ISSN No. 0976-5697

Volume 6, No. 6, July-August 2015

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Usability through Design and its Impact on Reusability

N. Hari Prasad
PG Scholar
Computer Science& Engineering
JNTUACEA, Anantapur
Anantapur, AP, INDIA

G. Ramesh
Lecturer
Computer Science& Engineering
JNTUACEA, Anantapur
Anantapur, AP, INDIA

Abstract: Usability is ''the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specific context of use''. Usability generally referred to as ''quality in use''. A software application even with more services and functionality is futile, if it possess usability problems. The existing work defined usability guidelines and design artifacts to build specific usability characteristics into software applications. These guidelines were named as *"usability guidelines for software development"*. The existing work hypothesized that applying the usability guidelines, reduces development time, reduces perceived usability-associated functionality difficulty for developers, and improves the resulting software designs quality. However, the existing work did not consider impact of using usability guidelines on reusability. In this work usability is increased through design using guidelines as well as assesses its impact on reusability. In this work the usability is trade off against reusability. The guidelines are applied with two modes of guidelines namely, No Guidelines (NG) and Full Guidelines (FG) on each application and compare the usability and quality achieved in each mode. The results obtained from each mode are assessed and compared each other from the perspective of reusability as an intention to find usability impact on reusability by calculating reuse percent of source code of each mode.

Keywords: Usability, Reusability, Usability Guidelines, Reuse Percentage, Effort Saved.

I. INTRODUCTION

Usability is one of the important quality attribute of software product, and refers to the ability of a software to offer its interfaces in a user friendly and elegant way. Among the heuristics for building usable software, particular features that represent concrete functionalities in a software system include feedback, cancel, undo, definition of user profiles, aggregation of commands, wizards etc,[3][8].

The usability characteristics are as follows.
Understandability : How much understandable and suitable a particular software task to user ?
Learnability: How easily application can be learned by user?
Operability: How easily a software can be controlled and operated by the user ?
Attractiveness: Attractiveness is the ability of software product to attract the user, graphical user interface, colour , design, background etc.

Usability attributes percentage in usability as per past published definition is as follows: learnability--19.12%, effectiveness--18.33%, user characteristics--16.73% followed by attitude--12.35%, efficiency--10.16% control/flexibility--3.78%, memorability/ retainability--3.78%, and usefulness--2.19% others--13.56% [5][6].

The Human Computer Interaction(HCI) community has specified particular usability functionalities, which should be incorporated into software applications to beat some of the most common usability issues. Though, for software developers who are not skilled in usability, integrating such usability functionalities into software applications could be complicated that is they don't know when, how, and why usability functionalities should been integrated in to software applications [1].

For more than a decade, the Software Engineering community has been enthusiastically following different lines of research aiming the integration of usability practices into software development. Fraction of this study focuses on integrating HCI techniques and activities into the software development process [1], [4].

Software reusability is generally considered as a approach to solve the software development crisis, and it become a area of much interest in the software community because of its potential benefits, which comprise enhanced product quality and decreased schedule and product cost.

Software reusability is an approach to reuse existing assets in some form within the software product development process. The software assets which can be reused includes software components, source code, test suites, designs(design patterns and architectures) and documentation. In this research work source code reuse is considered. Code reuse is the use of existing source code to build new software application. In this research work source code of Application-1 is reused to develop Application-2. The details about Application-1 and Application-2 are provided in section-III of this paper.

Many research articles and existing works detailed that there is always a tradeoff among usability and reusability. Means if usability is high in application then reusability is low and if usability is low then reusability is high. However, there was no existing work on finding usability impact on reusability when applications are developed by using the usability guidelines for software development provided in [1].

II. MOTIVATION AND BACKGROUND

Many research scholars contributed their ideas and work on usability and reusability separately and some on towards the finding of usability impact on reusability. Let us discuss about existing research works and motivation for this paper.

Miguel-Angel and Elena. [9] , examined the relationship between reusability and context-specific usability in learning objects, giving rise to a novel approach to formulate assessment criteria for learning objects. In addition, they outlined a hesitant assessment method that they borrowed from the field of HCI, as well as from their own experience and knowledge.

