A Survey of Heuristic Evaluation Tools

Martin Rabensteiner

706.414 Seminar/Project Interactive and Visual Information Systems 4SP WS 2025/2026
Graz University of Technology

16 Jan 2026

Abstract

User interfaces on several devices influence or everyday life. Usability inspection and testing methods help developers to improve their products and the usability. The heuristic evaluation is a wide spread and often used approach to conduct such usability inspections.

This survey gives a brief overview about the process and main elements of a heuristic evaluation and the tools to support it. The tools are either manual, install, or online tools to support such studies. Further, an outlook is given about a tool to implement and which specifications it can fulfil to combine advantages of examined tools.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Nowadays, a life without digital interfaces is hardly imaginable. The usage and popularity of an application is can be improved through several usability techniques. Nielsen [1993] pioneered usability engineering and described several methods and guidelines already decades ago. These tools and processes are still important in modern computer science and application development. Often *User Experience* (UX) or *User Interaction* (UI) are used as comprehensive terms.

1.1 Usability Inspection Methods

Usability inspection methods are used to review user interfaces in a deterministic way. [Nielsen and Mack 1994; Wilson 2013] The methods are non-experimental, but based on analysis and judgement[Andrews 2025]. Thus, experts are needed to conduct such methods. The experts can be experts of usability as well as domain, or both. Examples for usability inspection methods are the *Guideline Checking*, where reviewers compare the application with hundreds of guidelines. A variant of this is the *Guideline Scoring*, where the guidelines are added up to a total score at the end. In the *Cognitive Walkthrough*, a team of usability experts go through a task simulating a novice user and reconstruct a story of success or failure.

In contrast to usability *inspection* methods, there are also usability *testing* methods. [Andrews 2025] These do not require usability experts, but ordinary users. An example is the *thinking aloud test*, where tasks are given to users, they have to execute them and comment what they are seeing and thinking. Meanwhile, they are watched and recorded by the facilitator. Another popular inspection method is the *A/B testing*, where real live users are randomly split into two groups, with two different interfaces, and response differences are observed.

1.2 Heuristic Evaluation

Heuristic evaluation is a usability inspection method, where a small number of usability experts review a user interface and use their judgement and experience to find potential problems. Wilson [2013] refers to it as *perhaps the best-known inspection method*. The evaluation is guided by a small set of broad principles (the *heuristics*) for good user interface design. Outside academic circles, the technique is often known as an *expert review*.

Molich and Nielsen [1990] derived an initial set of nine principles, which were then used to familiarise the evaluators in three of the four (Experiments 1, 3, and 4) founding heuristic evaluation experiments [Nielsen and Molich 1990]. In 1993, Jakob Nielsen added a tenth heuristic (Help and Documentation) [Nielsen 1993, Chapter 5]. The ten heuristics are:

1. *Visibility of System Status*: The user knows what the system is currently doing and receives feedback, for example where he currently is or whether the system is loading.

2. *Match Between the System and the Real World*: The system follows natural mappings, terms, and concepts that the user is already familiar with.

3. *User Control and Freedom*: Users are able to keep control and undo mistakes or exit current situations.

4. *Consistency and Standards*: The system follows industry standards and consistency within its ecosystem.

5. *Error Prevention*: When there is an error, the user receives a meaningful and helpful hint. The system is also built to avoid mistakes.

6. *Recognition Rather than Recall*: Information is visible where it is needed (recognize) instead of forcing the user to remember (recall) it from before.

7. *Flexibility and Efficiency of Use*: The system supports the user with shortcuts and customization possibilities.

8. *Aesthetic and Minimalist Design*: Irrelevant information is missed out.

9. *Help Users Recognise, Diagnose, and Recover from Errors*: Error messages are composed in an understandable way and help to user to make it better.

10. *Help and Documentation*: If really necessary, the system provides further information, documentation and help right where and when it is required.

For over three decades, these ten heuristics have remained the fundamental set of generic principles behind heuristic evaluation. The latest in 2024, Nielsen updated the descriptions of the ten heuristics and provided convenient one-page posters for each of them with examples [Nielsen 2024].

Beside these, some domain, or application specific heuristics arose as well. Hermawati and Lawson [2016] conducted a literature review study about different heuristics and observed 70 sets. These are domain specific for websites, computer games, medical devices, robotic applications, and so on. Most of the studies used the set of Nielsen as a basis and adapted it to their needs. In a more recent literature review, Galavi et al. [2024] observed 17 studies in the field of mobile health applications. Most of them used the Nielsen heuristics as well, or a clear link to them could be made. Beyond that, 10 further heuristics were identified among studies. Yáñez Gómez et al. [2014] proposed 13 heuristics for mobile interfaces. They are also mostly based on Nielsen's heuristics and consider additionally other thoughts of similar literature. Quiñones and Rusu [2017] described a process and a template for developing specific heuristics in seven steps.

The ideal process of a heuristic evaluation looks as follows. The five steps in particular are described by Andrews [2025] and contain the following phases:

1. *Preparation*: The interface to be evaluated is available. The fidelity, or stage of implementation, whether it is a mock-up or already a running system, is less important. The checklist of heuristics is set up. If necessary, the evaluators can be trained in the specific domain of the application.

2. *Individual Evaluations*: Each evaluator does the evaluation on its own. This lasts approximately one to two hours. The evaluation is recorded, for example with a screen recording software, or with when camcorder when evaluating paper mock-ups. A list of all positive and negative findings is made. When possible, the findings are recreated in short video clips with more context. In some case, a screenshot can already be sufficient to describe a finding.

3. *Aggregation*: The evaluation manager summarises the findings of all evaluators in one list. Typically, the longest evaluator's list is taken and others then merged into it. Similar findings are merged into one, with link to the individual findings. Also small, related problems are taken together.

4. *Severity Ratings*: Each evaluator reviews the merged findings list and rates each severity. The individual ratings are averaged, and then the list sorted with decreasing severity.

5. *Debriefing and Report*: Possible solutions are presented. A report concludes all findings and further recommendations.

In comparison to the formal, automatic, and empirical evaluation, the heuristic evaluation claims to have a simpler set of rules, less complexity, calling to the intuition of usability experts, and though find a high percentage of user interface faults. It is important to mark, that to reach this high percentage, the evaluation has to be done with multiple users.

Nielsen and Moloch found a significant difference between different types of users. *Novice users* are those having no experience in usability, *regular specialists* with knowledge in usability, and *double experts* with knowledge in usability as well as domain knowledge. A single novice user finds approximately 20 % of known usability problems, a regular specialist 40 % and a double specialist 60 %. With five users of each, the amount of found issues rises to approximately 45 %, 90 %, respectively 98 % per type. They name an ideal number of about three to five evaluators to encounter most usability problems. In this range, the quantity of problems found rises the most. With more evaluators it flattens again. [Nielsen 1994]

To measure the heaviness of issues, Nielsen [1994] introduced and described the severity of usability problems. The severity is composed of three factors:

- *Frequency*: How frequent a problem occurs.
- *Impact*: How hard it is to overcome the problem.
- *Persistence*: Whether to problem occurs once and is then overcome, or whether it occurs again and again.

Additionally, the *market impact* influences how urgent the problem has to be solved. Although one might have a low severity, it can have a strong impact on the popularity of an application. Usually, a severity rating from 0 (no usability problem) to 4 (*usability catastrophe*) is sufficient.

To reach good severity estimates, it is necessary to get the evaluations and ratings of multiple evaluators individually and aggregate them. It is recommended to not ask for the severity during the evaluation to get. A questionnaire can be sent out after the evaluation to get better estimations and also to consider the findings of other evaluators. Some tools calculate a severity on its own. Evaluation issues should not be discussed among the evaluators before the rating, to achieve better comparable results. In the case study it turns out, that the severity rating of an evaluator does not depend on whether it is an own issue or found by someone else. The reliability of a severity estimation rises with the number of evaluators and their experience (see above).

Chapter 2

A Survey of Heuristic Evaluation Tools

Since the first description of heuristic evaluation by Rolf Molich and Jakob Nielsen in 1990 [Nielsen and Molich 1990], many tools have been introduced to support the technique in practice. Different tools follow different approaches, from locally installable software to web applications and browser extensions. Some tools are kept simple to just track issues, others have user management, merging options, and integrations for other tools used in common workflows. Some were only mentioned once in the form of a thesis or a paper and not further maintained, others are distributed as commercial solutions and are still available. This survey gives an overview of historic and current tools to support heuristic evaluations.

2.1 Terminology

Some concepts and wording can differ between the tools. To maintain consistent terminology in this survey, the following definitions will be used:

- *Evaluation*: Sometimes also called (expert) review, the inspection of user interface.
- *Tool*: An application (app), program, or process which supports the conducting of a heuristic evaluation.
- *Project*: An instance of a heuristic evaluation study, typically involving multiple expert evaluators and a project manager.
- *Evaluator*: Sometimes also called reviewer, a person conducting an evaluation, typically an expert in usability.
- *Project Manager*: The person managing a heuristic evaluation project: assigning evaluators, aggregating results, and producing a final report.
- *Finding*: A single issue or remark derived from an evalutor. There can be both positive findings and negative findings (problems). Similar or equivalent findings by multiple evaluators are often merged into one.
- *Heuristic*: A single, general principle of good usability.
- *Set of Heuristics*: A bundle of heuristics, sometimes generic, sometimes focused on a particular domain.

