International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869, Volume-2, Issue-2, February 2014

USABILITY EVALUATION OF OBJECT ORIENTED APPLICATIONS USING LEARNING CURVES

Dr.K. Alagarsamy and R. Madhavan

***Abstract*— Usability is an important factor for all software quality models for the development of successful interactive software applications. The methods are chosen for usability evaluation on the basis of available resources, abilities of evaluator, types of users and environment. Due to such wide importance of this quality factor various usability evaluation methods are proposed by usability experts and researchers. Parameters of usability evaluation like completion rate, task time, no of errors, expectation, experience etc. are used to test the usability nature of the software. In this paper, we have analyzed various applications based on the various parameters explained above to test the usability and we have implemented through learning curve. Based on the above analysis we can explain the usableness and perfectness of the application in a suitable environment.**

***Index Terms*— Task time, no of errors, completion rate**

I. Introduction

Usability is defined as 'the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component. Usability correlates with the functionality of the system and helps in its evaluation. The lack of usability causes failure of the software system that leads to a substantial monetary loss, user dissatisfaction, staff unproductivity and time wastage. Therefore, usability evaluation is very important for the process of designing usable software system. But still there are no apt criteria or models for usability evaluation because of its fuzzy characteristics.

Views of Usability

In the early days, usability was described as any application designed for people to use should be easy to learn (and remember), useful, that is, contain functions people really need in their work, and be easy and pleasant to use. Also particular testable aspects of usability are identified, which are expressed in terms of the following concepts:

Learnability: The time and effort required to reach a specified level of use performance (also described as ease of learning).

Throughput: The tasks accomplished by experienced users, the speed of task execution and the errors made (also described as ease of use).

Manuscript received Feb. 09, 2014.

Flexibility: The extent to which the system can accommodate changes to the tasks and environments beyond those first specified.

Attitude: The positive attitude engendered in users by the application.

Efficiency: How easy is the product to use and be productive?

Error tolerance: Do users make few errors? Are errors recoverable?

Relevance: Does the product meet users real needs?

Accessibility: Does the product support the usage needs of all potential users including those with special physical requirements?

Modern Views of Usability

Usability was described more often in terms of quantitative performance metrics than in terms of subjective or qualitative usability goals such as user enjoyment. At that point in time, there was a great expectation for computers to be useful for supporting humans in their work, for lightening their workload, or even perhaps, in some cases, for computers to replace human workers for certain tasks.

The focus for usability has moved beyond just the usability of work-based systems to the usability of all computer-based systems with which a human interacts. Whereas there is still a focus on measurable outcomes such as effectiveness and efficiency, the user's experience in interacting with a system has taken on a great level of importance.

To establish the connection between usability and the user experience, a framework has proposed for usability comprised of five dimensions, referred to as the five Es. These dimensions — effective, efficient, engaging, error tolerant, and easy to learn — each describe an aspect of the user experience. The five Es build on ISO 9241's three characteristics of usability (efficient, effective, and satisfying, which becomes engaging in framework), plus two other dimensions of error tolerant and easy to learn. these dimensions are as follows:

Effective: The completeness and accuracy with which users achieve their goals.

Efficient: The speed (and accuracy) with which users can complete their tasks.

Engaging: The degree to which the tone and style of the interface makes the product pleasant or satisfying to use.

Error tolerant: How well the design prevents errors or helps with recovery from those that do occur.

Easy to learn: How well the product supports both initial orientation and deepening understanding of its capabilities.

The five Es can be used in a number of ways to arrive at a successful system. They can be used to set priorities for design, as they can be used to identify users' needs for a product, and they can suggest design approaches. They can be useful for creating usability goals. It is important, when working with the five Es, to consider them all together, as they are interdependent. It is also important to keep the focus on each of the dimensions in balance; where one dimension takes on a greater significance (for example, the "engaging" dimension in a game), it can be easy to lose sight of the other four dimensions which may impact negatively on your final design. Usability requirements are targets to work toward for a usable UI and a pleasing user experience. Once gathered, the usability requirements and metrics are compiled into a usability specification, which also forms part of the requirements specification.