Ahmed Seffah et al. [2] identified usability scenarios. And proposed an algorithm for matching solutions to existing patterns as a substitute to defining new solutions for each scenario. For example, the progress indicator pattern(feedback type) is defined as a solution to the generally known usability problem of prolonged interactions that lead to user irritation and dissatisfaction and reduce risk of errors pattern is recommend as a solution to the usability problem of accidents arising from hazardous states.

N. Juristo, et al. [3] proposed a concept of Gathering Usability Information through Elicitation Patterns and proposed a Pattern-Based solution for gathering functional usability requirements. Examined whether software developers could produce usable software using HCI-type information as requirements, and ran a survey at the Universidad Politécnica de Madrid with final-year undergraduate students and concluded that elicitation patterns helps more than HCI-type information or definition of usability in incorporating usability features in software applications.

Laura Carvajal, et al.[1] defined Usability Guidelines for incorporating specific HCI community proposed usability features into software applications. The Figure 1. shows structure of Usability Guidelines for software development proposed by Laura Carvajal, et al.. Preliminary validation results shown that using usability guidelines reduction in development time, decrease perceived usability-associated functionality difficulty for developers, and improve the resulting software designs quality. However, they did not find out impact of using these usability guidelines on code reusability. They developed three applications 1).An online task manager, 2). A console for a home automation system and 3). An auction site, each in three modes NG(using no usability guidelines), PG(Partial usability guidelines) and FG(using full usability guidelines). And found that the particular application developed in FG mode possess high usability, reduction in development time, decrease in usability-associated functionality difficulty and high quality software designs than same application developed in PG and NG modes.

While by following the usability guidelines for software development proposed in [1] increases the usability of system by incorporating the usability features from design phase itself. Existing research is limited to increasing usability of a software application i.e. existing research did not focus on impact of using usability guidelines on other software quality attributes like reusability, performance, security ,… This research work focuses on finding impact of usability guidelines on reusability of source code. Next section of this paper will discuss how to achieve the same.

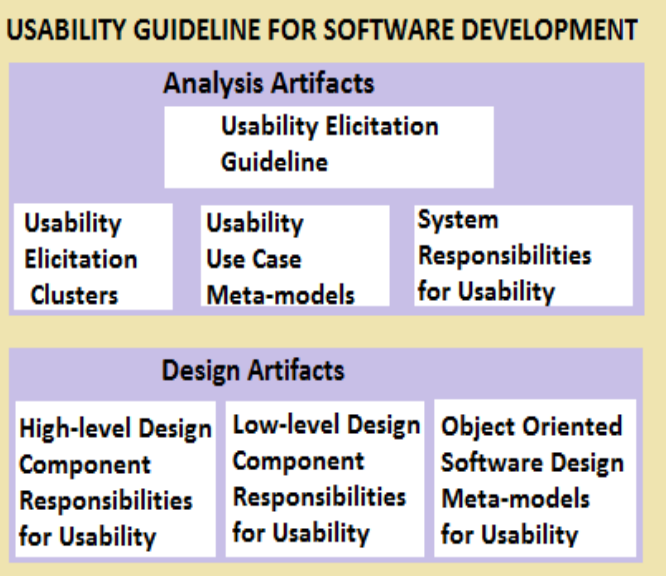

Figure 1. Structure of the usability guidelines for software development.

III. PROPOSED WORK

In this proposed work, impact of using usability guidelines on reusability is find out. At very first each and every usability guidelines i.e., analysis artifacts (Usability elicitation guideline, Usability elicitation clusters, Usability use case meta-models, and System responsibilities for usability) and design artifacts(High-level design component responsibilities for usability, Low-level design component responsibilities for usability, and Object oriented software design meta-models for usability) are well understood . A frame work is defined to find the usability impact on reusability.

A. Frame Work

The aim of this work is clear i.e., to find usability impact on reusability when software application is developed by using usability guidelines for software development, so for achieving the same a framework is defined as shown in Figure 2.