2.2 Structure of the Survey

This survey is structured into three chapters, one for each of three types of tool:

- *Manual Tools* (Chapter 3): As manual tools, some more procedures are collected to conduct a heuristic evaluation by hand, either on paper or digital. That means, results can be digital, but are not processed programmatically.
- *Installable Tools* (Chapter 4): Installable tools are applications that need to be locally installed on a computer. Although some of them need a browser because they are extensions, they count as installable tools since they need to be installed on the computer. They do not communicate with any server by themself and also work in an offline environment.
- *Online Tools* (Chapter 5): Online tools are accessible with a customary web browser like Mozilla Firefox or Google Chrome. They are either dependent on external servers ("cloud services") or can also be hosted local or on a private web server.

Every tool is explained through a textual description, where the properties, the process, and the user interface are described. Additional to the textual and tabular overview of the tools, showcase videos are made for those tools available and running. For screenshots and showcase videos, the website of Graz University of Technology [TU Graz 2025] is used when possible.

2.3 Comparison Criteria

To better compare the different tools, their characteristics are described according to a predefined set of criteria.

The criteria for the manual tools are:

- *Availability*: How information about the tool could be gathered and which resources are available.
- *Owner*: The authors of the code, or paper, or the company distributing the tool.
- *Launch date*: The year of publication/launch.
- *Last update*: The year the last update happened to the tool (if so).
- *Licence*: The licence of the tool, whether it is open-source or commercial.
- *Underlying Technology*: Some tools are constructed to be completed by pen, others need a spreadsheet software like Microsoft Excel or Google Docs.

The online tools and installable tools have the following 27 criteria:

- *Name*: The name under which the tool is published or advertised.
- *Availability*: How information about the tool could be gathered. This includes the following ways:
  - *Code*: The code is available, for example within in public GitHub or browser extension repository.
  - *Online service*: The tool is running on a server somewhere on the internet.
  - *Article*: The tool was described in the context of a scientific paper and this paper is available.
  - *Bachelor's/Master's thesis*: the tool was described either in the context of a Bachelor's or a Master's thesis and this thesis is available.
- *Owner*: The authors of the code, or paper, or the company distributing the tool.
- *Launch date*: The year of publication/launch.
- *Last update*: The year the last update happened to the tool (if so).
- *Licence*: The licence of the tool, whether it is open-source or commercial.

- *Hosted and/or self-hosted*: Whether the tool is tied to a server of the owner, or can also be self-hosted. Only applicable to online tools,
- *Technology stack*: The underlying software stack.
- *Dark and/or light mode*: Whether the tool adjusts to the system light/dark mode setting, or only one mode is available.
- *Responsiveness*: Whether the tool adjusts to the screen size, respectively it is suitable for mobile devices.
- *Support for multiple reviewers*: The possibility to combine evaluations of multiple evaluators, for example with merging.
- *Evaluator management*: It is possible to assign evaluators to findings, set a due date, or monitor the review progress.
- *Fixed and/or custom sets of heuristics*: The set of heuristics is given, can be selected from multiple sets or a custom set can be defined by the evaluation manager.
- *Distinguish between positive and negative findings*: It is possible to mark a finding
- *Attach a description to a finding*: The possibility to add a description for the finding.
- *Attach a screenshot*: The possibility to attach a screenshot to a finding.
- *Attach a video clip to a finding*: The possibility to attach a video to a finding.
- *Support for guided questionnaires/checklists*: The possibility or the obligation to go through a checklist of heuristics or guidelines.
- *Report export formats*: Possible formats to export the report (PDF, DOC, PPT, HTML, etc.)
- *Finding export formats*: Possible formats to export the findings in a structured format (CSV, JSON, etc.)
- *Collaboration, messaging*: How evaluators can interact with each other within the tool.
- *Documentation*: How the tool is documented and what further resources are available.
- *Integrations*: The support to automatically interact with other tools, for example the direct export to GitHub.
- *Customisable branding*: The possibility to adjust logos, colours, etc. to the own corporate branding.
- *Other limitations*: Other parameters that limit evaluations with this tool.
- *Additional features*: Some features that let the tool shine out.

Chapter 3

Manual Tools

Manual tools can be seen as the oldest method of documenting heuristic evaluation. The simple noting on sheets of paper came along with Jakob Nielsen's defintion of *discount usability engineering* [Nielsen 1994]. This chapter covers different approaches to document heuristic evaluation findings manually.

3.1 Heuristic Evaluation Workbook [2023-]

[Moran and Gordon 2023]

3.2 Axis

[Axis 2017b] [Axis 2017a]

3.3 Axure

Ritch Macefield [2014]

3.4 Extended Structured Report Format

Cockton et al. [2004]

3.5 Morae

Travis [2007] [TechSmith 2004] [TechSmith 2019]

Chapter 4

Installable Tools

This section covers the tools that need to be installed manually and run locally on the system. Some tools covered here are browser extensions. These only work in a specific browser and can therefore be used only for the evaluation of websites.

At this point, it should be noted once again that some online tools, for example Heurio (Section 5.11 or Capian (Section 5.10, also have installable software needed or available. However, they are still connected to a server. Tools listed here are completely stand-alone.

4.1 Review Assistant

The Review Assistant in version 1.0 was launched in 2002 by the company *Information & Design* and is a Microsoft Windows application. The contributors are Gerry Gaffney, Daniel Szuc and Andrew Rowsell [Information & Design 2006; Loitzl 2006, pages 50–56]. A version of the executable is preserved in the Web Archive [IaD 2004]. The source code is closed. The full version was lastly available for $19.99 and had in contrast to the free trial version the possibility to export reports.

Adding findings is done in the issues tab. A title, a description, the location, the frequency, the impact, the violated heuristic, and a recommendation can be entered, as seen in Figure 4.1. Frequency and impact have a scale of three options (high, medium, low). The severity is calculated as average of those two, but can also be overwritten. Highlighting positive findings is not possible. Screenshots cannot be saved. The set of heuristics can be set in the template. The template files for Review Assistant are defined in a specific, text-based format with the file suffix `.tra`. Two templates are available per default: one for web usability, and one for retail premises reviews. This supports the tools focus to be not only on software. Additionally, Review Assistant also supports a tab for guideline checking. The guidelines are defined in the template file as well.

As already mentioned above, only the full version supports a report export. Due to the discontinuation, report exporting is therefore not possible anymore. An example of a report is preserved in the Web Archive [Gaffney 2004]. The missing option of attaching screenshots or other media data to a finding is a strong disadvantage of Review Assistant.

4.2 uzReview

uzReview was a browser extension for Mozilla maintained by the Uzilla group, became later part of commercial framework *uzilla.net* and discontinued as an extension [uzilla 2003a; Loitzl 2006, pages 49–51]. Mozilla itself is also discontinued and replaced by Mozilla Firefox already for a long time. It is still possible to run a version of Mozilla version 1.3 and get the outdated extension files for uzReview version 0.7 [uzilla 2003b] and the depending on extension JSLib [uzilla 2012]. Due to the long discontinuation, it is not possible to open modern websites like `https://tugraz.at` in the outdated Mozilla version.

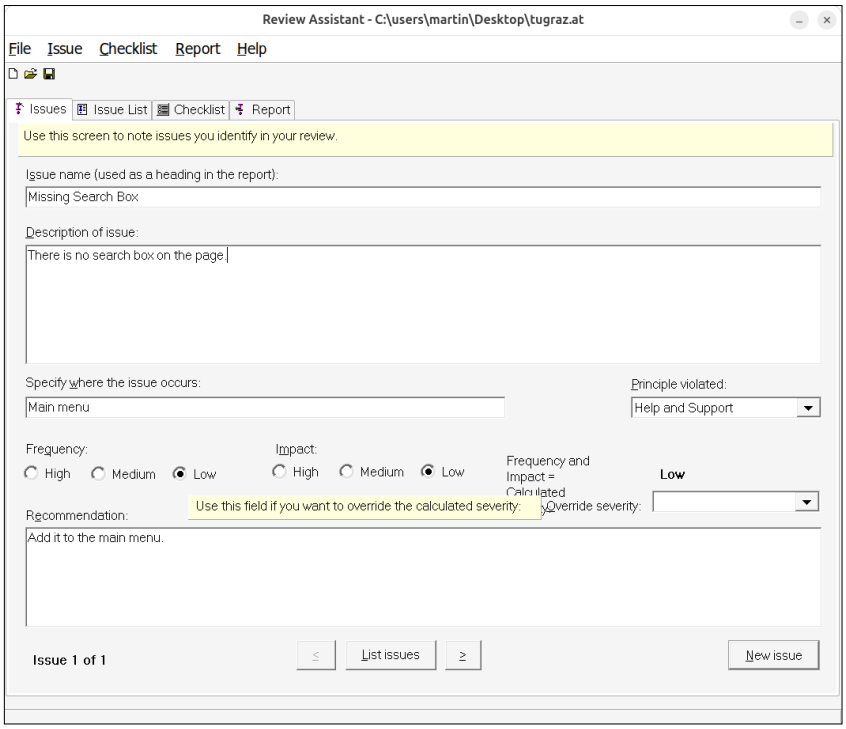

Figure 4.1: Review Assistant: Adding a finding in Review Assistant, based on text boxes, radio buttons, and select boxes. Adding other media, like screenshots, is not possible. [Screenshot by Martin Rabensteiner.]

