

Application of Automated Metrics in the Evaluation of Software Usability

Adnan Šahić¹, Selma Habota², Samir Čaušević¹

¹PhD candidate,

²MSc od Information Technologies,

¹Professor Dr.Sc.

¹Faculty of Transport and Communications, Zmaja od Bosne 8, Sarajevo 71000,
Bosnia and Herzegovina

²Faculty of Electrical Engineering, Zmaja od Bosne bb, Sarajevo 71000,
Bosnia and Herzegovina

Abstract: In this paper application for automatic evaluation of user interactions was implemented. Automatic evaluation was implemented in application that is used in real work environment with real users. Set of chosen metrics used for usability testing has been implemented, and selection was made based on previous review of literature in this field. Application monitors user activities and collects data in log files, and based on this data performed analysis. After detailed analysis of the collected data, results are visualized, discussed and recommendations how to improve the usability of application is given.

Keywords: Evaluation, Metrics.

1. Introduction

This paper will describe the usability characteristic, which is of great importance when evaluating software quality. Usability entails usability evaluation. There are three usability evaluations: user, expert and automatic. Automatic evaluation is performed by the software itself by collecting data, recording user-computer interactions and giving developers reports on how to improve the software. Automatic evaluation requires interventions on the software itself, its planned and dedicated expansion, as well as tools for monitoring and analyzing data, which are collected by automatic evaluation. The topic of the paper is the importance of automatic evaluation and the principles of its application, which are demonstrated through a practical example.

2. Usability evaluation

Usability evaluation determines the extent to which users can use the product to achieve the given goals efficiently, effectively and with satisfaction in a specific context of use. Usability evaluation includes:

- Testing prototypes that were made only on paper
- Testing the appearance of screens that have partial functionality
- Testing prototypes that look finished but have a human responsible behind the scenes (known as the "Wizard of Oz" technique)
- Testing the working version of the software before it is officially released
- Testing software that has already been implemented in existing systems

Usability evaluation, also known as UX testing (user experience testing), is very important for companies that want to constantly improve their experience with users. Usability testing is meant to check how the end users will use the product and if they have any problems with the same. The feedback from these tests allows the development team to better understand their product from the user's perspective. After that, the company can use the results of usability testing to fix bugs and problems that users may encounter.

2.1 Automatic evaluation

Automatic evaluation is performed by the software by collecting data, recording user-computer interactions and giving developers reports on how to improve the software. A set of interface guidelines and software that compares guidelines to interfaces are often used. These applications are often used when a large number of interfaces need to be tested and little time is available for human needs-based reviews. The main strength of these applications is that they can read the code very quickly and look for usability problems.

Automated tools are considered to be the most efficient and least expensive way to perform the frequently repetitive parts of the evaluation. Such tools do not require a lot of time or skills on the part of the people who use them. This makes them a very useful tool for reducing the effort required to manually analyze a

complete Web application. Also, they are irreplaceable when it is necessary to determine the syntactic correctness of the Web application and respect for good practices (eg the presence of descriptive data about the multimedia content of the page). However, such tools cannot assess features that require human judgment (eg, the relevance of that data and the overall content of the site).

The challenge is to create an automated usability testing tool that is meant to be used by a developer, not just a usability testing expert. The main advantage is that these applications can read the code very quickly, looking for usability problems, which can be picked up. These applications usually have features to offer advice on how to fix the code or actually correct the code.

The main disadvantage is that many aspects of usability cannot be detected by automated means. Automated usability testing apps are good for measuring certain statistics, such as the number of fonts used, average font size, average size of clickable buttons, and average graphics load time.

2.2 Usability testing steps

In practice, these steps are adapted to specific applications and goals of automated evaluation, and a more detailed approach to automated usability testing is developed through 7 steps as described in Figure 1.