II. Usability Heuristics

A. Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

B. Match between system and the real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

C. User control and freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

D. Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

E. Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

F. Recognition rather than recall

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

G. Flexibility and efficiency of use

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

H. Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

I. Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

III. USABILITY EVALUATION

Usability evaluation is an important part of today's software development process as it can helpimprove the usability of systems under development. When introduced into the process at theright time, usability evaluations can be cost effective in terms of time and money. The basicaim of usability evaluation is to improve the usability of products. Through usability evaluationpossible weaknesses with regards to a system's usability with the involvement of actual userscan be identified. Usability evaluation involves presenting the users with some tasks which arereflective of the future system use. The results of a usability evaluation can be represented indifferent forms, such as error rates, time taken to complete the task, and number of usabilityproblems found. Usability evaluation is generally carried out in usability laboratories (in-vitro)and in some cases can be carried out in field (in-situ). Holzinger [6] divided the usabilityevaluation techniques into inspection methods (without end users) and test methods (with endusers).

*HeuristicEvaluation*Thinking Aloud
CognitiveWalkthrough Field Observation
Action Analysis Questionnaires

Heuristic evaluations are expert evaluations of products or systems, including informationsystems and documentation. They're conducted by usability specialists, domain experts, orpreferably by "double experts" with both usability and domain experience. Advantage ofevaluation is that it can produce results in a limited time because it does not involve time-consumingparticipant recruiting. The disadvantage is that the results of heuristic evaluationcannot be fully trusted as no real users are involved.

Cognitive walkthrough is a task-oriented method by which the analyst explores the system'sfunctionalities; that is, it simulates step-by-step user behavior for a given task. It traces thecognitive issues, such as learnability of the user, by analyzing their mental processes. Cognitivewalkthrough is known for its benefits such as low cost and quick results. It is helpful in pickingout interface problems at an early stage. However, it can be sometimes time-consuming, andsince restructuring the interface is often expensive and difficult at later stages in development;the cognitive walkthrough is usually applied in early stages of software development.Action analysis involves an inspection of the user actions with regard to physical, cognitive, andperceptual loading. It is helpful in predicting the time a given task will take to complete and alsohelps in getting an insight into the users' behavior. The short coming of action analysis is that itrequires high expertise and time.

With the think aloud protocol, a user is required to verbalize his comments about areas where heis struggling and the reasons for the difficulties. The outcomes of think aloud protocols can beused by the usability practitioner to identify problem areas of the Web site or application beingassessed and to find appropriate improvements. One of the most common think aloud protocolsthat usability practitioners engage in today is concurrent think aloud under which the participantis encouraged to "think out loud" while working on a task.Field observation is the simplest of all methods. It is carried out by visiting the user'sworkplaces and observing their use with target interface or system. Possibly, notes about themajor usability problems being faced by the user are taken by the observer.Questionnaires are indirect usability measures. They don't study the interface directly; rathercollects the user's view about the interface. Questionnaires have to be designed by the expertsand should cover all the experiences with the interface. In order to validate the results of thequestionnaires large number of users has to be assessed.Usability evaluations cannot be simply based on the results of application of one or more of theabove techniques.

IV. LEARNING CURVE

The learning effect can be represented by a line called a learning curve, which displays the relationship between the total direct labor per unit and the cumulative quantity of a product or service produced. The learning curve relates to a repetitive job or task and represents the relationship between experience and productivity: The time required to produce a unit decreases as the operator or firm produces more units.
Inorder to develop a learning curve, we make the following assumptions:
_ The direct task required to produce the 1st unit will always be less than the direct task required for the *n*th unit.
_ Direct labor requirements will decrease at a declining rate as cumulative production increases.
The reduction in time will follow an exponential curve. In other words, the production time per unit is reduced by a fixed percentage each time production is doubled. It use a logarithmic model to draw a learning curve. The direct labor required for the *n*th unit, , is

$$k_n = k_1 n^b$$

$k1$ = direct labor hours for the first unit

n = cumulative numbers of units produced

$$b = \frac{\log r}{\log 2}$$

r = learning rate (as decimal)

V. USABILITY EVALUATORS

A. Completion Rates:

Often called the fundamental usability metric, or the gateway metric, completion rates are a simple measure of usability. It's typically recorded as binary metric (1=Task Success and 0= Task failure). If users cannot accomplish their goals, not much else matters.