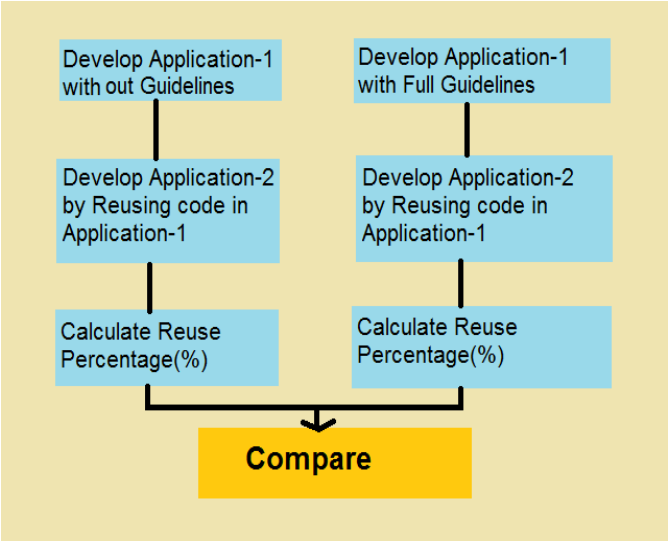

Figure 2. Framework for finding usability impact on reusability.

According to the framework, first Application-1 has to be developed without usability guidelines and with usability guidelines. So a web application named '*Mapping of Colleges and Industries' is developed* as Application-1. Secondly Application-2 has to be developed by reusing the source code written for Application-1. Application-2 without usability is developed by reusing the source code of Application-1 without usability guidelines, Application-2 with usability is developed by reusing the source code of Application-1 with usability guidelines. A web application '*Online Interview Preparation*' is implemented as Application-2.

Usability features included in Application-1 with usability are: Progress indicator, Warning, System status , Abort operation, Step-by-step execution.

Usability features included in Application-2 with usability are: Warning, Progress indicator, Abort operation, System status.

Application-1: Mapping of Colleges and industries

A website to show a JNTUA Anantapur affiliated college location and its near by industries location on google map and vice-versa industry and its near by colleges location. All colleges affiliated to JNTUA Anantapur are listed in the website, when we select a particular college, selected college location and its nearby industries (if any) locations are shown on google map and vice-verse industry and its nearby colleges. The URL of website is http://www.jntuamci.in

Application-2: Online Interview Preparation

A website for preparing for written test and interview of top MNC companies such as TCS, IBM, Google, Microsoft, etc. A new user can register his/her details and already registered user can login into his account. After login a user can post his question, can answer a question asked by other users and can view a answers for a particular question asked by others or his/her own.

After completion of development of Application-1(with and without usability guidelines), and Application-2 (with and without usability) by reusing the code of Application-1, the next task is to calculate reuse % (percent) of source code achieved in Application-2 without usability and Application-2 with usability. The Reuse percentage is calculated by following formula.

RP = (RLC / TLC) * 100 ---------- > Eq(1)

Where

RP --- Reuse Percentage.

RLC --- Reused Lines of Code.

TLC --- Total Lines of Code (New and Reused Lines of Code).

There should be Additional Effort(AE) needed to incorporate the reused lines of code into Application. Additional Effort Factor AEF is taken as 0.2. For example Reused LOC is 1000, there is a need of additional effort AE = AEF*1000 = 0.2*1000 = 200. Means effort require to reusing a 1000 LOC is equal to effort required to writing new 200 LOC.

Next Total Effort Saved(ES) is calculated as follows.

ES=(1-AEF)*RP ---------->(Eq2).

Finally Return on Investment(ROI) is calculated . In order to calculate ROI there is a need to calculate Reuse Cost Avoidance(RCA) and Additional Development Cost(ADC).

Reuse Cost Avoidance(RCA):

RCA=DCA+SCA ---------->(Eq3)

Where

DCA= (NDC * RSI) * (1-RCR)

SCA= ER * CPE * RSI*.001

DCA---Development Cost Avoidance

SCA--- Service Cost Avoidance

NDC---New Develoment Cost

RSI---- Reused Source Instructions (LOC)

RCR--- Relative Cost of Reuse

ER ---- Error Rate (errors/KLOC)

CPE--- Cost per Error ($/Error)

Additional Development Cost(ADC):

ADC=NDC * SIWRO * (RCWR-1)----------> (Eq4)

Where

NDC--New Develoment cost

SIWRO--Source Instructions Written for Reuse by Others (LOC)

RCWR--Relative Cost Writing Reuse

Now ROI:

ROI= RCA-ADC ---------->(Eq5)

SIWRO represents the amount of source code intentionally wrote for reuse (reusable code) by others in future. An extra effort is needed to write this source code, which is nothing but an investment in reuse. The units are lines of code (LOC). In this work SIWRO=0, because no source instructions are written intentionally for reuse. *NDC* represents the historical average cost per lines of code. Uses dollars ($) for units. In this work NDC is industry average i.e. $100/LOC is taken as default value.