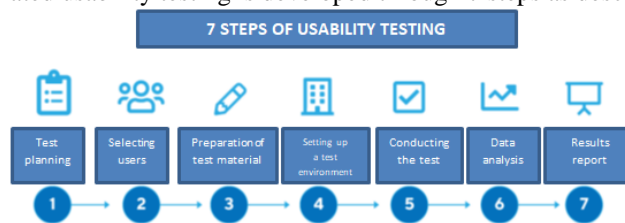


Figure 1 Usability testing steps

2.3 Usability metrics

There is no usability thermometer to tell how usable a software or website is. Instead, we rely on the impact of good and bad usability to assess the quality of the user experience. Testing with users isn't just about improving their experience. It's also about getting more people to use a particular product. People will not use a product that does not seem useful to them, so they will choose the one that is most convenient and usable for them if they have the option. This is especially true on the web. People leave websites if they can't find what they want, or if they can't choose what they want. Research shows that a website has 5 to 10 seconds for visitors to demonstrate their value. This article presents several usability metrics, as shown in the following table.

Table 1 Usability metrics

USABILITY COMPONENT	METRICS
<i>Effectiveness</i>	Task success Task completion
<i>Efficiency</i>	Duration of the task Steps to complete the task
<i>Satisfaction</i>	Rating scale for atmosphere, ease of use, usefulness Expressing satisfaction / disappointment
<i>Discoverability</i>	First click First impression Expressing satisfaction / confusion
<i>Learnability</i>	Task time for repeating tasks Task success for repeating tasks Number of errors for repetitive tasks Expression of skill / confusion
<i>Error proneness</i>	Number of errors Error Severity Level

2.4 Software tools for logging and recording user activities

Software tools for monitoring, logging and recording user activities can offer a lot of data for usability studies. Tools that track mouse or keystrokes and other interactions can help us identify common sequences, understand the actions used to complete tasks, and often collect data about transitions between different tools. These tools can be applied to many different applications, providing the ability to understand the use of e-mail, office productivity tools and basic operating system features.

These tools generally fall into two categories:

- proxy
- Interaction recording tools.

The proxy intercepts the user's activities and records the corresponding data points before the actions are transferred to the source software. Data returned from the application can also be intercepted and modified before it is returned to the user. User interaction data and corresponding application data can be stored in a log file.

Interaction recording tools generally record screen video and microphone audio to record what happened and when. The resulting video and audio provide context and detail not possible with simple proxies. Besides being able to track that the user has been working with a word processor for example, this tool can also track what the user has typed and why. Some usability monitoring tools use a combination of these two categories, which provides a lot of data that can be put to good use in research.

3. Implementation of the project

For the implementation of the project, an application was used in a real environment, with real users. Since this application is protected, the application screens will not be shown below. Its main purpose is bookkeeping services, where there are several modules that have been used within the application. The application, as well as additional metrics, were created in Visual Basic 6. The goal of the implementation was to monitor user activities and perform analysis based on that data. A code will be inserted into the application that will monitor and save all user actions during the session in the log file.

3.1 Activities being measured

There are several activities that will be monitored within the project:

- Applications and documents used by an individual user
- Reports used and their frequency of use. According to the analysis, optimization of the number of reports or possible additional training will be carried out
- Application usage time per user (total or effective)
- Entering a new syllable, changing the existing content, deleting a syllable, etc.
- The workplace from which the program is used
- Using the keyboard/mouse when calling commands/commands

3.2 Scope of automatic evaluation

During the implementation of the project, 8 users within one company were monitored. Within the application there are 57 forms, 78 tables and 155 reports. Given that all users log in with the same data, the results that will be obtained will refer to the company and not to an individual user. Activities within the following ten modules were monitored: production, sales, wholesale, finance, salaries, open/closed items, invoicing, BII/BOI, material and cash register.

3.3 Log file syntax

A code was inserted into the application that was used to save all user actions during the session in a log file. Each session, access to the application, is defined by one log file. The name of the log file is defined as follows:

<Workstation><Date><Time>.log.

Recording of log files is possible on the server as well as on the given workstation. If network resources are tested, it is preferable to record log files on a given workstation. The log file is defined with the following syntax:

Date(dd.MM.yyyy);Time(hh:mm:ss)<Command>optional<Number1><Number2><Description>

Date – the date when the action was performed

Time - the time when the action was performed

Command - which command was executed (new record, recording, mouse click, etc.)

