Journal of Theoretical and Applied Information Technology
15th July 2019. Vol.97. No 13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AUTO-MEASURING USABILITY METHOD BASED ON RUNTIME USER'S BEHAVIOR: CASE STUDY FOR GOVERNMENTAL WEB-BASED INFORMATION SYSTEMS

ABDELRAHMAN OSMAN ELFAKI, ZAID BASSFAR
Faculty of Computers and Information Technology
University of Tabuk, Saudi Arabia
{a.elfaki; zbassfar}@ut.edu.sa

ABSTRACT

Usability testing is a key process in quality assurance of information systems. The traditional usability testing is based on questionnaire techniques which are expensive, time-consuming, and complete by human evaluators including end users opinion, relatively, any evaluation done by human is subjective and expensive. There are no standard usability values that match with all users, hence conduct usability evaluation for each project is recommended. The previous two facts are providing motivation for developing auto-measuring usability method based on runtime user's behavior. The proposed method is developed based on six metrics extracted from literature. Two software applications represent governmental web-based information system have developed and the six metrics are embedded in these two applications. An experiment has been conducted by using these two software applications. The results are analyzed by statistical methods; and the results prove the practicality and applicability of the proposed method.

Key Words: *Human Computer Interaction; User Interfaces; Information System*.

1. INTRODUCTION AND MOTIVATIONS

Currently, Saudi Arabia is one of the leading countries in the automation of government transactions. Almost, all government sites are now using, or on the way to use, web base information systems. Saudi vision 2030 is supporting totally movement to E-government by establishing transformation procedures and rules (Saudi National Transformation Program 2020). Hence, in the government sites, there is continuous improvement by adding new, replacing, or updating existing web based information systems as response for the vision of 2030. Quality is the most important aspect that must be considered in selecting web based information system, especially if the web based information system is candidate to serve in government site. The foregoing mentioned has motivated us to develop software metrics for ensuring the quality in governmental web based information systems. As quality has wide range of disciplines, and aspects, we have decided to focus on usability. According to [1] measuring usability of web-systems is a major factor in measuring the quality of these systems.

There are numerous numbers of academic and industrial works that have tackled usability of web applications which reflect the importance of measuring web based systems usability. Insfran and Fernandez [2] provided a systematic review of usability evaluation in web development. Their results prove that there is a critical need for new methods that consider users' perspectives. Fernandez et al. [3] analyzed about 2703 papers in usability of web applications. Fernandez's et al. [3] results have supported the need for customization of usability test to be adaptable with diversity of users' environments and behaviors and proved that values and significance of usability factors differ from place to place. On the other hand, according to Yusop et al [4] reporting usability defects can be a challenging task and is implemented based on questionnaires or direct evaluations which do not provide a comprehensive and reliable evaluation can help developers discover gaps that they have not been aware of. Grigera, et al. [5] stated that "usability assessment of web applications continues to be an expensive and often neglected practice". Kumar and Owston [6] developed an automated method for evaluating accessibility in e-learning systems. Their research proved that the automated evaluation tools and the accessibility guidelines that they are based on are not effective at identifying all potential barriers to accessibility. Results from [6] have proven that studying website accessibility without considering the usability is not enough to prove quality of a web-site. According to Paz and Pow-Sang [7] questionnaires or direct evaluations do not provide a real strong evaluation that reflect the real difficulties and problems the user are suffer from.

The previous research results show two important facts:

1) There are no standard usability values that match with all users. In other words, for instance, suppose the usability metric "Time to complete the task", could have two different acceptable values. For instance, if " Time to complete the task " in system A is less than 5 seconds then systems A is usable, while if " Time to complete the task " in system B is less than 10 seconds then system B is usable. Moreover, even in one system the acceptable value for one metric could be varying from one user to another. Hence conduct usability evaluation for each project is recommended. In this paper, a method based on automatic measuring of usability is introduced that could be used any time anywhere to provide usability measurement.

2) Conduct usability test based on questionnaires are expensive, time-consuming. Questionnaire should be prepared, distributed, evaluated, and analyzed. In addition, in questionnaire methods there is human bias ,i.e., dependent on user (some users find it difficult to express their feelings).

3) Usability is the most important fact in proving the quality of a web-system.

Recently, there is an obvious interesting in developing automated tools for evaluating usability. In fact, it is impossible or at least very difficult to develop fully automated usability tests. Hence, automated tools are intended to be used as a component in the comprehensive evaluation of web site usability [7, 8, 9]. Thus at the end, the web based information system is a web site; therefore the Human Computer Interaction (HCI) researchers and practitioners with software engineering researchers and practitioners consider all web-based applications as web site. Therefore, in usability engineering, any discussion related to web page, or web site easily means all type of web based information systems.

The tools for automated usability testing have been classified into two classes: guideline checkers and interaction analyzers. The guideline checkers are used for evaluating the usability through compare the products with usability guidelines while interaction analyzers are tools that works with system interfaces to evaluate user actions. Guideline checkers are aiming to evaluate software usability with respect to standard usability knowledge. Interaction analyzers tools are aiming to find and extract new usability issues. In this paper, we will follow the both two classes (guideline checkers and interaction checkers) in developing our metrics. In other words, besides evaluating the system's usability our proposed metrics could be used for evaluating the user behavior in general.

2. RELATED WORK

According to ISO-1998 [10].usability has three main pillars:

Effectiveness: measuring the accuracy and completeness in achieve specific goals. Metrics are related to number of errors.

