M., Charvát, K., 2019. User evaluation of map-based visual analytic tools. ISPRS Int. J. Geo-Information 8 (8).

Pucciani, G., 2022. Boozang from the Trenches: Learn Test Automation with Boozang in an Enterprise Environment. Springer, pp. 217–224.

Puspitasari, T.D., Kurniasari, A.A., Puspitasari, P.S.D., 2023. Analysis and testing using boundary value analysis methods for geographic information system. IOP Conf. Ser.: Earth Environ. Sci. 1168 (1).

Renteria-Agualimpia, W., López-Pellicer, F.J., Muro-Medrano, P.R., Nogueras-Iso, J., Zarazaga-Soria, F.J., 2010. Exploring the advances in semantic search engines. In: Distributed Computing and Artificial Intelligence: 7th International Symposium. Springer, pp. 613–620.

Rice, B., Jones, R., Engel, J., 2023. Welcome to behave!. https://behave.readthedocs.io/en/latest/.

Shneiderman, B., Plaisant, C., 2004. Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison Wesley.

Shneiderman, B., Plaisant, C., 2006. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In: Proceedings of the 2006 conference Advanced Visual Interfaces. AVI'04.

Sivarajkumar, S., Mohammad, H., Oniani, D., Roberts, K., Hersh, W., Liu, H., He, D., Visweswaran, S., Wang, Y., 2024. Clinical information retrieval: A literature review. J. Heal. Informatics Res..

Sun, Z., Di, L., Gaigalas, J., 2019. SUIS: Simplify the use of geospatial web services in environmental modelling. Environ. Model. Softw. 119, 228–241.

TMMi Foundation, 2022. Test Maturity Model integration(TMMi). Guidelines for Test Process Improvement. Release 1.3. https://www.tmmi.org/tmmi-documents/.

Van Banerveld, M., Kechadi, M.T., Le-Khac, N.A., 2016. A natural language processing tool for white collar crime investigation. In: Hameurlain, A., Küng, J., Wagner, R., Dang, T.K., Thoai, N. (Eds.), Transactions on Large-Scale Data- and Knowledge-Centered Systems XXIII: Selected Papers from FDSE 2014. Springer, pp. 1–22.

van Veenendaal, E.P., 2022. Building on success – beyond the obvious: A closer look at good enough testing. In: Proceedings of the Federated Africa and Middle East Conference on Software Engineering. Association for Computing Machinery, pp. 91–92.

Vandewalle, R., Barley, W., Padmanabhan, A., Katz, D., Wang, S., 2021. Understanding the multifaceted geospatial software ecosystem: a survey approach. Int. J. Geogr. Inf. Sci. 35 (11), 2168–2186.

Vos, T., Aho, P., Pastor Ricós, F., Rodriguez-Valdes, O., Mulders, A., 2021. Testar – scriptless testing through graphical user interface. Softw. Test. Verif. Reliab. 31.

Vukovic, V., Djurkovic, J., Trninic, J., 2018. A business software testing process-based model design. Int. J. Softw. Eng. Knowl. Eng. 28 (05), 701–749.

Wiemann, S., Brauner, J., Karrasch, P., Henzen, D., Bernard, L., 2016. Design and prototype of an interoperable online air quality information system. Environ. Model. Softw. 79, 354–366.

Wrigley, S., Reinhard, D., Elbedweihy, K., Bernstein, A., Ciravegna, F., 2010. Methodology and campaign design for the evaluation of semantic search tools. In: Proceedings of the Semantic Search 2010 Workshop.

Xu, K., Chen, M., Yue, S., Zhang, F., Wang, J., Wen, Y., Lü, G., 2024. The portal of OpenGMS: Bridging the contributors and users of geographic simulation resources. Environ. Model. Softw. 180.

Zhou, Z.Q., Xiang, S., Chen, T., 2015. Metamorphic testing for software quality assessment: A study of search engines. IEEE Trans. Softw. Eng. 42 (3), 264–284.

Zhou, Z.Q., Zhang, S., Hagenbuchner, M., Tse, T., Kuo, F.C., Chen, T., 2012. Automated functional testing of online search services. Softw. Test. Verif. Reliab. 22, 221–243.