Number 1 – number of the table or form (it can be as a parameter, but it doesn't have to be)

Number 2 – which key ascii command was executed/Id record number/link to the table (can be as a parameter, but it doesn't have to be)

Description – describes the command, e.g. username of the user/SQL query in the event of an error within the application/Name of the application/filter (it can be as a parameter, but it doesn't have to be)

Table 2 Log file syntax

Command	Command description	Number 1	Number 2	Description
The beginning	Getting started	-	-	User (username)
End	End of work	-	-	-
G	SQL query error	-	-	SQL query
I	Call for reports	-	-	The name of the report
M	Mouse click	-	-	-
K	Command	Table number	Key/Ascii commands	-
N	Command	Table number	Key/Ascii commands	-
SN	Record the input of the new syllable	Table number	Record ID number	-
SA	Save the change	Table number	Record ID number	-
A	Application selection	-	-	Application name
T	Document form invitation	Table number	Table link	Filter
Z	Form closure	Number of the table form	-	-

Given that 443 log files were created during testing, it could not fit in one picture, so the list of only a few log files will be shown below. The analysis was carried out from the beginning of January to the end of August 2020.

RadnaStanica1 06-01-2020 11-42-50	06.01.2020 11:43	Text Document	1 KB
RadnaStanica1 06-01-2020 11-44-23	06.01.2020 11:44	Text Document	1 KB
RadnaStanica1 06-01-2020 13-17-19	06.01.2020 15:54	Text Document	3 KB
RadnaStanica1 07-01-2020 10-56-57	07.01.2020 11:00	Text Document	1 KB
RadnaStanica1 07-01-2020 12-08-31	07.01.2020 12:41	Text Document	2 KB
RadnaStanica1 07-01-2020 15-17-20	07.01.2020 15:21	Text Document	1 KB
RadnaStanica1 08-01-2020 08-21-44	08.01.2020 10:25	Text Document	4 KB
RadnaStanica1 08-01-2020 10-25-45	08.01.2020 10:46	Text Document	2 KB
RadnaStanica1 08-01-2020 11-51-44	08.01.2020 11:53	Text Document	2 KB
RadnaStanica1 08-01-2020 13-39-33	08.01.2020 15:16	Text Document	1 KB
RadnaStanica1 09-01-2020 08-40-52	09.01.2020 09:23	Text Document	11 KB
RadnaStanica1 09-01-2020 09-32-50	09.01.2020 09:59	Text Document	2 KB
RadnaStanica1 9-1-2020 9-59-00	09.01.2020 10:03	Text Document	1 KB
RadnaStanica1 09-01-2020 10-13-43	09.01.2020 10:22	Text Document	2 KB
RadnaStanica1 09-01-2020 10-22-44	09.01.2020 10:24	Text Document	1 KB
RadnaStanica1 9-1-2020 10-13-59	09.01.2020 10:27	Text Document	3 KB
RadnaStanica1 9-1-2020 10-41-59	09.01.2020 10:48	Text Document	1 KB
RadnaStanica1 09-01-2020 10-24-15	09.01.2020 11:29	Text Document	6 KB
RadnaStanica1 10-01-2020 09-31-47	10.01.2020 09:58	Text Document	4 KB
RadnaStanica1 10-01-2020 11-11-03	10.01.2020 11:11	Text Document	0 KB
RadnaStanica1 10-01-2020 11-11-11	10.01.2020 12:05	Text Document	7 KB
RadnaStanica1 10-01-2020 13-13-58	10.01.2020 15:16	Text Document	1 KB