B. Usability Problems

(UI Problems) encountered (with or without severity ratings): Describe the problem and note both how many and which users encountered it. Knowing the probability a user will encounter a problem at each phase of development can become a key metric for measuring usability activity impact and ROI. Knowing which user encountered it allows you to better predict sample sizes, problem discovery rates and what problems are found by only a single user.

C. Task Time :

Total task duration is the de facto measure of efficiency and productivity. Record how long it takes a user to complete a task in seconds and or minutes. Start task times when users finish reading task scenarios and end the time when users have finished all actions (including reviewing).

D. Task Level Satisfaction:

After users attempt a task, have them answer a few or just a single question about how difficult the task was. Task level satisfaction metrics will immediately flag a difficult task, especially when compared to a database of other tasks.

E. Test Level Satisfaction:

At the conclusion of the usability test, have participants answer a few questions about their impression of the overall ease of use. For general software, hardware and mobile devices consider the System Usability Scale (SUS), for websites use the SUPR-Q.

F. Errors:

Errors provide excellent diagnostic information and, if possible, should be mapped to UI problems. They are somewhat time consuming to collect as they usually require a moderator or someone to review recordings(although my friends at Webnographer have found a way to automate the collection).

G. Expectation:

Users have an expectation about how difficult a task should be based on subtle cues in the task-scenario. Asking users how difficult they expect a task to be and comparing it to actual task difficulty ratings (from the same or different users) can be useful in diagnosing problem areas.

H. Page Views/Clicks:

For websites and web-applications, these fundamental tracking metrics might be the only thing you have access to without conducting your own studies. Clicks have been shown to correlate highly with time-on-task which is probably a better measure of efficiency. The first click can be highly indicative of a task success or failure.

I. Conversion:.

Conversion rates are also binary measures (1=converted, 0=not converted) and can be captured at all phases of the sales process from landing page, registration, checkout and purchase. It is often the combination of usability problems, errors and time that lead to lower conversion rates in shopping carts.

J. Single Usability Metric (SUM):

SUM is a standardized average of measures of effectiveness, efficiency of satisfaction and is typically composed of 3 metrics: completion rates, task-level satisfaction and task time.

not have spaces: write "C.N.R.S.," not "C. N. R. S." Do not use abbreviations in the title unless they are unavoidable (for example, "INTERNATIONAL JOURNAL OF ENGINEERING AND TECHNICAL RESEARCH" in the title of this article).

VI. LEARNING CURVE ANALYSIS

Parameters

- Task Time
- Completion Rates

Case 1:Task Time
Task=Opening the Software

Softwares	Task Time
MS OFFICE	0.0197
ADOBE READER	0.0117
JAVA	0.0114
MOZILA FIREFOX	0.0177
NOKIA SUITE	0.0117
POWER ISO	0.0118

Log file
C:\Program Files\Adobe\Reader 10.0\Reader\AcroRd32.exe - 1 executions
0.0117
C:\Program Files\Mozilla Firefox\firefox.exe - 1 executions
0.0177
C:\Program Files\Java\jdk1.5.0_03\bin\packager.exe - 1 executions
0.0114
C:\Program Files\PowerISO\PowerISO.exe - 1 executions
0.0118
C:\Program Files\Nokia\Nokia Suite\NokiaSuite.exe - 1 executions
0.0117
C:\Program Files\Microsoft Office\Office14\WINWORD.EXE - 1 executions
0.0197
Case 2:Completion Rate
The above table shows the comparitive study of the various softwares for a document reader that shows the completion time based on the size of the file. Based on the analysis, the learning curve is drawn over the completion time of the various applications for the same file.