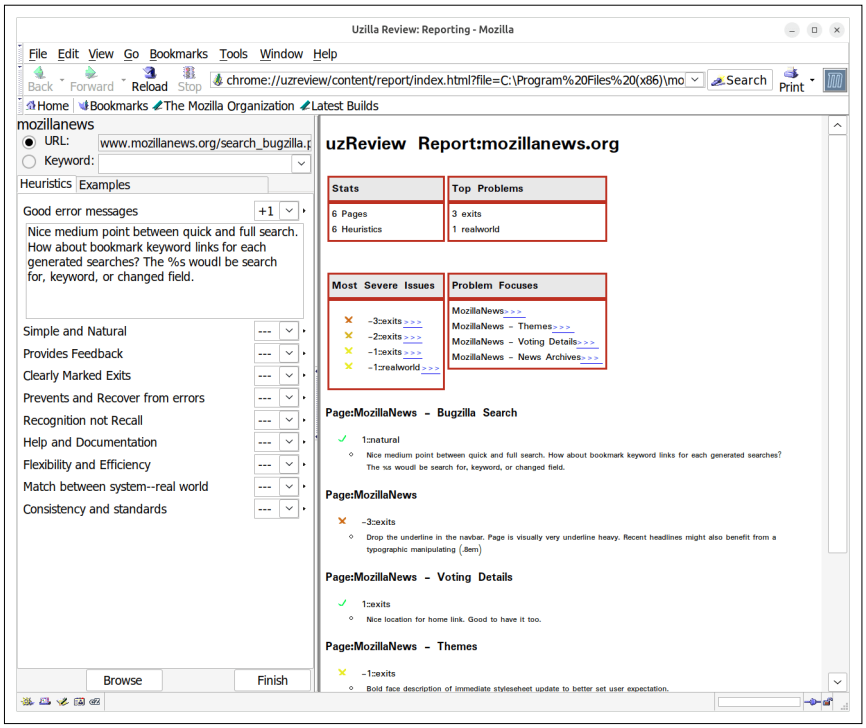

Figure 4.2: uzReview: Finding tracking with uzReview in the sidebar to left. On the right, a sample report is shown. [Screenshot by Martin Rabensteiner.]

The tool interface is in Mozilla a sidebar to the left. On the main screen, the user sees a list of projects and can add a new project below. When a project is opened, uzReview always tracks concerning page automatically in the URL field. Therefore, a finding is always coupled to a URL. Additionally, findings can be logged per keyword, for example to capture workflows. For each finding, there is a severity and a text area for each heuristic. There are four sets of heuristics predefined. Additional ones can be added in an XML format. The severity ranges from +1 to −4 (see Figure 4.2).

When the evaluation is finished, the report can be shown within the browser window and saved as HTML file or printed as a PDF. The report has some table as stats and overview on the top and below the findings listed per page. Heuristics are displayed as there ID and not as their user-friendly title.

4.3 UX Check

UX Check is a browser extension for Google Chrome by Chris Gallello [Gallello 2014]. The code is licensed under the *General Public Licence* and available on GitHub and was last updated in 2017 [Gallello 2017]. The last version in the Chrome Web Store dates from 2019 [Gallello 2019]. Since then, Google Chrome deprecated its extension manifest version 2 and allows version 3 only in its current releases [Google 2021]. With an older version of Google Chrome, for example version 138, the tool is still usable.

When activated, UX Check is shown as a sidebar to the left. The project's heuristics are always shown with a short description in the sidebar. Use heuristics, the Nielsen heuristics or custom ones can be used. While hovering over the page, the evaluator can select on HTML element that should be highlighted. For one finding, a description, and a recommendation can be entered. A heuristic and a severity on a scale from zero to four can be selected. Adding findings without a highlighted screenshot is not possible. This behaviour is similar to Heurio (Section 5.11) and can also be seen in Figure 4.3. The darkening can be changed to lightening in the settings. The sidebar always stays in a dark colour scheme. Findings can be viewed in a tabular view. Deleting findings is possible, but not editing.

The evaluation can be exported as a Microsoft Word file. Findings are printed one by one, with the heuristic, notes, recommendation, and screenshot. With the process of the export, the evaluation is finished and can not be recovered.

4.4 UX Teardown

UX Teardown is a border case in sense of heuristic evaluation tools, since it does not link heuristics to findings. However, due to its high similarity to the tools mentioned above it is still listed here. UX Teardown is as well a Google Chrome extension and developed by Christopher Silvestri. A licence is not mentioned within the store page or the code [Silvestri 2022].

Shift + F adds a finding with a full screen screenhot, Shift + A lets select an free range area on the page and Shift + S adds a text only finding. As already said, a heuristic can not be added.

In the dialoug window, a title, a description, an impact and an effort can be entered. The severity is calculated automatically from the last two.

4.5 Heuristic Evaluation Inspector

Similar to UX Check, the Heuristic Evaluation Inspector is a Google Chrome extension as well. Developed by Mohamed Amasha, it is quite recently updated and compliant with the browsers current extension guidelines. It is published under the MIT licence.[Amasha 2025].

The user interface is located as a sidebar to the right with four tabs: *Capture* for adding a finding, *Notes* for the overview of findings, *Export* for exporting a report and *Settings*. The tool does not differentiate between projects. URLs are tracked, but there is no filtering possible. The only way to end an old and

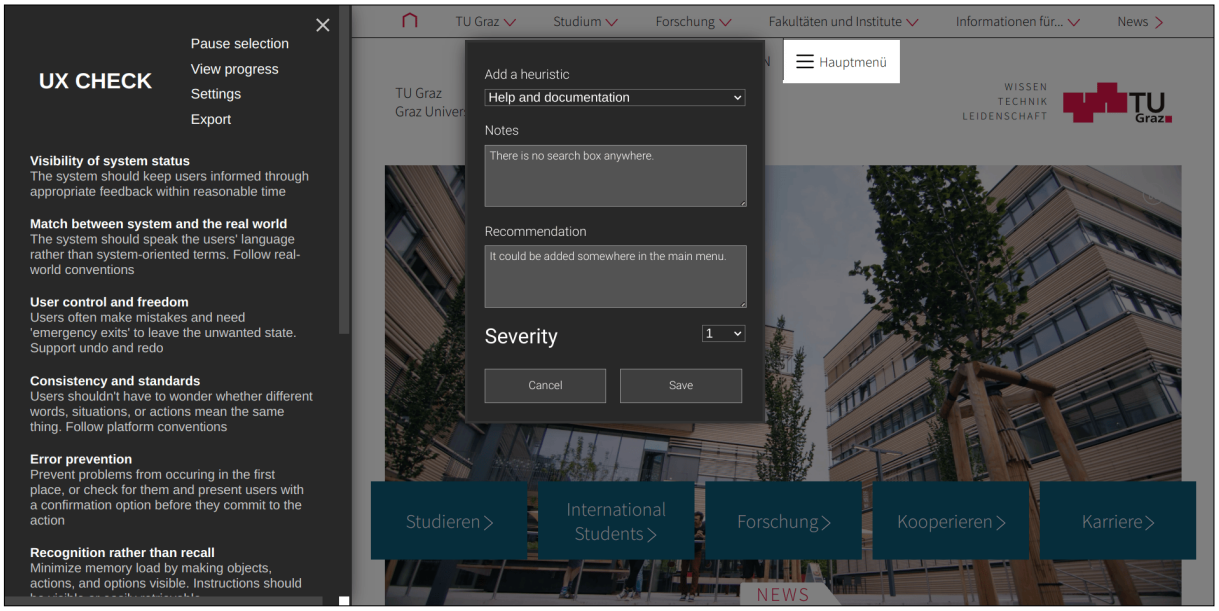

Figure 4.3: UX Check: The sidebar with the set of heuristics is always visible to the left. When hovering over the page to evaluate, everything than the hovered element is darkened. After clicking on the element, a context menu for the finding's metadata appears. [Screenshot by Martin Rabensteiner.]

start a new evaluation is to clear all data. The set of heuristics is limited to the 10 Nielsen heuristics plus an option for others. The caption of screenshots is not fixed to HTML elements, as in similar tools. For the screenshot, a free range on page can be selected. The screenshot's resolution is a little smaller than the actual region and therefore a bit unsharp. The adding of custom screenshots is not supported. As further data, a description, a recommendation, and a source URL can be entered. The severity ranges in steps from 1 to 4 and the creation date is set automatically. When added, findings can not be edited anymore, only removed.

For exporting, an evaluator's name can be added. The report is then provided as print to PDF. The first page is a short overview with a statistic about the severity and heuristic distribution. Per page, each finding is then described with all details.

Chapter 5

Online Tools

This chapter covers the heuristic evaluation tools that do not have to be installed and can be used within web browser.

more text

5.1 H&J

In 2007, Ebba Hvannberg and Effie Lai-Chong Law and Marta Lérusdóttir conducted a study about the difference between reporting usability problems using a web tool or on paper. To support the study, they built the *H&J* tool. It is used for collecting usability problems, in comparison to the paper form. To report an issue, the user can enter a description, the likely difficulties, the placement of the error and the cause and select the fitting heuristic and severity. The tool is self is not published, and further information is not available [Hvannberg et al. 2007].

5.2 Middlesex University (MDX)

5.3 Systematic Usability Inspection Tool (SUIT)

The Systematic Usability Inspection Tool (SUIT) was implemented by Carmelo Ardito, Rosa Lanzilotti, Paolo Buono, and Antonio Piccinno from the University of Bari in 2006 [Ardito et al. 2006]. Unfortunately, the code is also on request not available. Therefore, any description bases on the paper. SUIT is written in PHP, needs an Apache web server, and a MySQL server for the database.

Projects are organised as multi-user evaluations. The multi-user support is a main goal of SUIT. A project manager assigns multiple evaluators to the project. Evaluators are notified per email about changes. For one finding, a location, a heuristic (called principle), a severity, a description, and a suggested solution are entered by every evaluator. After all evaluators have entered their findings, the project manager can merge the findings detected by multiple evaluators. The last step for the evaluators is the discussion. All users of a project examine the findings of other users and contribute agreements, disagreements or the own point of view. Finally, the project manager reviews the discussions and makes a list for the final inspection report. This report can be printed or saved.