Efficiency: assesses the resources that are used in the system. Metrics are related to the loading time, minimal memory load.

Satisfaction: assesses the user's comfortability during usage of the interface. Metrics are related to satisfaction rating, ease of learning, and error handling. In this research paper, we have followed usability definition as in [10].

In the following, the selected related works have been investigated and analyzed in order to extract the strengths and weakness of each work. The main aim of this section is to define the research gap and to highlight the problem. We have used Google scholar search engine, we believe that all works that are not cited in Google scholar either are not academic or of relatively with low quality. The papers in Google scholar are selected from different famous databases such as Scopus, World of Science and Science Direct. Therefore, our search is limited only to Google scholar. The search strings are: "usability measurement + web based", "usability evaluation + web based", "interface metrics + web site". The search is limited to 2000 only, i.e., from 2000 to 2017. The search with the previous string was resulting in 2520 papers. All the papers have been deeply investigated but only few of them have been selected to be highlighted here in this section. We have followed the following conditions as selection process for the papers:

- The title should have the word "usability" and any of these words: evaluation, measuring, measurement, analysis, test, testing, or metrics. This condition is related to the fact that the title should reflect the main contribution of the work.
- The work should deal with of usability evaluation of web based system and the work should provide a real solution and practical service.

The exclusion criteria are:

- Works that are focused on mobile applications. Although mobile applications are dealing with the information systems through web sites, but it has special needs which are not our concern. In fact, our focusing in governmental web-

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

based information systems which is provided in computers platform, hence, mobile applications are far from our concern.

- Works that are not focused only on usability alone. In other words, the works that mentioned usability combined with other domains.
- Works that do not provide validation for their results.

In the following, the selected works are discussed and investigated.

Ivory [11] built an empirical foundation for automated interface evaluation. Ivory used statistical models classified web pages in good, average, and poor pages. Ivory used a mixture of different methods: statistical, observation, simulation, and, automatic methods.

Braz et al. [12] suggested a relational model between security and usability and they defined nine criteria for usability that have been extracted from Seffah et al. [13], which are: efficiency, satisfaction, productivity, learnability, safety, trustfulness, accessibility, universality, and usefulness. In addition, [12] defined seven websites usability measurable criteria that are directly measurable via at least one specific metric. The seven websites usability measurable criteria are: minimal action, operability, loading time, security, privacy, resource safety, and minimal memory load. Table 1 shows the matching between nine criteria for usability and the seven measurable websites usability.

Table 1: Matching between nine criteria for usability and the seven measurable websites usability [12]

The nine usability criteria	Suitable websites usability measurable
Efficiency, satisfaction, and accessibility	minimal action, operability, privacy, loading time, minimal memory load
productivity	loading time
learnability	minimal action, minimal memory load
safety	security, resource safety
trustfulness	Operability, privacy, security
universality	loading time, minimal memory load
usefulness	Operability, privacy, security, loading time, minimal memory load

Barriocanal et al. [14] have used web page length and number of links to developed fuzzy sets rules to automated usability analysis. Figure 1 shows samples of these rules. Wong et al. [15] defined ten usability metrics for e-learning which are: 1) E-learning System Feedback, 2) Consistency, 3) Error Prevention, 4) Performance/Efficiency, 5) User like/dislike, 6) Error Recovery, 7) Cognitive load 8) Internationalization, 9) Privacy, 10) On-Line Help.

```
if reading-efficiency is low then
efficiency is low
if efficiency is low then
<<take the appropriate corrective or reporting
action>>
```

Figure 1: Sample of [16] rules

Chi et al. [16] developed an automated tool for analyzing the web systems navigation. Navigation is most concerning usability issue in web sites. In addition, [16] developed Web agents to predict the user traffic flow through a Web site to predict the success of each site. The successful website is a site that its users can reach a target position in a suitable time. Granić et al. [17] developed usability evaluation methodology for web-based educational systems. They have used the usability guidelines that are presented in Table 2. They mixed automation with questionnaire to extract their results. We will follow these guidelines in our automated measurement.

Table 2: The usability guidelines as in [17]

1	Design an effective home page that will establish the site identity and give a clear overview of the content.
2	Structure information hierarchically to be useful for user.
3	Use a consistent page layout and indicate similar concepts
4	Integrate the information across different media types
5	Use terminology familiar to the user
6	Design for recognition rather than recall - make actions and options visible do not rely on the user remembering information
7	Make a pleasing and minimalist design
8	Provide links on each page to a list of local content, a site map and home

Mariage et al. [18] developed a software tool for web site usability testing called WebMetro. The tool allows a designer to input usability guidelines and have the tool automatically evaluate a website. This tool is used for improving the website system development. The WebMetro tool is an instance for guideline checkers. Norman [19] developed automating usability test by dynamically adding items and modifying the form and the tasks. Norman [19] tracked the user's action and provides

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

modified interfaces based on usability results. work in [19] is limited to tracked the user's action only. Baker et al. [20] developed a software tool for automated usability testing called Handheld User Interface (HUI). HUI compare expected action sequence (EAS) for a particular task with actual action sequence (AAS) and the obtained results have been used to improve the usability performance. According to [10] we can consider that the tool in [20] has measured the effectiveness. Bangor et al. [21] developed an empirical evaluation of the system usability scale. To implement the evaluation they compared different tasks within the same interface, and compared iterative versions of the same system. Oztekin et al. [22] developed an assessment methodology for usability of web-based information systems. They are used seven metrics which are: reliability, assurance, responsiveness, integration of communication, navigation, controllability, and quality of information. Table 3 shows the measurement of the nine usability metrics as they proposed by [22]. Fatimah et al. [23] developed a web-based automated system for managing usability testing systematically. The response time is the main usability factor that has been considered. Fatimah et al.[23] ignored that response time is linked to correctness, i.e., response time for incorrect user's actions should be not considered. The work in [1] showed that the usability measurement as an important quality factor. The work in [1] uses the standard definition of usability: effectiveness, efficiency, satisfaction. He defined satisfaction as: likeability, pleasure, comfort, trust.