Figure 2 List of log files

The following figure shows what one of the log files looks like.

```

RadnaStanica1 28-04-2020 09-13-23 - Notepad
File Edit Format View Help
28.04.2020;09:13:27;admin;Početak
28.04.2020;09:13:28;A;FINANSIJE
28.04.2020;09:13:33;A;PLATE
28.04.2020;09:13:35;T;15;51;
28.04.2020;09:13:41;K;51;45
28.04.2020;09:13:48;K;51;45
28.04.2020;09:14:13;SN;51;202
28.04.2020;09:14:17;M
28.04.2020;09:14:17;K;51;113
28.04.2020;09:14:22;M
28.04.2020;09:14:22;K;51;118
28.04.2020;09:23:04;Z;51
28.04.2020;09:23:06;T;80;49;
28.04.2020;09:23:09;K;49;72
28.04.2020;09:23:18;K;49;17
28.04.2020;09:23:18;K;49;16
28.04.2020;09:26:33;Z;49
28.04.2020;09:26:34;T;15;51;
    
```

Figure 3 Example of log file content

For the purposes of analyzing the collected data, the mentioned log files were imported into two Access database tables "List of files" and "List of commands". The list of files has the attributes Number, FileName, Workstation, Date, Start, End and Status. The command list has the attributes FileNumber, ID, Date, Time, Command, Description, Table, IDNumber. On the basis of these tables, queries were created, with the help of which the necessary data for the diagrams shown in the images below were obtained. The purpose of this

application was to detect possible errors (date, time, user, workplace, SQL query or other command). A detailed analysis tries to reveal the cause of the error, be it an error in the code, network resources, error by the user, etc. In the next chapter, the diagrams and analysis of each of them are presented.

4. Discussion and results

After the data was collected in the log file, where we got a lot of useful information, their analysis was performed. After the analysis, the results are shown in the diagrams below. 57 forms, 78 tables, 155 reports and 10 modules were analyzed.

4.1 Visualization of results

It was analyzed how much the keyboard was used, and how much the mouse was used when calling commands/commands, which is shown in the following picture. The goal was to see if users save time and how efficient they are by using keyboard shortcuts.

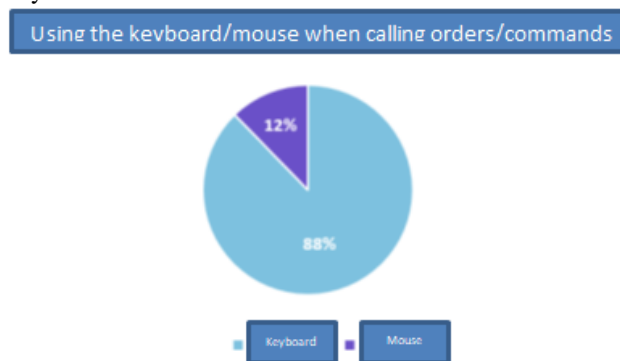


Figure 4 Keyboard/mouse usage

From the attachment, we can see that users have well adopted the use of abbreviations, they are maximally efficient in their work, they save time and no additional training is required for using the application. In other companies, where this project was implemented, it was observed that the users did not adopt the use of abbreviations, and they were shown how to save their time and increase their efficiency. Consultations revealed that these were new users, so they were given additional training.

The following image shows how users closed the program. There are three options, namely correct termination, where after successful operation the program is closed regularly, then incorrect termination, which includes possible scenarios: power outage, network failure or violent termination of the program. The last option is to cancel the call of the program, which implies opening the application and at the same time canceling the continuation of work.

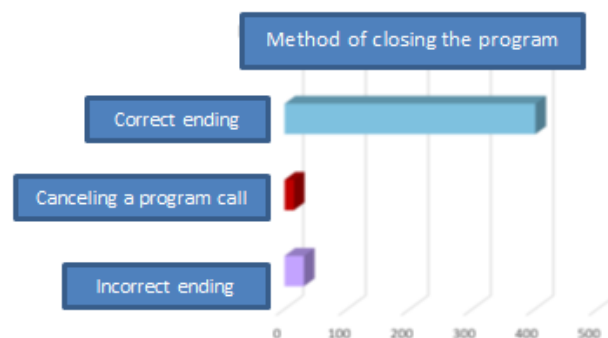


Figure 5 How to close the program

We obtained that in 90% of cases there was a correct termination, while 3% gave up on calling the program and 7% was an incorrect termination. After the analysis, it was determined that the incorrect endings were the result of a power outage. It is suggested that all equipment (server, switch, workstations) be provided with quality UPS devices. The following figure shows the user's actions in percentages, that is, the operations performed by the user.