5.4 Usability Heuristic Evaluation Tool (UHET)

5.5 URM (UsabML)

5.6 Heuristic Evaluation Manager (HEM)

The Heuristic Evaluation Manager (HEM) is a Master's thesis by Martin Loitzl, conducted in 2007 at the Institute for Information Systems and Computer Media (IICM), Graz University of Technology and

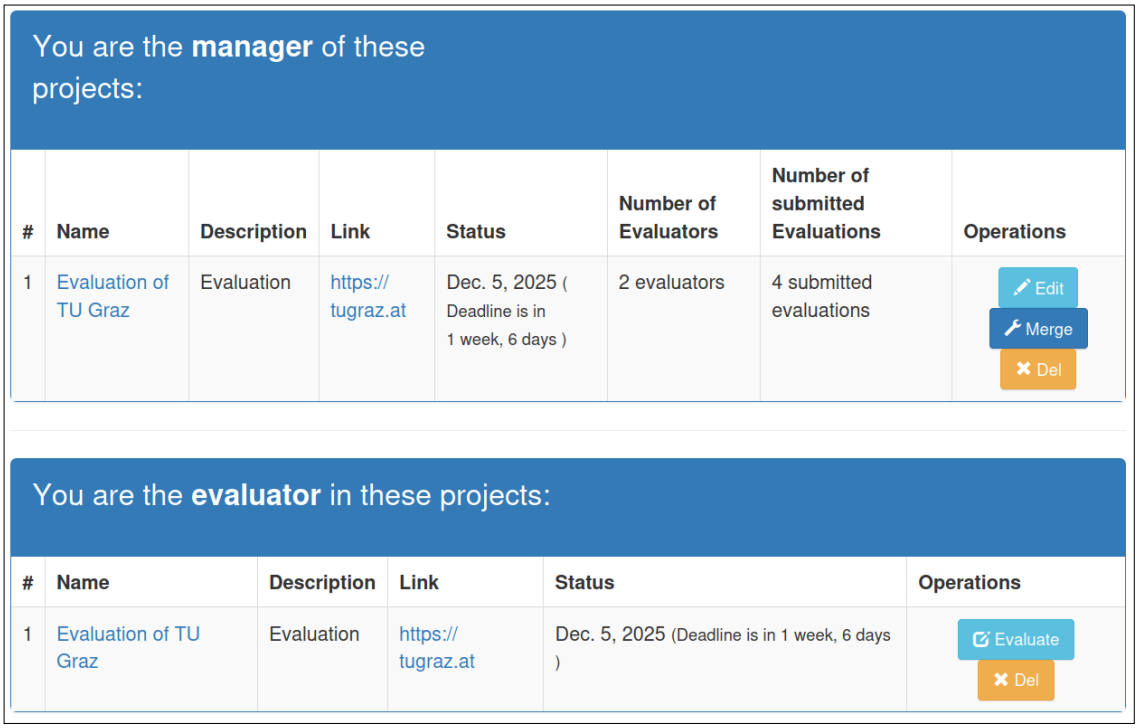

Figure 5.1: SHE: The dashboard in with the manager and evaluator view for the same project. [Screenshot by Martin Rabensteiner.]

supervised by Keith Andrews [Loitzl 2006]. HEM is implemented in PHP 4, needs an MySQL database to store the data, and can be hosted locally. The tool was not further developed since its release, but Loitzl provides the code on GitHub and published under the MIT licence[1].

HEM aims to be a multi-user and platform independent solution, to evaluate websites in any browser as well as installable applications. It allows to create several projects with several evaluators. The user roles are

5.7 Distributed Heuristic Evaluation Tool (DHET)

In his Master's thesis, Johannes Ortmann [2007] at University of Technology Vienna described and implemented the *Distributed Heuristic Evaluation Tool* (DHET). The structure of the application as well as the content and the research focus on the thesis are very similar to HEM (Section 5.6). The tool aims to support usability experts as a central information platform and to advance administrative processes.

5.8 Smart Heuristic Evaluation (SHE)

Isaac Rahnema wrote his Bachelor's thesis at the University of Paderborn in 2017 about a the *Concept and Implementation of a Smart Web-Application for Heuristic Evaluation Based on Data Mining Methods*. Within this thesis he developed the tool *Smart Heuristic Evaluation* (SHE) [Rahnehma 2017a]. The project is under the MIT licence public available on GitHub [Rahnehma 2017b]. SHE is implemented in Python 2.7 using the Django framework (version 1.9) for web rendering.

Within the thesis, he also surveys existing and related tools. Rahnema names collaboration between facilitators and the recognition of similar findings as two key features for a useful tool and that the

[1] https://github.com/mloitzl/hem

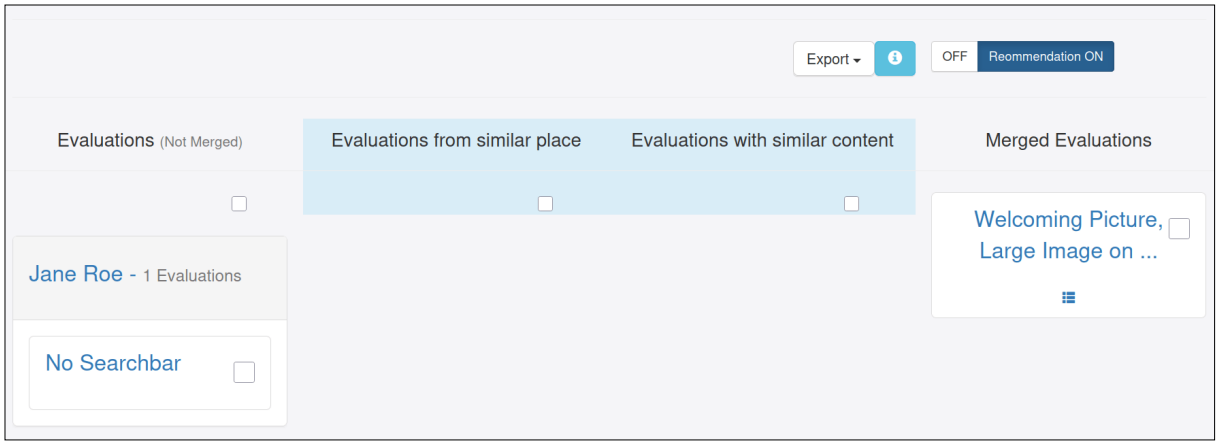

Figure 5.2: SHE: The merging view in SHE. Left are unmerged findings grouped by user. When two findings are selected, a button for merging would appear. The two middle columns are for recommended findings when selected. Merged findings are in the list to the right. [Screenshot by Martin Rabensteiner.]

examined tools lack either one or both of them. Further, he draws a strong connection to the Heuristic Evaluation Manager from Martin Loitzl (see Section 5.6). He mentions it as a good example for a heuristic evaluation tool and cultivates and revises many of its concepts in SHE. UX check (Section 4.3) and uzReview (Section 4.2) are also addressed.

In SHE, several users can work together and can enter their findings (called projects) individually. The overview of project is delivered by a dashboard with all involved projects, as seen in Figure 5.1. Confoundingly, findings are also called evaluations within the user interface. A finding can be distinguished between positive and negative and can be one of five points on a severity and frequency scale. Further, the user can enter text into the fields a title, the place, link, tags, description, and a recommendation (for a possible fix). Independent of an evaluation, users can document their environment with the fields age, gender, operating system, browser, monitor size, monitor resolution, and *other relevant data*.

As a downside of HEM, he mentions the merging process and that the merging facilitator receives no help from the system for this, as already mentioned above. To address this problem, Rahnema sets the easier merging as one of the main goals of SHE. When in the merging process, selecting a finding triggers the recommendation engine. Recommendations are made based either on the place- based or content-based. Content-based recommendations are calculated with the term frequency - inverse document frequency (TF-IDF) method, applied to the fields description, recommendation, and tags, and combined with the nearest neighbour algorithm. Place-based recommendations are calculated by the TF-IDF. Findings with the same link are also added to this list of recommendations.

For the merging recommendation, SHE uses the library GraphLab, for which a licence is needed. GraphLab was replaced in 2018 by the open source software Turi create, which later also was discontinued [Apple 2021]. With this, also the licensing stopped functioning. In combination with the remaining, also outdated software stack, it was not possible to restore the functionality of the recommendation engine. The merging view is shown in Figure 5.2.

5.9 Testpad

Testpad is rather a tool to conduct checklists, but can also be suitable for heuristic evaluations when set up right.

The free trial is valid for one week. The licences start from $ 59.- per month with up to three users (approximately € 17.- per user). To include images, at least the next larger package for $ 99.- per month with up to ten users is needed. The licence policy already shows that Testpad is very multi-user oriented.

The subdomain also is adapted to the users organisation, for example `tugraz.testpad.com`.

To use Testpad for heuristic evaluations, it is suitable to first create a template for those. Libraries in Testpad contain checklist template. In this example, this list is templated is defined through the set of heuristics on the top layer. One layer below is a placeholder finding for each heuristic, to lift to title to the top layer. The description of the title contains a further description of the heuristic. The template can then be duplicated to a project. This can be done for all evaluators separately to keep the findings separated. With notes, an introduction, conclusion, or other short comments can be added to the evaluation. Text format is not supported and therefore not suitable for those fields. When evaluating, users can replace the placeholders.

In the test report, *test detail* needs to be turned on and script sections to *inline* in order to see all findings. Positive and negative findings can be differentiated through the check status of a finding. Tags like severity or priority could only be depicted as tag. However, tags are not shown in the report and can not be evaluated even less.