Table 3: The measurement of the nine usability metrics [22].

Metric	Description
Reliability	Possibility of replace or restore default setting easy
Assurance	Existence of security statements and feedback notification
Responsiveness	Existence of help and task Reponses
Integration of communication	Using user technical language
Navigation	Tracking of user position
Controllability	Providing facility to reach any position at any time
Quality of information	Providing correct timely, relevant, and accurate information

Elfaki et al. [24] suggested using five metrics for measuring of e-learning usability through user interface. These metrics are: time of user feedback, average of using help methods, average of using undo, average time spent in any page, and average of using e-learning system's search engine. Elfaki et al [24] metrics are covered the three pillars of usability which are ease of use, ease to learning, and task matching. These three pillars of usability have been defined in [25]. These three pillars of usability have been defined as "satisfaction" in [10].
Ammar et al. [26] developed a model-driven approach for usability engineering of interactive systems. They used ISO/IEC 9126-1 usability standard: learnability, understandability, operability, attractiveness, and compliance.
Marenkov et al. [27] developed a framework including category specific metrics with methodology for immediate automatic usability evaluation of web application user interfaces during design and implementation phase. They used 117 usability guidelines covering different usability areas. Marenkov et al. [27] measurement has provided a direct link between user and developers, in such the user can immediately deliver their opinions, comments, or, suggestions. The framework in [27] is considered as online questionnaire. Yusop et al. [4] issued a research to understand usability defect reporting. Their results and analysis of 147 responses show that reporters often provide observed result, and expected result when describing usability defects. This research shows the critical need for automated the usability testing.
Takahashi [28] proposed method for usability investigation by measuring and describing a user's operation flow. The metrics that have been used in [28] are not mentioned clearly.
Grigera et al. [5] described usability problems on running web applications, and the process in which they can be identified by analyzing user interaction events. Table 4 shows the user interaction events that have been used to measure usability by Grigera et al. [5]. In fact, Grigera et al. [5] measured only the navigation which is only part of usability.

Table 4 shows the user interaction events [5]

Event ID	Event	Usability smell
A	Tooltip attempt	Undescriptive element
B	Click attempt	Unresponsive element
C	Flash scrolling	Overlocked content

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

D	Flash Navigation	Misleading link
E	Navigation Path	Distant Content
F	Search	Useless search result
G	Option selection	Wrong default value
H	Form submission	Late validation
I	Unfilled Form	Abandoned form
J	Text input	Unformatted input
K	Long Request	No processing page

Hu et al. [29] developed formalization, i.e., mathematical representation for data-driven navigability which is divided to power, efficiency, and directness. Table 5 shows the definitions for navigability components according [29].

Table 5: the definitions for navigability components according to [29].

Power	the probability that a visitor accurately locates target information by navigating through a website's hyperlink structure
Efficiency	with which a visitor locates target information
Directness	with which a visitor can decide where to move from the current page to the target information

Navigability could be easy considered under effectiveness, and satisfaction. Hence definition of usability in [10] is a complete and accurate usability definition.

In the following we have highlighted the research gap and summarized the problem in points as a conclusion from the previous discussion:

- Usability measurement currently is done by developers themselves to improve their product. This is means the measurement is reflecting developer perspectives and ignoring the customer or end user perspectives, is not enough to depend on usability measurement that has been implementing in developers site or based on developers perspectives. Missing usability measurement that is done by end user is critical shortage in usability measurement.
- Measuring web information systems usability still needs to be improved due to difficulties in reporting usability problems [4]. Usability measurement is usually completed by human evaluators or experts and including end users opinion, i.e., it is human dependent. According to real life fact, relatively, any evaluation done by human is subjective and expensive.

The two problems above in the usability of web based information systems are general and it has been noticed and happened regardless of the place. In addition to these two problems, there is a problem related to the selection of a new government web based information system. For instance, if there are two web based information systems and there is only one system should be selected, and these both systems have passed the functionality test equally, then the decision will be very tough or will depend on nonscientific factor such as a price. In addition, It is clear that the research gap is lacking of open source method that is could be easily automatically measure user behavior at the runtime, and in same time is developed based on [10]. Table 6 shows the summary of related works. We have highlighted the limitation of each work in Table 6 and as a result, the research gap has been defined. The research gap is lacking of a usability measurement method that is fully automated and developed based on [10].