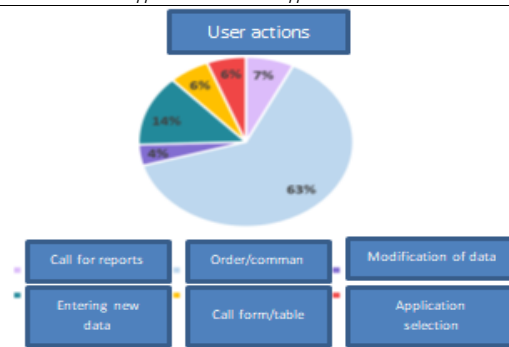


Figure 6 User actions

The following figure shows the number of commands per day of the week.



Figure 7 User activity by day

The most intensive days of the week are Thursday and Friday, which proved to be identical in other companies as well. The least amount of work was done on the weekend, given that they are non-working days, but sometimes you had to come for urgent interventions. The following figure shows the number of commands in a time period by hours.



Figure 8 Number of commands in the time period

We can see that the user's working hours are from 7:00 AM to 3:00 PM, and the activity shown at 11:00 PM was an emergency intervention. Effectiveness and efficiency were highest in the period from 09:00 to 11:00, while the least amount of work was done in the first and last two hours of working hours.

The following picture shows activities by module. There are ten modules in the application that was tested in this company, namely production, sales, wholesale, finance, payroll, open/closed items, invoicing, BII/BOI, material and cash register.

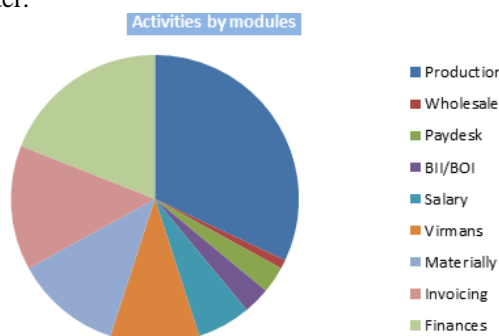


Figure 9 Activities by modules

Given that the main activity of the company in which we tested the project is the production of water heaters, it was to be expected that the production module would be used the most (32%). In the case of companies dealing with representation and distribution, the largest percentage was the wholesale and invoicing modules. Given that the application is quite automated (automatic posting to finance, BII/BOI), the expressed percentages of using the finance and BII/BOI modules are lower than expected. A true analysis of the use of reports is only possible by comparing the results obtained within a group of companies with the same activity. It was interesting that out of 155 reports, users used 61 reports, which requires additional analysis of several similar companies in order to reduce the number of unnecessary reports or to point out to users existing reports that are not used and are essential for business analysis.

In the above diagrams and displays of collected data, it is not possible to see the "G" command, which represents an error. During the test application of the mentioned project in 2019, two errors in SQL queries were observed, which were immediately corrected. Given that I took the analysis for work in the period from January to August 2020, these errors were not present in the analysis.

5. Conclusion

In this paper, in the theoretical part, a detailed consideration of the importance of usability and usability evaluation is given. Special attention is paid to automatic evaluation, which has a large focus in education and business, and is not a frequent topic in the academic environment due to the significant amount of work required to implement it, as well as the follow-up time required to reach relevant results. The theoretical part was demonstrated through the practical part of the work, where metrics were incorporated into the application used in a real business environment, logs were created, results were visualized, which were collected through automatic measurement. The basis for a serious and useful analysis of the use of a particular application has been created. Based on the data provided, additional and further processing can be done, and more metrics can also be added.

After analyzing the results, it is planned to expand the syntax of the log files with at least three additional commands to be followed, in addition to the commands listed in the paper. The significance of what was done for the business environment in which the practical part of the work was implemented can also be seen in the fact that in the coming period it is planned when a new version of the application for users will be implemented that includes new legal provisions, along with updating the application, future implementation will be supported by the project that was done in this paper. It is expected that the analysis of the collected data will facilitate and speed up the successful use of the new version.