5.10 Capian

Capian is an online tool by the Canadian company UX-co Conseil Inc.[Capian 2025c]. The paid version is available from € 35,- per month and user, up to 15 users. For larger teams, special plans are offered on request. The free trial has the full function range, but works only for seven days. After that, it is still possible to view the projects, but not to edit or export them.

Eleven sets of criteria are already available by default. Custom ones can be added in the user settings. Custom scales for priority and effort can be defined as well. Another customisation feature is to upload a logo that is then printed at the very top of the report. The Capian branding will still be present.

Adding a finding in Capian is equivalent to uploading a screenshot. A finding can not exist without a corresponding screenshot. One or more criteria, a description, a recommendation, and an arbitrary number of tags can then be added to a finding. For priority and effort, a selection of four states for each is possible. The severity can be minor, moderate, critical, or observation for a positive finding. The status can be open, completed or approved, as seen in 5.3.

The dashboard functions as project overview and has a couple of filter options to the top. The findings are below, organised as cards, and can be dragged and dropped to reorder them. On the top and the bottom are two text fields to support the project with an introduction and a conclusion. Simple text styling options like bold, italic, links, or lists are possible through all text fields. The collaborative work on projects is possible, but there are no features to differentiate between findings, for example merging.

Simple annotations to the screenshots are possible directly when editing the finding. Rectangles, arrows, circles with counting numbers, and the blackening of areas are possible. Users can add text comments to the finding. These will not remain internal, are also shown in the project report, but not on the report.

Capian provides browser extensions for both Mozilla Firefox [Capian 2025b] and Google Chrome [Capian 2025a], as well as mobile apps for Android [Capian 2024] and iOS [Capian 2023]. With the browser extension, the adding of findings including screenshot annotation and filling out the fields can be done directly when conducting a website. It is not necessary to take a separate screenshot or to switch the tab. In comparison to Heurio (Section 5.11), the project is selectable when adding a finding and findings can be added to any project. Findings can then be edited later in the project overview. A downside to classic screenshot through the system itself is that they are not as sharp in terms of image quality. The mobile apps on the other hand come with an integrated browser. This makes Capian the only tool in this survey that works native on a mobile phone. Screenshots can also be uploaded from the phones image gallery, for example when evaluation mobile apps.

A preview of the report can be seen directly in HTML. On top is the title, the user's preset logo, and the introduction. After the findings, the evaluation concludes with same statistics about the distribution

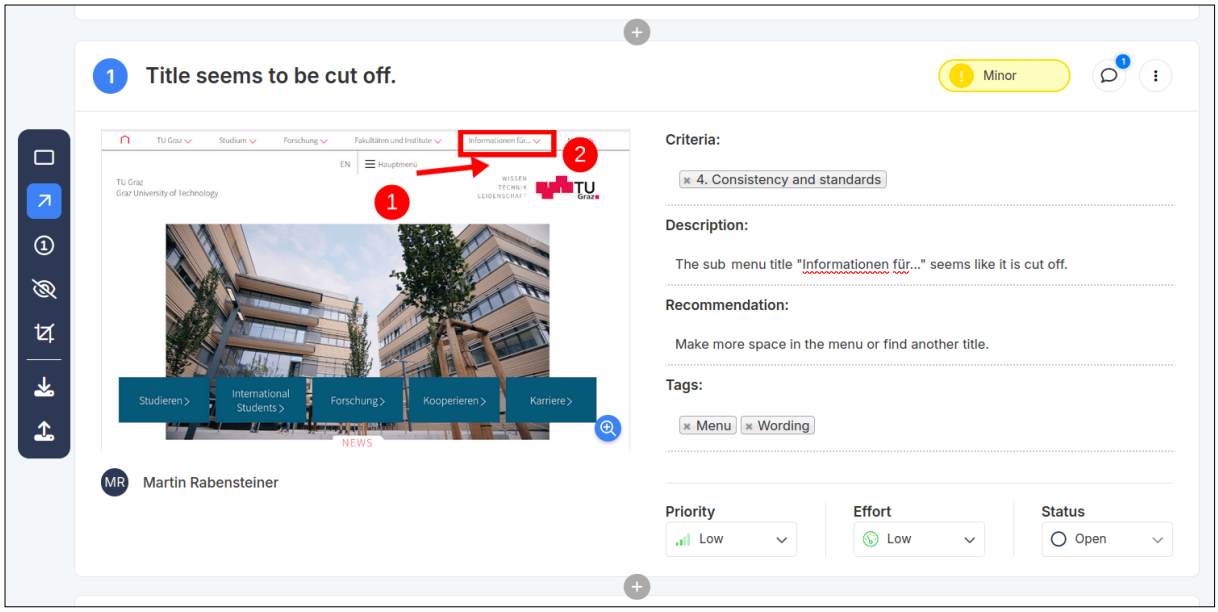

Figure 5.3: Capian: Adding a finding is equivalent to uploading a screenshot. Left to the Screenshot are the annotation possibilities. Right to it is the meta data for a finding. [Screenshot by Martin Rabensteiner.]

of rules, severity, tags, and the URL distribution. The link to the report can also be shared and viewed by users without a Capian account. Exported reports are generated in the background, a download link is then sent via email. The PDF reports have the Letter aspect ratio as their format and have one page per finding. The layout of the finding is very similar to the one in Figure 5.3. The findings itself can also be exported as Excel file. The screenshots (called captures) can be downloaded all together within a ZIP file. Another export possibility are PowerPoint files. In the long option, screenshots and text alternate per slide, while in the short version they are combined on one. Another possibility to proceed with findings and an outstanding feature of Capain is its cross-plattform integration. Findings can be directly exported as issues to Jira, GitHub and GitLab. For adding to GitHub, the GitHub account has to be connected to the Capian account. When exporting, a repository to export to can be selected. The issues then are added with the title, heuristic, description, recommendation, and link to the corresponding Capian issue, as seen in Figure 5.4.

Capian is a modern, commercial tool that meets many requirements. It scores with innovative features from extensions and apps to integrations into other tools. However, it lacks the possibility to merge findings and is therefore not very appropriate for evaluations with multiple evaluators, and the ongoing costs are quite high compared with other commercial tools.

5.11 Heurio

Heurio is a closed source, commercial online service developed by Apptum Hungary Ltd. The premium version is available from $ 9.9 per month and as a free test version for two weeks. The free version is limited to two projects [Heurio 2023].

The online service is strongly connected to its mandatory Google Chrome browser extension. This limits the tool to this browser and websites itself. Projects are strictly connected to website URLs. Manual adding of issues for example screenshots from other browsers or installable applications are therefore not possible. Multiple users can be added to a project, either with full access or with view-only permission.

To add a finding (called *heurio*), the concerning website needs to be opened in a separate tab. With the browser extension activated, any HTML element can be selected to add a finding. With a context menu,

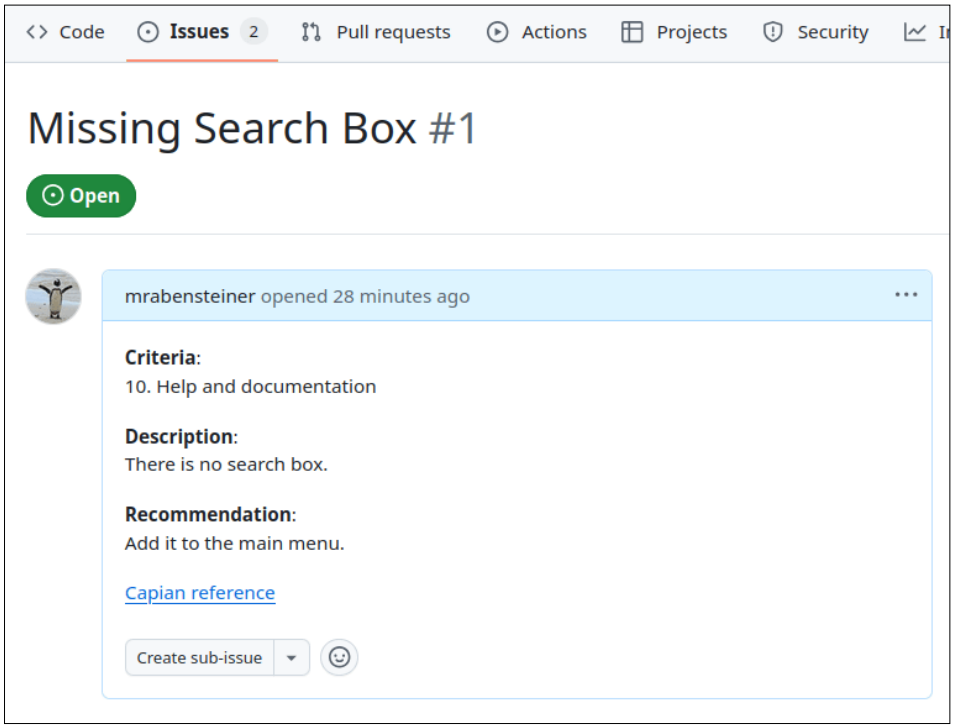

Figure 5.4: Capian: Findings can be exported as issues to GitHub. [Screenshot by Martin Rabensteiner.]

a description, a suggestion, and the guideline for this finding can be entered. It is helpful, that there are many keyboard shortcuts and hints for them to accelerate finding adding. Added findings remain as circle with their ID on the screen and can be opened in the sidebar for more details and further remarks, as assignee, due dates, etcetera.

Already ten sets of heuristics are offered, and more custom sets can be added. The severity has a scale of five options, including neutral which can be used for positive findings. System environment parameters like the operating system, the browser, the viewport size, or the device (PC, tablet, phone) are tracked as well and cannot be changed. Screenshots are automatically taken from the current page's viewport with everything darkened but not the selected HTML element. A downside of this is, that the mouse pointer does not get captured and mouse actions like hovering get lost while taking the screenshot.