Table 6: summary of related works

Work	Measurement method	Metrics	Limitation
Ivory [11]	Measurement is done by mixing of statistical, observation, simulation, and, automatic methods	Not mentioned	Not fully automated
Braz et al. [12].	Measuring of efficiency, satisfaction, productivity, learnability, safety, trustfulness, accessibility, universality, and usefulness	Minimal action, operability, loading time, security, privacy, resource safety, and minimal memory load	Is not based on ISO-1998. Measuring of effectiveness (accuracy, and completeness) is not clear

Seffah et al. [13]	Questionnaires, observation, and from log files	127 metrics	Is not based on ISO-1998.
Barriocanal et al. [14]	Rule-based	Web page length and number of links	Is not based on ISO-1998.
Wong et al. [15]	Programming	1) E-learning System Feedback, 2) Consistency, 3) Error Prevention, 4) Performance/Efficiency, 5) User like/dislike, 6) Error Recovery, 7) Cognitive load 8) Internationalization, 9) Privacy, and 10) On-Line Help	Is not fully automated. Is ignoring accuracy, and completeness)
Chi et al. [16]	Web agents (program)	User traffic flow	Measuring only navigation
Granić et al. [17]	Mixed automation with questionnaire	Clear content, useful information, consistency, Integrate information, familiar terminology, recognition actions, minimalist design, and providing site map	Not fully automated
Mariage et al. [18]	Programming (guideline checker)	Users insert his own usability guidelines	Is not based on ISO-1998.
Norman [19]	Tracking the user’s action	Users’ actions	Is not based on ISO-1998. limited to tracked the user’s action only
Baker et al. [21]	Compare expected action sequence (EAS) for a particular task with actual action sequence (AAS)	Log file (user actions), and predefined list(expected actions)	Measuring only effectiveness
Oztekin et al. [22].	Programming	reliability, assurance, responsiveness, integration of communication, navigation, controllability, and quality of information	Is not based on ISO-1998.
Fatimah et al. [23]	programming	response time	Measuring only The response time
Elfaki et al. [24]	Programming	time of user feedback, average of using help methods, average of using undo, average time spent in any page, and average of using e-learning system’s search engine	Especially for ELearning only. Is not based on ISO-1998.
Ammar et al. [26]	Programming, model driven approach	learnability, understandability, operability, attractiveness, and compliance	Measuring only interactivity
Marenkov et al. [27]	Programming	117 usability guidelines	Considered as online questionnaire

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Yusop et al. [4]	Questionnaire	Not clear	Not automated
Takahashi [28]	Describing a user's operation flow	Not clear	Is not based on ISO-1998.
Grigera et al.[5]	Analyzing user interaction events	Click attempt, flash scrolling, flash Navigation, navigation Path , search, option selectionm and Form submission.	Is not based on ISO-1998.
Hu et al. [29]	Mathematical representation	Power, efficiency, and directness	Measuring only navigation

3. SOFTWARE METRICS FOR AUTO-MEASURING USABILITY

In this section, the metrics that have been used to develop our auto-measuring usability method have been defined. According to [10] usability has three components: effectiveness, efficiency, and satisfaction. For each one of these components we have defined software metrics that can be calculated automatically. These software metrics have been extracted from related works which have been discussed in section 2. Table 7 shows usability component and its association metrics. In Table 7, the first column demonstrates the usability component, the second column demonstrates the metrics that have been used to evaluate usability component, and column three demonstrates the description of each metric. These metrics have been selected from the previous works based on the following three categories:

- The metric should be mentioned in research work that has made an achievement in measuring usability, i.e., the work should be published and its contribution should be validated.
- The metrics should use to measure one of the usability components as described in [10]. In some papers, usability has been described by different components but still these components could be covered under ISO -1998. For instance, the work in [26] described usability as: learnability, understandability, operability, and attractiveness; these four components could easily be described under ISO-1998 effectiveness.
- The metric can be represented as a mathematical formula, as our target is developing automatic method by programming these metrics to work in the background and notices users' behaviors.

Table 7: usability component and its association metrics

Usability Component	Metric	Description
Effectiveness	No. of errors	Number of errors that have been made by a user to complete one specific task.
	Number of actions to complete the task	Number of actions that have been made by a user to complete one specific task.
	Accuracy	Proportion of accuracy in achieving a specific task. The accuracy has been calculated by comparing user's number of actions by predefined number which represents task's actions. There are three levels of accuracy: accurate, semi_accurate, and not accurate which is represented by 1, 2, and 3 respectively.
Efficiency	Time to complete the task	Time taken by a user to finish specific task.

Satisfaction	Number of using undo options	The number of times in which escape key, backspace key, esc key, delete key, or undo key has been used by a user in specific task.
	Number of using help methods	The number of times in which help methods (if any) has been used by a user in specific task.

The six metrics in Table 7 had been used in our proposed tool to auto-measure of run-time user's behavior to define the usability. According to [30] there are five conditions to develop useful and applicable software metric. In the following, the six selected metrics (from Table 7) have been explained based on the five conditions from [30] to prove the applicability and usefulness of the proposed metrics.

1. Objective Statement. This condition means that metric should provide information that help better understanding of software products. The proposed metrics, which are defined in table 4, column 2, provide an understanding of a system based on users' perspective.
2. Clear Definitions: This condition means that metrics should have a standard definition. The proposed metrics have been defined under ISO-1998 standard.
3. Define the Model: This condition means that metric should have a clear model of its usages. The proposed metrics have been defined under three main usability components which provide clear and complete model for usability.
4. Establish Counting Criteria: This condition means that metric should be obviously counted. Each proposed metric generates numeric value as had been discussed in table 7.
5. Decide what's "Good": This condition means that metric should be used to make a usability decision based on metric's result.

In the next section, the implementation and evaluation of the proposed metrics is presented.