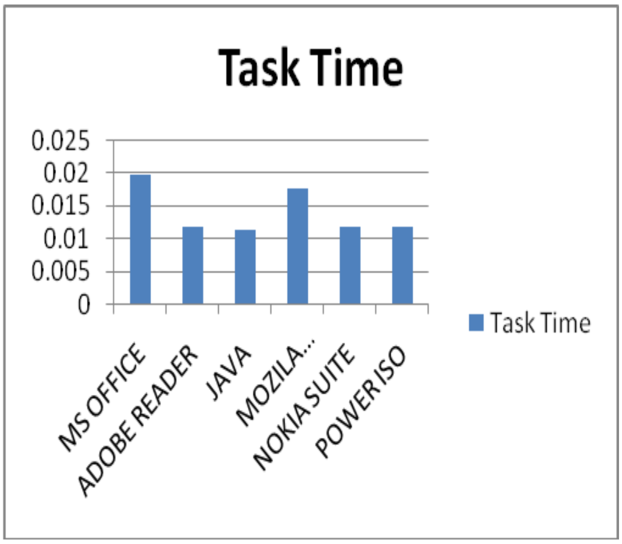

Fig 1. Task time for various appa

Document Size(mb)	MS OFFFICE	OPEN OFFICE	GOOGLE DOCS
5	0.0117	0.0116	0.0125
10	0.0118	0.0117	0.0127
15	0.0120	0.0120	0.0131
20	0.0121	0.0121	0.0132
25	0.0123	0.0123	0.0133
30	0.0130	0.0125	0.0135
35	0.0131	0.0129	0.0139
40	0.0135	0.0133	0.0139

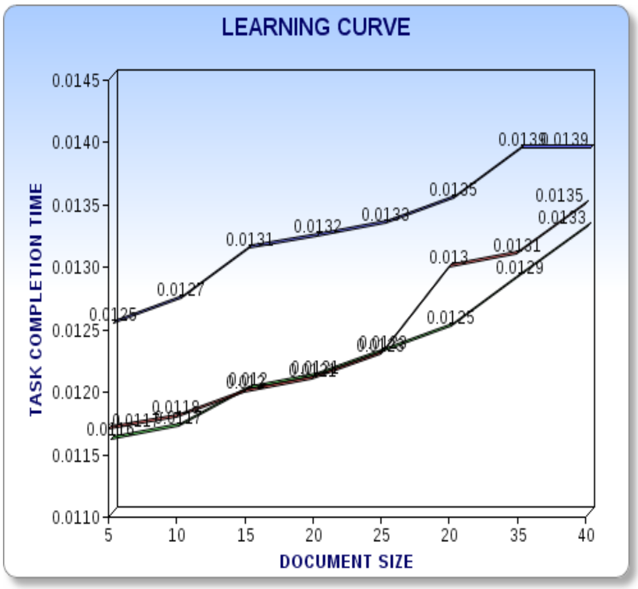

Fig 2. Completion time for various apps

VII. Conclusion

Learning curves provide their greatest advantage in the early stages of new service or product production. As the cumulative number of units produced becomes large, the learning effect is less noticeable. We have analyzed the various softwares with the parameters for the evaluation usability using the learning curve.

References

[1] Venturi, G. & Troost, J. (2004). Survey on the UCD integration in the industry. Proceedings of NordiCHI '04, pp. 449-452, Tampere, Finland, October 23-27, 2004, ACM Press, NewYork, USA.
[2] Woletz, N. (2006). Evaluation eines User-Centred Design-Prozessassessments - Empirische Untersuchung der Qualität und Gebrauchstauglichkeit im praktischen Einsatz. Doctoral Thesis, 4/2006, University of Paderborn, Paderborn, Germany
[3] L. Trenner and J. Bawa, The Politics of Usability,Springer-Verlag, London, 1998
[4] Lamsweerde, A van. Goal-Oriented Requirements Engineering: From System Objectives to UML Models to Precise Software Specifications. Proceedings of the 25th International Conference on Software Engineering (ICSE.03), 2003, 744-745.
[5] Object Management Group, Inc. OMG Unified Modeling Language Specification Version 1.5, March 2003.