RCR represents the effort to use existing source code as compared to writing new source code. Based on several factors the RCR can vary and can range from about 0.03 up to about 0.4. In this research RCR is taken as 0.2, which represents that the effort needed to reuse source code is 20 percent, as if we write same code newly it takes 100 percent effort. *RCWR* represents the additional effort takes to write a source code with the intent of reusing it later that is reusable code versus writing non-reusable source code i.e, for one-time use only. Based on several factors the RCWR may vary from 1.0 to 2.2. In this research *RCWR = 1.5*, which represents that the additional effort required to produce reusable source code is 50%.

ER represents the average error rate in previously new developed softwares by your organization or team, in errors per kilo lines of code. In this research ER is taken as 0.5 errors/kLOC. The units are errors per kilo lines of code (errors/kLOC). *CPE* represents average cost to fix errors in previously new developed softwares after releasing to the customer, in dollars per error. In this research CPE is taken as $1,000/error.

DCA represents the cost that can be avoided while developing the application by reusing source code. Uses dollars ($) for units. *SCA* represents the cost that can be avoided while maintaining the application by reusing source code. Uses dollars ($) for units. *RCA* represents the total cost avoided from reuse of source code obtained from some other place. Since some cost can be avoided during both development phase (because you reuse existing code, don't have to write source code newly) and maintenance phase(because errors in the code are already fixed, you don't have to fix them in maintenance). Uses dollars ($) for units.

ADC represents the additional cost resulting from writing reusable source code for use in other applications or software in future. Uses dollars ($) for units.

IV. RESULTS

This research results state that the reuse percent of code, effort saved, ROI is high in Application-2 with out usability than in Application-2 with usability. i.e., reusability is high when Application-2 is developed by reusing the source code of Application-1 with out usability.

Application-2 without usability has a source code of 1640 LOC in which 775 LOC is reused code of Application-1 without usability and Application-2 with usability has a source code of 2148 LOC in which 709 LOC is reused code of Application-1 with usability. The reuse percent is calculated with Eq(1), reuse percent is high when Application-2 is developed by reusing the code of Application-1 without usability, Figure 3. shows the reuse percent achieved in Application-2 without usability and with usability.

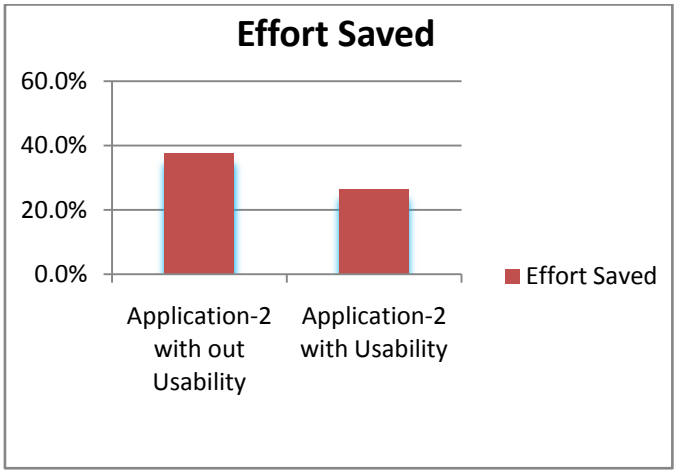

Figure 4. Effort Saved in Application-2 without and with usability.

Table 1. ADC, RCA and ROI for both Application-2 without usability and Application-2 with usability.