4. IMPLEMENTATION AND EVALUATION

The methodology of developing our propose methods is the following steps: defining the research gap in the field of auto-measuring usability from literature, defining the best definition for usability, defining suitable metrics which together form a proposed method, conduct experiments by using the selected metrics to prove applicability of proposed method, and finally, evaluate the propose method.

To test the proposed method in real environment the following steps has been used:

I. Developing software application for testing the proposed method. The software had been developed internally by our Information Technology department. The proposed metrics have been added inside the code.
II. Preparing the usability test experiment;
III. Conducting the prepared experiment;

Below, details of each step has been explained.

Develop software application for testing the proposed method: In this step, we have collaborated with IT department to develop software for applying the proposed method. We have assigned one programming task to two different programming teams. The programming task is developing a stock control system. Developing steps of this software system are out of scope of this paper. At the end, we have two stock control systems with the proposed metrics embedded into them. We will entitle the first system "A" and the second system "B". The software metrics have been implemented by defining specific counter for each metric. For instance, the metric "time to complete the task" is implemented by a counter that defined the task time since its start until it is finished. Table 8 shows the technical design of methods that had been used in implementing the proposed metrics. Each metric have been implemented by a specific method with task number as a parameter. Task number is a unique number that is used to define the task. Table 9 shows the explanation of functions that are used in the technical design.

Table 8: technical design of the proposed metrics

Metric	Technical Design

No. of errors	int method_ NoOfError(int task)
	{
	int NoOfError = 0;
	While not (finish (task))
	{
	get (user_action);
	If not(member(user_action,correct_action_list(task)))
	NoOfError = NoOfError +1;
	}
	return (NoOfError);
	}

Number of actions to complete the task	int method_ NoOfActions(int task)
	{
	int NoOfActions = 0;
	While not (finish (task))
	{
	if (user_action) = true
	NoOfActions = NoOfActions + 1;
	}
	return(NoOfActions)
	}

Accuracy	int method_accuracy(int task)
	{
	int NoOfActions = 0; int flag = 0;
	NoOfActions = method_ NoOfActions(task)
	Flag = Abs(NoOfActions – PreNoOfActions(task));
	if (flag >= 0 and flag <= 2)
	return (1);
	if (flag >2 and flag <= 5)
	return (2);
	if (flag >5)
	return (3);
	}

Time to complete the task	int method_TaskTime(int task)
	{
	int t1= 0;
	While not (finish (task))
	{
	t1 = t1 + to_int(get(time));
	}
	return (t1);
	}

Number of using help methods	int method_use_help(int task)
	{
	int NoUseHelp = 0;
	int current_ button ;
	While not (finish (task))
	{
	current_ button = get (button_press);
	if ((current_ button) == button(help)) or (current_ button) == F1)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	NoUseHelp = NoUseHelp +1; } return (NoUseHelp); }

Table 9: explanation of technical design functions

function	Explanation
finish (task)	Function is used to check the task status. It will return true if the task has been completed.
member(user_action,correct_action_list(task))	Function is returned true if the user action (first parameter) is member in the list of correct actions that describe the task (second parameter).
Abs(NoOfActions – PreNoOfActions(task))	Function returns absolute value of the difference between number of actions done by user and a predefined number represents the ideal number of actions that should be taken to implement the action.
to_int()	Function converts the parameter to integer number.
get(time)	Function used to return the current time according to the server.
get (button_press)	Function is used to return the current selected button
button(help)	Function return true is help button is pressed

Preparing the usability test experiment

Research design is an important step in all research studies. In this research, experiments have been used as a research design methodology. Seltman [31] saw experimental design as a careful balancing of several features including power, generalizability, validity, practicality and cost. Seidenfeld [32] argued that proper experimental design is needed to ensure that questions of interest can be answered and that this can be done accurately, given experimental constraints (such as cost). An experimental study is a scientific test that investigates the relationship between an outcome and one or more conditions manipulated by the researcher.

The Experimental Design: The target of the experiment is testing the usability by collecting users' actions that represent runtime user's behavior. In the following, the experimental design has been explained:

- Select employees for implementing the experiment. Five employees from stock department have been selected to conduct the experiment. Those that have good experiences in computer applications have been selected. These five employees having equal expertise in doing stock control function by using computer systems.
- Preparing the two developed applications by adding the proposed metrics embedded into it.
- Prepare the applications to work in client-server environment to enable users to access the applications remotely. By adopting a client-server service, the application will be hosted in a server which all logs and user activity time stamps can be recorded based on the server clock. User activities such as clicking on a function/button or moving mouse was captured and recorded along with the time invoked from these activities. These records were kept in a table in the MySQL database. The applications were built using Java, PHP and MySQL with free license. The applications interface size could be expanded based on the user's web browser size where the size is controlled using Java Script. Login page is created using PHP to allow access only to authorized users.
- Experiment tasks: According to [33] experimental task should have three properties which are: matching the user's goal, formal representation, and actionable. Our experiment task is reflecting these properties very obviously. Table 10 shows example of experiment tasks. The number of actions represents number of suitable predefined actions to execute the task. For instance, the task "open new store" has the following actions: 1) select the control menu, 2) select new store, 3) insert store name, 4) insert address, 5) define capacity, 6) insert registration date, insert store keeper (steps 3 to 7 are representing filling the new store page), and 8) press submit.