References

- [1] Lazar J, Feng JH, Hochheiser H. Research methods in human-computer interaction. Morgan Kaufmann; 2017 April 28
- [2] Au, Fiora TW, Simon Baker, Ian Warren, and Gillian Dobbie. "Automated usability testing framework." In Proceedings of the ninth conference on Australasian user interface-Volume 76, pp. 55-64. 2008.
- [3] Niranjnamurthy, M., Archikam Nagaraj, Himaja Gattu, and Puneeth K. Shetty. "Research Study on Importance of Usability Testing/User Experience (UX) Testing." International Journal of Computer Science and Mobile Computing, IJCSMC 3, no. 10 (2014): 78-85.
- [4] Genise, Pauline. "Usability Evaluation: Methods and Techniques: Version 2.0" August 28, 2002. University of Texas
- [5] Preeti Sharma, „Advantage and Disadvantage of Automation Software Testing“, mart 2017, online: <http://paasis.com/advantage-disadvantage-automation-software-testing/>
- [6] Vipul Mishra, „User Testing v/s Usability Testing“, Medium, juni 2016, online: <https://medium.theuxblog.com/user-testing-v-s-usability-testing-c3a9edd04612>
- [7] Ashley Dotterweich, „What is Usability Testing“, Rainforest, august 2016, online: <https://www.rainforestqa.com/blog/2016-08-31-what-is-usability-testing/>
- [8] Justin Mifsud, „Usability Metrics – A Guide To Quantify The Usability Of Any System“, juni 2015, online: <https://usabilitygeek.com/usability-metrics-a-guide-to-quantify-system-usability/>
- [9] Bertoa, Manuel F., and Antonio Vallecillo. "Usability measures for software components." IEEE Latin America Transactions 4, no. 2 (2006): 136-143
- [10] Danny, Patricia Franzreb, „Setting Standards for Usability Testing“, UX Booth, juni 2016, online: <https://www.uxbooth.com/articles/setting-standards-for-usability-testing/>
- [11] Kluth, Wolfgang, Karl-Heinz Krempels, and Christian Samsel. "Automated Usability Testing for Mobile Applications." In WEBIST (2), pp. 149-156. 2014
- [12] Jeff Sauro, „10 essential usability metrics“, Measuring U, novembar 2011, online: <https://measuringu.com/essential-metrics/>

- [13] Sugandha Lahoti,, A UX strategy is worthless without a solid usability test plan“, Packt, juni 2018, online:<https://hub.packtpub.com/a-ux-strategy-is-worthless-without-a-solid-usability-test-plan/>
- [14] Michaela Zanello, „Track These 4 Metrics When You’re A/B Testing“, Conversion Fatantics, novembar 2018, online: <https://conversionfanatics.com/track-4-metrics-youre-ab-testing/>
- [15] Marketing Optimizer,, Which Metrics You Must Track While A/B Testing“, online: <https://www.marketingoptimizer.com/blog/marketing-optimization/metrics-must-track-ab-testing/>
- [16] International Organization for Standardization, ISO/IEC 9126-1: Software engineering-product quality-part 1: Quality model. Geneva, Switzerland: International Organization for Standardization. 2001
- [17] Abran A, Khelifi A, Suryan W, Seffah A. Consolidating the ISO usability models. In Proceedings of 11th international software quality management conference 2003 Apr 23 (Vol. 2003, pp. 23-25)
- [18] International Organization for Standardization , ISO/IEC 25010, online: <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010?limit=3&limitstart=0>
- [19] International Organization for Standardization, „ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models“, online: <https://www.iso.org/standard/35733.html>
- [20] Philips M. Heuristic Analysis for UX – How to Run a Usability Evaluation [Internet]. Toptal Design Blog. , Designers, 2019, online: <https://www.toptal.com/designers/usability/usability-analysis-how-to-run-a-heuristic-evaluation>
- [21] Usability Testing, Product Plan, online: <https://www.productplan.com/glossary/usability-testing/>