The findings main overview is organised like a Kanban board with a column for each severity. The findings ID is also always present in the overview. The view can be changed to a customised with own column titles. Findings can be drag-and-dropped to be moved from one severity to another. Tabs above the board let the view switch between all findings and specific pages. The board offers many options for filtering, for example by the facilitator, the due or creation date, or the device. Further, a view with the findings as a list and an analytics dashboard can be shown. Bar charts for findings per user, findings per device, and findings by rules as well as pie charts for severity, and status (active/resolved) are offered, as seen in Figure 5.5.

To share a report, a clean summarisation of the findings is generated as HTML, ready to be saved as PDF (print to PDF). The names of the users can be hidden as well as findings outside a specific date range, findings by specific users, or findings in specific paths. The findings themselves can be exported as a CSV file.

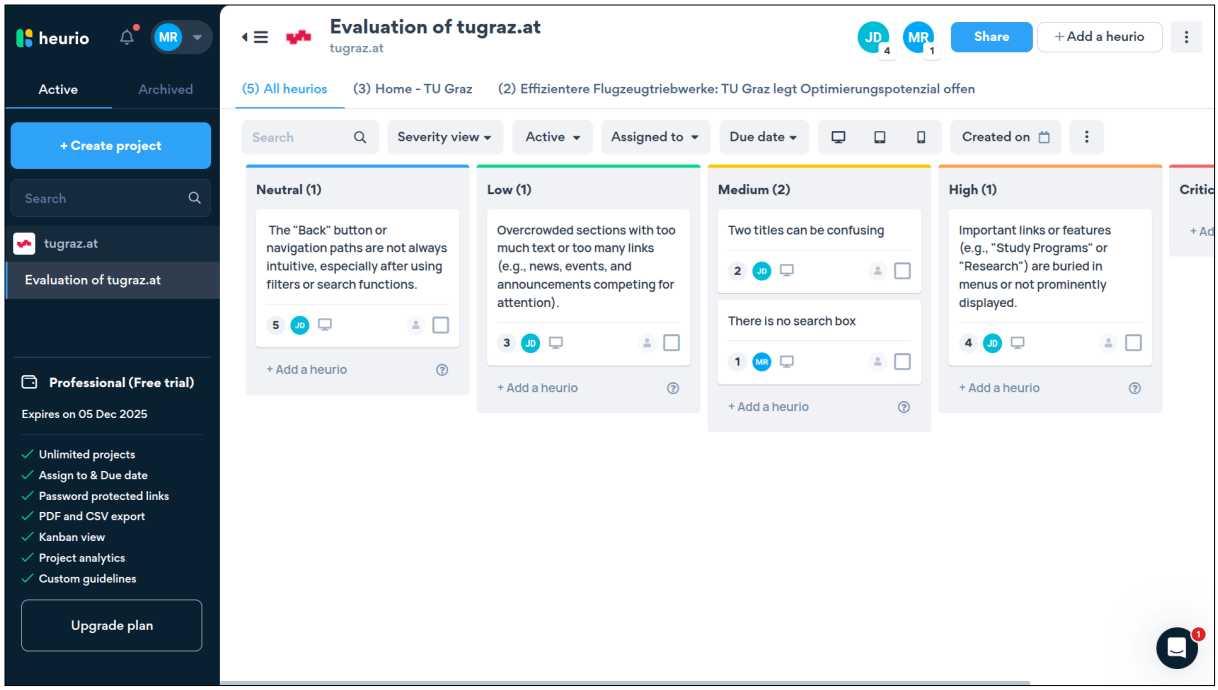

Figure 5.5: Heurio: Findings are by default organised as a Kanban board and sorted by severity. [Screenshot by Martin Rabensteiner.]

5.12 Heurix

Heurix was released in 2020 and was then acquired in 2021 by the UX agency Cyber-Duck [Heurix 2025]. The tool is an online service, currently in beta, and with only a limited set of features. An evaluation can only be performed by a single evaluator, and only in response to a predefined set of questions. Heurix is closed source, but free to use.

A user must first register for an account, and can then create an (individual) evaluation project. When creating a project, the user can select one or more of seven sets of criteria (called sections): First Impressions, Site Navigation, Information, Trust and Persuasion, Interaction, Forms, and Search. In total, there are 48 criteria across the seven sets, formulated as questions such as: "Is the search box visible wherever you are on the site?".

When evaluating, the user can choose between four responses to each criterion: Yes, Room to Improve, No, or Not Applicable. Further, one screenshot and one plain-text note can be added, as shown in Figure 5.6. Each criterion is shown separately to minimise distraction. After going through all criteria, the responses are summarised in a Results page. An overall score at the top of the page is calculated by how many criteria are met, as seen in Figure 5.7. A met criterion counts as one point, partially met half a point, and unmet zero points. The sum then is divided by the overall number of criteria, excluding any not applicable criteria. Next to the overall score, a generic text describes the attained score.

Projects can be re-evaluated by the same evaluator, at different points in time. In the Results page, a particular version of the evaluation can be selected to be shown or exported. In theory, it would be possible for different users to use the same account to evaluate the same interface (to simulate a classic HE), but there is no possibility to compare or merge the different iterations of the evaluations with one another. The scores of different iterations are also only partially comparable, because the number of criteria that are not applicable may differ.

The results of an evaluation can be exported as PDF in two different modes. Compact mode is very similar to the Results page in the browser; screenshots are not included. The more detailed version contains the questions asked per criterion, notes, and screenshots.

Heurix offers a simple quick start solution to performing an individual evalution with a predefined set

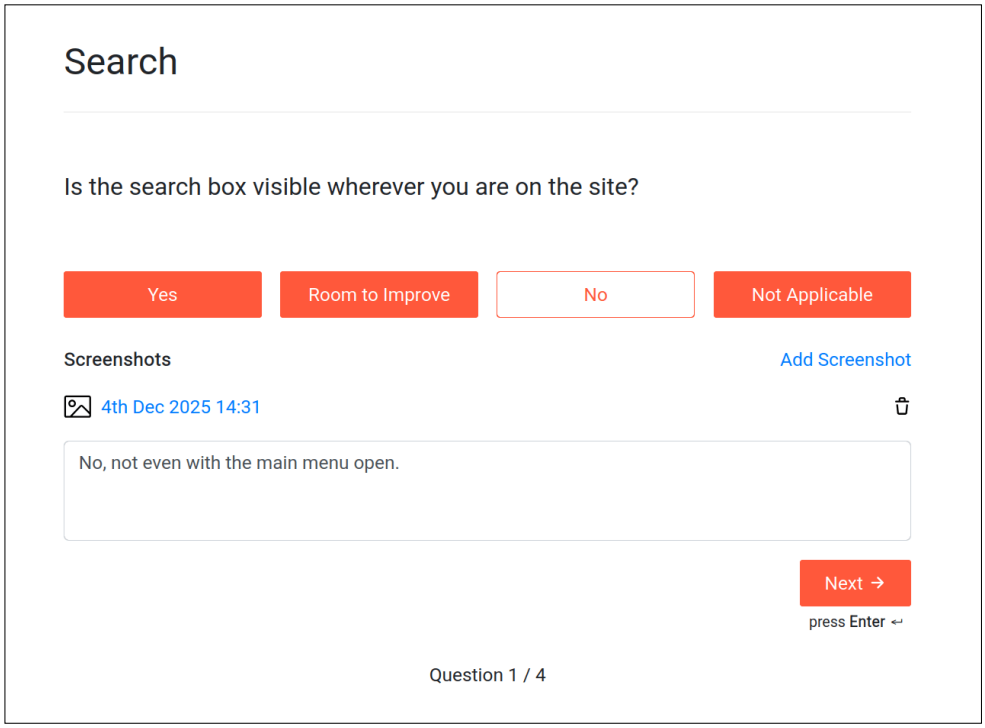

Figure 5.6: Heurix: Criteria are evaluated by asking preset questions. There are four possible responses. Furthermore, a screenshot and a note can be added. [Screenshot by Martin Rabensteiner.]

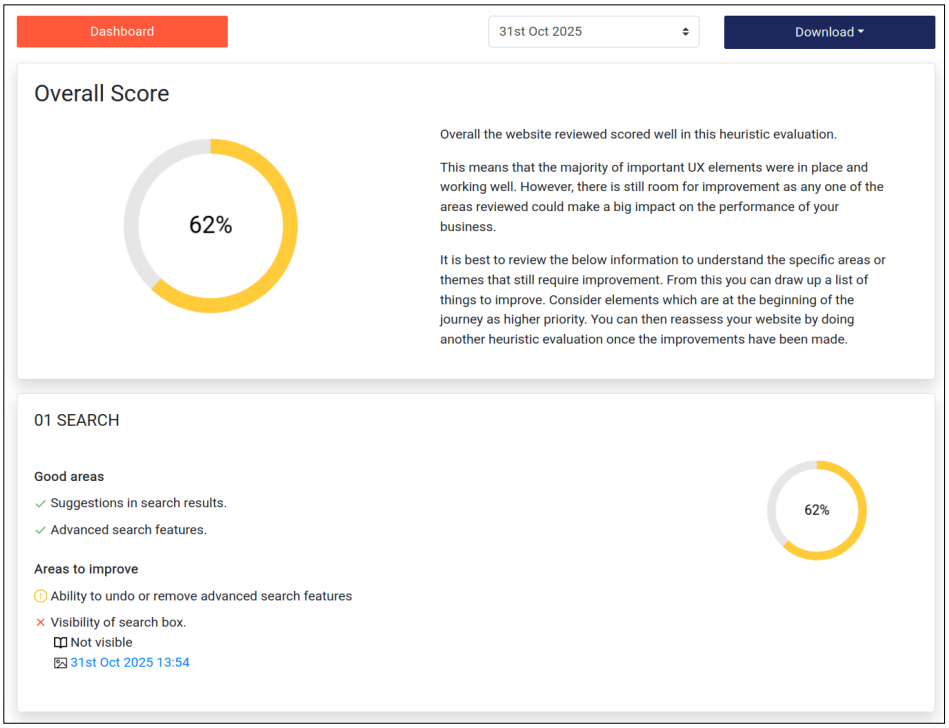

Figure 5.7: Heurix: The Results page has an overall score and an overview of the criteria and responses. [Screenshot by Martin Rabensteiner.]