Table 10: Example of Experiment Tasks

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

#	Task	No of Actions
1	Open new store (M)	8
2	Register new item (X)	9
3	Add new comer of item (X) in store (M). Quantity= 100	7
4	Register new item (Y)	9
5	Add new comer of item (Y) in store (M). Quantity= 100	7
6	Open new store (N)	8
6	Add new comer of item (X) in store (N). Quantity= 50	7
7	Add new comer of item (Y) in store (N). Quantity= 70	7
8	Drop item (X) from store(M); Quantity= 20	9
9	Drop item (Y) from store(M); Quantity= 30	9

Conducting the prepared experiment
The experiment was conducted during a length of two days. On the first day, the group (five employees) has been asked to do tasks (similar to in Table 8) by using application A. Then, after two hours the group of five employees have been asked to do different tasks, but in the same context, by using application B. In the second day, two different sets of tasks have been provided, but still in same context. First, the employees applied first set of tasks by using application B then after two hours they applied the second set of tasks by using application A.
All of the four sets of tasks have the same qualitative and quantitative attributes. Table 6 shows one set of these four sets. The experiment has been conducted in five offices. Each experiment's employee has its own office. Each office is completely isolated from the other offices.

5. RESULTS AND DISCUSSION

The statistical package SPSS was employed to analyzed results; were the mean and standard deviation had been hired. Mean and Standard deviation are a statistical measurement of the variation in a set of data, which indicate how much the values of a certain data set differ from the mean on average. This reflect significant differences between the two sets. Results of analysis of the difference in means and ANOVA test analysis of variance were compared to investigate usability performance of participants being given the chance to apply four set of tasks to two applications: A and B.

Number of errors: The mean scores for the "number of errors" of the two applications A, and B reflect that; significant differences were recorded as (49.88 and 47.16) respectively. There was a progressive decrease in number of errors orientation ratings from application A to application B. Considering the standard deviations it was evident that the variance of scores application B(28.7) is relatively the smallest compared with application A (29.0).

Number of actions to complete the task: The mean scores for the "number of actions to complete the task" of the two applications A, and B reflect that; significant differences were recorded as (42.40, and 39.22) respectively. There was a progressive decrease in number of actions orientation ratings from applications A to B. Considering the standard deviations it was evident that the variance of scores in the application B (23.2) is relatively the smallest compared with application A (23.4).

Accuracy: The mean scores for the "accuracy" of the two applications A, and B reflect that; significant differences were recorded as (50.66%, and 54.96%) respectively. There was a progressive increase in accuracy orientation ratings from applications A to B. Considering the standard deviations it was evident that the variance of scores in the application B (14.6%) is relatively better to compare with application A (19.1%).

Time to complete the task: The mean scores for the "time to complete the task" of the two applications A, and B reflect that; significant differences were recorded as (1:36:06.92, and 1:24:36.72) respectively. There was a progressive decrease in time to complete the task orientation ratings from applications A to B. Considering the standard deviations it was evident that the variance of scores in the application B (0:19:47.678) is relatively less than results of application A (0:21:21.820).

Number of using undo options:
The mean scores for the "Number of using undo options" of the two applications A, and B reflect that; significant differences were recorded as (65.44, and 57.28) respectively. There was a progressive decrease in number of using undo operations orientation ratings from applications A to B. Considering the standard deviations it was

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

evident that the variance of scores in the application B (34.3) is relatively the smallest compared with those of application A (35.7).

Number of using help methods:
The mean scores for the "number of using help methods" of the two applications A, and B reflect that; significant differences were recorded as (118.46, and 112.08) respectively. There was a progressive decrease in number of using help methods; ratings from applications A to B. Considering the standard deviations it was evident that the variance of scores in the B (58.6) is relatively the smallest compared with A (59.5). Figure 2 show results comparison of the application A, and B.

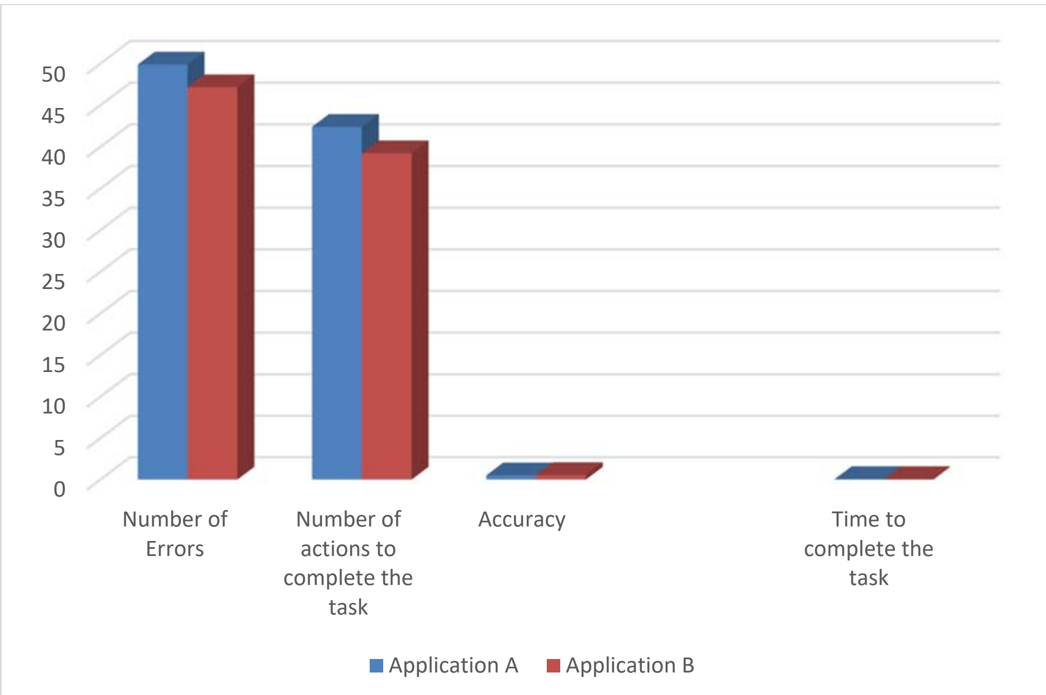

Figure 2: Comparison between applications "A" and "B"