Reuse Metric / Application	Reuse percent	Effort Saved	RCA	ADC	ROI	Usability features
Application-2 without usability	47 %	37.6 %	$62.39K	$0K	$62.3875K	No
Application-2 with usability	33 %	26.4 %	$57.07K	$0K	$57.0745K	Yes

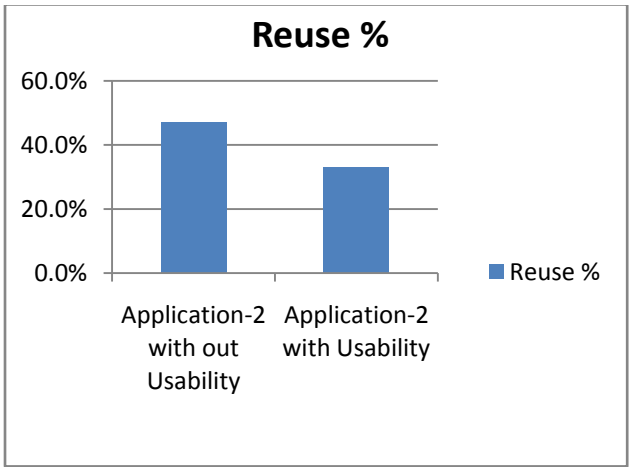

Figure 3. Reuse Percent of source code in Application-2 without and with usability.

The effort saved is calculated with Eq(2), as that of reuse percent ,effort saved is also high, when Application-2 is developed by reusing the code of Application-1 without usability, Figure 4. shows the effort saved in Application-2 without and with usability. RCA, ADC and ROI are calculated with Eq(3), Eq(4) and Eq(5) respectively, ROI is also high in Application-2 without usability, Table 1. shows the ADC,RCA and ROI for both Application-2 without usability and Application-2 with usability. ADC is $0 for both Applications because SIWRO is 0 in both Application-1 without usability guidelines and Apllication-1 with usability guidelines.

V. CONCLUSION

In this research impact of using usability guidelines through design on reusability is investigated. First each and every usability guidelines for software development are well understood. In this work a framework is proposed to find usability impact on reusability when applications are developed by using usability guidelines. Two applications are developed, Application-1 in two modes without usability guidelines and using guidelines, Application-2 by reusing the code in Application-1.

In this work usability impact on reusability is findout by comparing the reuse percent achieved in Application-2 without usability and usability. And the results shown that reuse percent is high in Application-2 without usability. Also effort saved and ROI is high in Application-2 without usability. This research work concluded that reusability is reduced when usability is increases i.e there is trade off between usability and reusability when applications are developed using usability guidelines. Software engineers or research scholars can find usability impact on other quality attributes like performance, security, maintainability etc,.

VI. REFERENCES

[1] Laura Carvajal, Ana M. Moreno, Marı´a-Isabel Sa´nchez-Segura, and Ahmed Seffah, "USABILITY THROUGH SOFTWARE DESIGN", IEEE Tran. On Soft. Eng., Vol. 39, No. 11, November 2013.

[2] Seffah, T. Mohamed, H. Habied-Mammar, and A. Abran, "Reconciling Usability and Interactive System Architecture

Using Patterns", J. Systems and Software, vol. 81, no. 11, pp. 1845-1852, 2008.

[3] N. Juristo, A.M. Moreno, and M. Sanchez-Segura, "Gathering Usability Information through Elicitation Patterns", J. Systems and Software, vol. 80, no. 9, pp. 1507-1516, 2007.

[4] ISO IEC 9126-1:2001, "Software Engineering-Product Quality-Part 1-Quality Model," 2001.

[5] Satyam Saxena, Sanjay Kumar Dubey, "Impact of Software Design Aspects on Usability ", International Journal of Computer Applications (0975 – 8887) Volume 61– No.22, January 2013.

[6] Chen,Y., "Defining Usability: How Library Practice Differs from Published Research" .

[7] Ralph M. DeFrangesco, "A Cost Model for Software Reuse".

[8] L. Constantine, L. Lockwood. "Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design". Addison-Wesley, 1999.

[9] Miguel-Angel, Elena . "On the Concepts of Usability and Reusability of Learning Objects"., October-2003.

[10] Ms.Kirandeep Kaur, Ms.Rekha Rani, Ms.Jagdeep kaur, "Code Reuse & Reusability of the Software" ,The International Journal Of Engineering And Science (IJES) ,Vol. 2 ,Issue. 4 , Pages. 28-30, 2013.

[11] https://en.wikipedia.org/wiki/Reusability.

[12] http://jeffreypoulin.info/html/reucalc_basic.html.