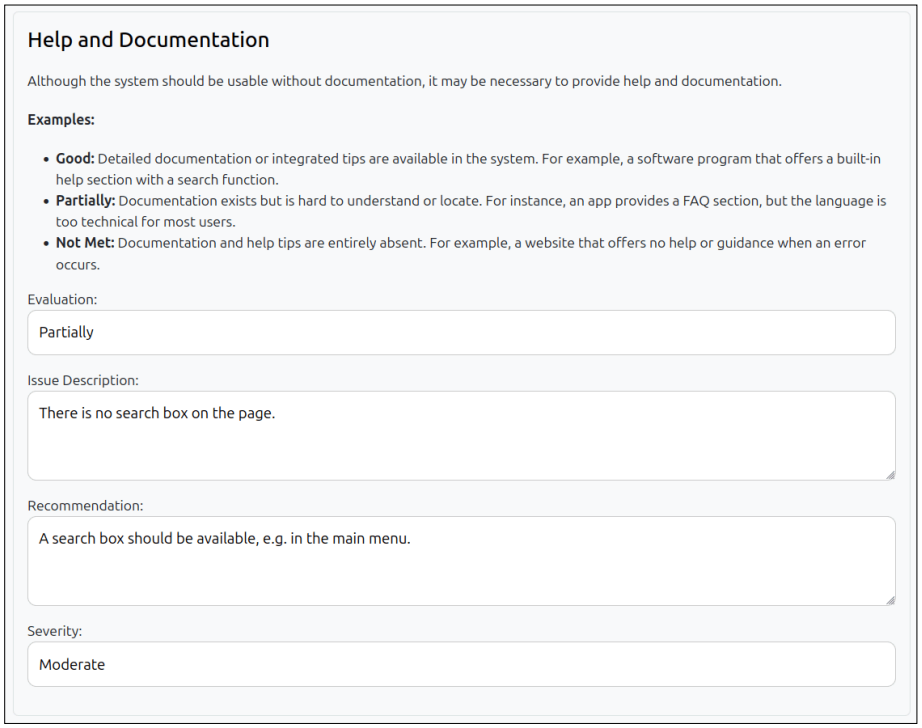

Figure 5.8: ISO9241.org: In the second step on ISO9241.org, the user goes through the heuristics per screenshot and can enter his finding data. The lower to fields only appear if the evaluation is not *good*. [Screenshot by Martin Rabensteiner.]

of criteria. The principle of leading users through the evaluation does not require much knowledge about the tool. Through its simplicity, it may not be suitable for larger projects with multiple evaluators or custom requirements.

5.13 ISO9241.org

The website ISO9241.org provides a simplistic approach to generate a heuristic evaluation report [ISO9241.org 2025]. The owner, the code, and other technical parameters are unknown. The website is registered in the Ukraine [Whois 2025]. The name refers to the ISO 9241:210 standard [ISO 2019], although there might be no organisational connection.

Users can upload one to maximum five screenshots to the page. In the next step, the screenshots are confronted with the 10 Nielson Heuristics. In the second step, the Nielson heuristics are listed with a short description. For each screenshot, the evaluator can select between *good*, *partially*, and *not met* for the concerning heuristic. Examples for the heuristic are given as well to facilitate the selection. A description can also be added to support the decision. When the rating is not *good*, another text box for improvement recommendations and a select box for the severity (5 steps) appear, as shown in Figure 5.8.

After conducting all the screenshots, the system leads to an overview page. In two textboxes the facilitators name and a description or the report can be entered. The report description contains already a structure with the numbers calculated from the evaluation. Below are the heuristics in a table with the score, description, severity, and the recommendation of the findings. The descriptions of positive findings are somehow lost. On the bottom is again the list of findings, ranked by their severity.

Finally, the report can be exported as PDF (print to PDF) or the link can be shared for one hour. After this time frame, every personal data is deleted. Keeping the report online for a longer time or other standard user authentication features are not available.

The simplicity of ISO9241.org enables a fast and uncomplicated heuristic evaluation. The missing

scalability. The missing user management features and limitation of screenshot makes it unappealing to use it for a larger scale. Due to its non-transparency, it is not foreseeable who long the website is being supported and continued.

5.14 HeuristicsEvaluationTool

5.15 Overview

The following Table 5.1 gives a more structured comparison about the tools mentioned above.

name	**Heurix**	**Heurio**	**Capian**	**ISO9241.org**	**HEM**	**Testpad**
availability	online service	online service	online service	online service	code, thesis	online service
owner	Cyber-Duck	Apptum Hungary Ltd.	UX-CO Conseil Inc.	unknown, registered in Ukraine	Martin Loitzl	Testpad Ltd.
launch date (year of publication/launch)	2020	2020	2014	2024	2005	2010
last update [discontinued]	ongoing	ongoing	2025	2025 (legacy build)	2025	2025
licence (open-source, commercial)	closed source	closed source	closed source	closed source	MIT	none
service	online service	online service	online service, app	online service	self-hosted	self-hosted
hosted and/or self-hosted	hosted	hosted	hosted	hosted	self-hosted	hosted
technology stack	unknown	Vue.js	Ruby	unknown	PHP 4.4.9, MySQL 4.1.22	Python
dark and/or light mode	light	light	light	light	light	light
responsive (review on tablet?)	✓	✓	✓	✓	✗	✓
multiple reviewers	✗	✗	✗	✗	✓	✓
reviewer management	✗	✓	✓	✗	✓	✓
sets of heuristics	fixed	fixed and custom	fixed and custom	fixed	fixed and custom	custom
positive/negative findings	✗	✓	✓	✓	✓	✓
attach a description to a finding	✓(one)	✓	✓	✓	✓	✓
attach a screenshot to a finding	✓(one)	✓	✓	✓	✓	✓
attach a video clip to a finding	✗	✗	✗	✗	✓	✗
support for guided questionnaires/checklists	✓	✗	✓	✗	✓	✓
report export	PDF, PPT	PDF	PDF, PPTX	PDF	ZIP (HTML)	-
finding export	-	CSV	XLSX	-	-	TXT, CSV
collaboration, messaging	-	-	-	-	-	-
documentation	-	GitBook documentation incl. videos	guide, videos, UX resources	-	thesis	videos
ntegrations	-	-	GitHub, GitLab, Jira	-	-	
customisable branding	-	-	logo on report	-	-	
online service	`https://heurix.io/`	`http://heurio.co/`	`https://capian.co/`	`https://iso9241.org/`		`https://testpad.com`

Table 5.1: Online Tools - Prototype/WIP

Bibliography

Amasha, Mohamed [2025]. *Chrome Web Store - Heuristic Evaluation Inspector*. 07 Oct 2025. https://chromewebstore.google.com/detail/opjfancdaoajjokijfdmdjlmhocepnde (cited on page 13).

Andrews, Keith [2025]. *Human Computer Interaction: Lecture Notes*. 27 May 2025. http://courses.isds.tugraz.at/hci/hci.pdf (cited on pages 1–2).

Apple [2021]. *Turi Create Repository*. 29 Nov 2021. https://github.com/apple/turicreate/ (cited on page 17).

Ardito, Carmelo, Rosa Lanzilotti, Paolo Buono and Antonio Piccinno [2006]. *A Tool to Support Usability Inspection*. Proc. Working Conference on Advanced Visual Interfaces (AVI 2006) (Venice, Italy). ACM. 23 May 2006, pages 278–281. doi:10.1145/1133265.1133322. https://researchgate.net/publication/220944663_A_tool_to_support_usability_inspection (cited on page 15).

Axis [2017a]. *Evaluation Spreadsheet*. Axis Group, 07 Nov 2017. https://docs.google.com/spreadsheets/d/1lfcPwG4gH-rQQhl5MuXgNevy8_hlJPvdx6_RiLT34qw (cited on page 9).

Axis [2017b]. *Evaluation Toolkit*. Axis Group, 06 Dec 2017. https://github.com/axisgroup/evaluation-toolkit (cited on page 9).

Capian [2023]. *Capian: iOS App*. Version 2.0.1. 06 Jun 2023. https://apps.apple.com/de/app/capian/id1316858374 (cited on page 18).

Capian [2024]. *Capian: Android App*. Version 2.0.2. 04 Sep 2024. https://play.google.com/store/apps/details?id=com.capianmobile (cited on page 18).

Capian [2025a]. *Capian: Chrome Extension*. Version 2.0.5. 25 Apr 2025. https://chromewebstore.google.com/detail/edpkoigbbhaeapdgjohogckmeddejncp (cited on page 18).

Capian [2025b]. *Capian: Firefox Browser Add-On*. Version 2.0.1. 01 Feb 2025. https://addons.mozilla.org/en-GB/firefox/addon/capian/ (cited on page 18).