6. CONCLUSION

In this paper, an empirical method for auto-measuring usability method based on runtime user's behavior has been introduced. The proposed method is useful to provide an independent factor to compare between software applications which could be used for supporting the decision maker to select the appropriate software application. We believe that the results will vary from place to place even in one place it will vary from time to time. The results are completely dependent on user's behavior at runtime.

According to [34] good experiments in human computer interaction should be explained by statistical analysis, hence we have followed this advice and the results have been explained by two statistical methods: mean and standard deviations. The methodology of developing our propose methods is the following steps: defining the research gap in the field of auto-measuring usability from literature, defining the best definition for usability, defining suitable metrics which together form a proposed method, conduct experiments by using the selected metrics to prove applicability of proposed method, and finally, evaluate the propose method.

The contribution of this paper could be summarized in two points: the first contribution is defining metrics that could be calculated automatically. These metrics are extracted based on ISO-1998 definition of usability. These metrics are covered all the three pillars of usability (effectiveness, efficiency, and satisfaction). The practically of these metrics has been proved in literature, as discussed in related work. The second contribution is providing applicable method with full technical details for auto-measuring usability method based on runtime user's behavior. All traditional methods of testing usability are based on questionnaires. Questionnaires' methods have two weakness:1) are expensive as questionnaire should be distributed, evaluated, and analyzed whenever there is a need for measuring usability, on the contrary, our proposed is developed only once and could provide results whenever there is a need without any additional effort. 2) Questionnaires' methods are human bias but the proposed method is fully clear from human bias. Moreover, the proposed methods could be used to compare between two or more software applications, in fact, there is no limit for number of applications in a comparison process, in contradictory of traditional methods that are compared one or two application.

The contributions have been proved by empirical experiments and the results is proved statistically. In addition, this paper has presented technical details that are missed in many related works; these technical details provide a concrete foundation for future empirical studies.

The main contribution of this paper is to provide open source method for auto-measuring usability of information systems. Our method has been described technically with ful details, which make it easy for any developer to conduct and develop this method in his application. The finding of this paper proved that our proposed method is applicable and could be used to auto-measure usability of any information system.

ACKNOWLEDGEMENT

The authors are thankful to the Deanship of Scientific Research (DSR), the University of Tabuk, Saudi Arabia, for financial support under the grant number S-1438-0211.

REFERENCES

[1] Bevan, N. (2012). Extending quality in use to provide a framework for usability measurement. Proceedings of HCI International 2009, San Diego, California, USA.

[2] Insfran, E., Fernandez, A. (2008). A Systematic Review of Usability Evaluation in Web Development. Web Information Systems Engineering – WISE 2008 Workshops Volume 5176 of the series Lecture Notes in Computer Science pp 81-91.

[3] Fernandez, A., Insfran, E., Abrahão, S. (2011). Usability Evaluation Methods for the Web: A Systematic Mapping Study. Journal of information and Software Technology, Volume 53 Issue 2

[4] Yusop, NS. M., Grundy, J., Vasa, R. (2016). Reporting Usability Defects – Do Reporters Report What Software Developers Need? EASE '16 Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering, Limerick, Ireland.

[5] Grigera, J., Garrido, A., Rivero, M,J., Rossi, G. (2017). Automatic detection of usability smells in Web Applications. International Journal of Human-Computer Studies Volume 97, January 2017, Pages 129–148.

[6] Kumar, K.L.,Owston, R. (2016). Evaluating e-learning accessibility by automated and student-centered methods. Education Tech Research Dev (2016) 64:263–283.

[7] Paz, F., Pow-Sang. J.A. (2016). A Systematic Mapping Review of Usability Evaluation Methods for Software Development Process. International Journal of Software Engineering and Its Applications Vol. 10, No. 1 (2016), pp. 165-178.

[8] Kaur, S., Kaur, K., Kaur, P. (2016). Analysis of Website Usability Evaluation Methods", In INDIACom -2016; ISSN 0973-7529; ISBN 978 -93-80544-20-5.

[9] Montero, F., González, P., Lozano, M., Vanderdonckt, J.(2014). Quality Models for Automated Evaluation of Web Sites Usability and Accessibility, Advances in Human-Computer Interaction Volume 2014 (2014), Article ID 479286, 13 page.

[10] ISO, (1998). "Ergonomic Requirements for Office Work with Visual Display Terminals (vdts), -- Part 11: Guidance of Usability", International Organization for Standardization, Geneva, Switzerland, ISO 9241-11:1998.

[11] Ivory, M.Y. (2001). An Empirical Foundation for Automated Web Interface Evaluation. A dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Computer Science UNIVERSITY of CALIFORNIA at BERKELEY. USA.