Capian [2025c]. *The All-in-One Tool for UI/UX Audits and UI/UX Reviews*. 2025. https://capian.co/ (cited on page 18).

Cockton, Gilbert, Alan Woolrych and Mark Hindmarch [2004]. *Reconditioned merchandise: Extended Structured Report Formats in Usability Inspection*. CHI '04 Extended Abstracts on Human Factors in Computing Systems (Vienna, Austria). ACM. 24 Apr 2004, pages 1433–1436. doi:10.1145/985921.986083. https://researchgate.net/publication/200553190_Reconditioned_merchandise_extended_structured_report_formats_in_usability_inspection (cited on page 9).

Gaffney, Gerry [2004]. *Review Assistant: Sample Usability Review*. 26 May 2004. https://web.archive.org/web/20080719232339/http://infodesign.com.au/usabilityresources/reviewassistant/report.rtf (cited on page 11).

Galavi, Zahra, Somaye Norouzi and Reza Khajouei [2024]. *Heuristics used for evaluating the usability of mobile health applications: A systematic literature review*. Digital Health 10 (2024). ISSN 2055-2076.

doi:10.1177/20552076241253539. https://journals.sagepub.com/doi/epub/10.1177/20552076241253539 (cited on page 2).

Gallello, Chris [2014]. *UX CHECK - Improve Your Website's UX*. 2014. https://uxcheck.co (cited on page 13).

Gallello, Chris [2017]. *UXCheck Repository*. 29 May 2017. https://github.com/cgallello/UXCheck (cited on page 13).

Gallello, Chris [2019]. *Chrome Web Store - UX Check*. 29 Jun 2019. https://chromewebstore.google.com/detail/giekhiebdpmljgchjojblnekkcgpdobp (cited on page 13).

Google [2021]. *Manifest V2 Support Timeline*. 23 Sep 2021. https://developer.chrome.com/docs/extensions/develop/migrate/mv2-deprecation-timeline (cited on page 13).

Hermawati, Setia and Glyn Lawson [2016]. *Establishing Usability Heuristics for Heuristics Evaluation in a Specific Domain: Is there a Consensus?* Applied Ergonomics 56 (Sep 2016), pages 34–51. ISSN 0003-6870. doi:10.1016/j.apergo.2015.11.016. https://sciencedirect.com/science/article/pii/S0003687015301162 (cited on page 2).

Heurio [2023]. *Review and Approve Websites 10x Faster in a Collaborative, User-Friendly Way*. 2023. https://heurio.co/ (cited on page 19).

Heurix [2025]. *Heurix: A Free Heuristic Evaluation Tool*. 28 Nov 2025. https://heurix.io/ (cited on page 21).

Hvannberg, Ebba Thora, Effie Lai-Chong Law and Marta Kristín Lérusdóttir [2007]. *Heuristic Evaluation: Comparing Ways of Finding and Reporting Usability Problems*. Interacting with Computers 19.2 (Mar 2007), pages 225–240. ISSN 0953-5438. doi:10.1016/j.intcom.2006.10.001. https://kth.diva-portal.org/smash/get/diva2:527675/FULLTEXT01.pdf (cited on page 15).

IaD [2004]. *Review Assistant Setup*. Information & Design, 21 Nov 2004. https://web.archive.org/web/20080719193841/http://infodesign.com.au/ftp/RASetup.zip (cited on page 11).

Information & Design [2006]. *Review Assistant*. 07 Aug 2006. https://web.archive.org/web/20080724190035/http://www.infodesign.com.au/usabilityresources/reviewassistant/default.asp (cited on page 11).

ISO [2019]. *Ergonomics of Human-System Interaction – Part 210: Human-Centred Design for Interactive Systems. ISO 9241-210:2019*. International Organization for Standardization, Jul 2019. https://iso.org/standard/77520.html (cited on page 23).

ISO9241.org [2025]. *ISO9241.org - Heuristic Evaluation Tool*. 28 Nov 2025. https://iso9241.org/ (cited on page 23).

Loitzl, Martin [2006]. *The Heuristic Evaluation Manager (HEM): An Online Collaborative Environment for Heuristic Evaluation*. Master's Thesis. Institute for Information Systems and Computer Media (IICM): Graz University of Technology, Feb 2006. 228 pages. https://ftp.isds.tugraz.at/pub/theses/mloitzl.pdf (cited on pages 11, 16).

Macefield, Ritch [2014]. *An Overview of Expert Heuristic Evaluations*. 02 Jun 2014. https://uxmatters.com/mt/archives/2014/06/an-overview-of-expert-heuristic-evaluations.php (cited on page 9).

Molich, Rolf and Jakob Nielsen [1990]. *Improving a Human-Computer Dialogue*. Commun. ACM 33.3 (Mar 1990), pages 338–348. ISSN 0001-0782. doi:10.1145/77481.77486. https://dl.acm.org/doi/pdf/10.1145/77481.77486 (cited on page 1).

Moran, Kate and Kelley Gordon [2023]. *Heuristic Evaluation Workbook*. Nielsen Norman Group, 28 Jun 2023. https://nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/ (cited on page 9).

Nielsen, Jakob [1993]. *Usability Engineering*. Academic Press, 01 Jan 1993. 377 pages. ISBN 0125184069. doi:10.5555/2821575 (cited on page 1).

Nielsen, Jakob [1994]. *Heuristic Evaluation*. In: *Usability Inspection Methods*. Edited by Jakob Nielsen and Robert L. Mack. Wiley, 1994. Chapter 2, pages 25–62. ISBN 0471018775 (cited on pages 3, 9).

Nielsen, Jakob [2024]. *10 Usability Heuristics for User Interface Design*. 30 Jan 2024. https://nngroup.com/articles/ten-usability-heuristics/ (cited on page 2).

Nielsen, Jakob and Robert L. Mack, editors [1994]. Wiley, 09 May 1994. 448 pages. ISBN 0471018775 (cited on page 1).

Nielsen, Jakob and Rolf Molich [1990]. *Heuristic Evaluation of User Interfaces*. Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI 1990) (Seattle, Washington, USA). ACM. 01 Apr 1990, pages 249–256. doi:10.1145/97243.97281. https://dl.acm.org/doi/pdf/10.1145/97243.97281 (cited on pages 1, 5).

Ortmann, Johannes [2007]. *Konzeption, Implementierung und Evaluierung eines Web-Tools zur Unterstützung von verteilten heuristischen Evaluierungen*. Master's Thesis. Institut für Rechnergestützte Automation: TU Wien, 30 Apr 2007. 78 pages. https://repositum.tuwien.at/handle/20.500.12708/14570 (cited on page 16).

Quiñones, Daniela and Cristian Rusu [2017]. *How to Develop Usability Heuristics: A Systematic Literature Review*. Computer Standards & Interfaces 53 (Aug 2017), pages 89–122. ISSN 0920-5489. doi:10.1016/j.csi.2017.03.009. https://sciencedirect.com/science/article/pii/S0920548917301058 (cited on page 2).

Rahnehma, Isaac [2017a]. *Concept and Implementation of a Smart Web-Application for Heuristic Evaluation Based on Data Mining Methods*. Bachelor's Thesis. Department of Computer Science: Paderborn University, 2017. 82 pages. https://cs.uni-paderborn.de/fileadmin-eim/informatik/fg/mci/Bachelorarbeiten/2017/Rahnema__Isaak.pdf (cited on page 16).

Rahnehma, Isaac [2017b]. *SHE Repository*. 04 Aug 2017. https://github.com/boogh/she/ (cited on page 16).

Silvestri, Christopher [2022]. *Chrome Web Store - UX Teardown*. 02 May 2022. https://chromewebstore.google.com/detail/giekhiebdpmljgchjojblnekkcgpdobp (cited on page 13).

TechSmith [2004]. *Introducing Morae*. 02 Apr 2004. https://web.archive.org/web/20040402063412/techsmith.com/products/morae/default.asp (cited on page 9).

TechSmith [2019]. *Introducing Morae*. 29 Dec 2019. https://web.archive.org/web/20191229201005/techsmith.com/morae.html (cited on page 9).

Travis, David [2007]. *Heuristic Evaluation with Morae*. 02 Aug 2007. https://userfocus.co.uk/articles/morae-he.html (cited on page 9).

TU Graz [2025]. *Graz University of Technology*. 18 Dec 2025. https://tugraz.at/ (cited on page 6).

uzilla [2003a]. *uzReview*. 2003. https://web.archive.org/web/20130316044619/http://uzilla.mozdev.org/heuristicreview.html (cited on page 11).

uzilla [2003b]. *uzreview_0_7_1.xpi*. 17 Sep 2003. https://ftp.knoppix.nl/os/Linux/distr/mozdev/uzilla/uzreview_0_7_1.xpi (cited on page 11).

uzilla [2012]. *jslib_current.xpi*. 01 Feb 2012. https://ftp.knoppix.nl/os/Linux/distr/mozdev/jslib/xpi/jslib_current.xpi (cited on page 11).

Whois [2025]. *Whois iso9241.org*. 02 Jan 2025. https://whois.com/whois/iso9241.org (cited on page 23).

Wilson, Chauncey [2013]. *User Interface Inspection Methods: A User-Centered Design Method*. Morgan Kaufmann, 15 Nov 2013. 132 pages. ISBN 012410391X (cited on page 1).

Yáñez Gómez, Rosa, Daniel Cascado Caballero and José-Luis Sevillano [2014]. *Heuristic Evaluation on Mobile Interfaces: A New Checklist*. The Scientific World Journal 2014 (11 Nov 2014). ISSN 2356-6140. doi:10.1155/2014/434326. https://onlinelibrary.wiley.com/doi/abs/10.1155/2014/434326 (cited on page 2).