[12] Braz, C., Seffah, A., M'Raihi, D. (2007). Designing a Trade-Off Between Usability

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and Security: A Metrics Based-Model. Book chapter in C. Baranauskas et al. (Eds.): INTERACT 2007, LNCS 4663, Part II, pp. 114 – 126.

[13] Seffah, A., Donyaee,M., RB Kline, RB, Padda HK.(2006). Usability measurement and metrics: A consolidated model. Software Quality Journal 14 (2), 159-178.

[14]Barriocanal, G. E., Urbán, M., Gutiérrez,J.A. (2008) On the vague modelling of web page characteristics regarding usability. AWIC'03 Proceedings of the 1st international Atlantic web intelligence conference on Advances in web intelligence, Pages 199-20. Spain.

[15] Wong, S., Nguyen, T.T., Chang, E., Jayaratna, N. (2003). Usability Metrics for E-learning. R. eersman and Z. Tari (Eds.): OTM Workshops 2003, LNCS 2889, pp. 235–252, © Springer-Verlag Berlin Heidelberg.

[16] Chi, E.H, Rosien, A., Supattanasiri, G., Williams, A., Royer, C., Chow, C., Robles, E., Dalal, B., Chen, J., Steve Cousins, S. (2003). The Bloodhound Project: Automating Discovery of Web Usability Issues using the InfoScent Simulator, CHI 2003 Conference on Human Factors in Computing Systems, Fort Lauderdale, Florida, USA.

[17] Granić, A., Glavinić, V., Stankov, S. (2004). Usability Evaluation Methodology for Web-based Educational Systems. 8th ERCIM Workshop "User Interfaces for All" – Workshop Adjunct Proceedings. Stary, Christian; Stephanidis, Constantine (Eds.). Heraklion (Crete), Greece: ERCIM - The European Research Consortium for Informatics and Mathematics,. 28.1-28.15.

[18] Mariage, C., Vanderdonckt, J. and Chevalier. (2005). Using the MetroWeb Tool to Improve Usability Quality of Web Sites. In proceedings of the IEEE Latin American Web conference, Buenos Aires, Argentina.

[19] Norman, K.L., (2006). Levels of Automation and User Participation in Usability Testing. Journal Interacting with Computers archive Volume 18 Issue 2, Pages 246-264.

[20] Baker, S., Au, F., Dobbie, G., Warren,I.(2008). Automated Usability Testing Using HUI Analyzer, 19th Australian Conference on Software Engineering, Australia.

[21] Bangor, A., Kortum, James T. Miller, J.T. (2008). An Empirical Evaluation of the System Usability Scale. International Journal of Human–Computer Interaction Volume 24, 2008 - Issue 6, pages 574-594.

[22] Oztekin, A., Nikov, A., Zaim, S. (2009). UWIS: An assessment methodology for usability of web-based information systems. The Journal of Systems and Software 82 (2009) 2038–2050.

[23] Fatimah, W. A., Sulaiman,S., Johari, F.S. (2010). Usability Management System (USEMATE): A Web-Based Automated System for Managing Usability Testing Systematically. International Conference on User Science and Engineering, Shah Alam, Malaysia.

[24] Elfaki, A., Bachok, R., Duan, Y., Du, W., Md Johar, M.G., Fong, S. (2013). Towards measuring of E-Learning Usability through User Interface. International Conference on Advanced Applied Informatics (IIAI AAI 2013). Matsue, Japan.

[25] Gonzalez, C. (2012). Student Usability in Educational Software and Games: Improving Experiences, IGI, USA.

[26] Ammar, L., Trabelsi, A., Mahfoudhi, A. (2016) A model-driven approach for usability engineering of interactive systems. Journal of Software Quality (2016) 24:301–335.

[27]Marenkov, J. Robal, T., Kalja, A.(2016). A Study on Immediate Automatic Usability Evaluation of Web Application User Interfaces, book chapter in G. Arnicans et al. (Eds.): DB&IS 2016, CCIS 615, pp. 257–271.

[28]Takahashi, A. (2016). A "User-Flow Description" Method for Usability Investigation. Chapter at M. Kurosu (Ed.): HCI 2016, Part I, LNCS 9731, pp. 340–349, 2016.

[29] Hu, P.JH., Hu, HF., Fang, X. (2017). Examining the mediating roles of cognitive load and performance outcomes in user satisfaction with a website: A field quasi-experiment, MIS Quarterly Vol. 41 No. 3.

[30] Westfall, L. (2005). 12 Steps to Useful Software Metrics. Technical Report. The Westfall Team, 2005 - win.tue.nl.

[31]Seltman, Howard J. (2015). Experimental Design and Analysis. Publisher: Carnegie Mellon University. Number of pages: 428.

[32]Seidenfeld, T. (1992) R. A. Fisher on the Design of Experiments and Statistical Estimation. In: Sarkar S. (eds) The Founders of Evolutionary Genetics. Boston Studies in the Philosophy of Science, vol 142. Springer, Dordrecht.

[33] Crystal, A., Ellington, B (2004). Task analysis and human-computer interaction: approaches, techniques, and levels of analysis Proceedings of the Tenth Americas Conference on Information Systems, New York,USA,

[34] Cairns, P. A. (2016). Experimental methods in Human-Computer Interaction. In M. Soedergaard, & R. Dam (Eds.), *Encylopedia of Human-Computer Interaction, 2nd edn* [34] Interaction Design Foundation.