Impact of Software Design Aspects on Usability

Satyam Saxena
Amity School of Engineering & Technology,
Amity University, Sec-125
Noida (U.P.), India

Sanjay Kumar Dubey
Amity School of Engineering & Technology,
Amity University, Sec-125
Noida (U.P.), India

ABSTRACT
Software usability of software system specified how much user feels comfort and attract towards specific system environment, on the other hand Software designing is also an important factor affecting software system success. Usability is not a function of time although researchers have come up with various models and matrices relating these two concepts. It is very important quality attribute of software system. The success and failure of system in economic manner depends on the software usability, which is directly related to software design. This survey paper discusses on various aspects of designing and their effect on usability of system. Measurement in software is still in its initial stage and it has been found that modern day software's have become so complex that it directly affects the software usability. In this survey paper also discuss about various impacts that occur on usability due to software design complexity and discussed various methodologies and previous work related to design complexity and Usability separately as well as together, so that it will become easily analyze impact of software design on software usability.

Keywords
Usability, Software Design Complexity, Human Computer Interaction, Maintainability

1. INTRODUCTION
To perform any task or subtask the interaction between system and software is referred to as software complexity [2]. Study of usability conducted as a primary job function by analyst designers, technical writers, marketing personnel, and others. Software design complexity and usability both concepts depend on each other. If there is increment in the software design complexity then its impact on usability occurs in the form of delay, fault and unexpected outcomes. A study suggested that an user friendly redesign of an automated system of call center support increased 2 million dollar revenue within starting 10 month which provide better experience [13]. Another survey on usability suggests that every dollar spent on usability returns 30.25$ [4]. Usability experts suggest that understandability of developer towards interface also affect a system's perceived usability [27]. Expected desired solutions do not support usability that cause user interaction decrease towards system, it requires some effort in software design if developer want to deliver a usable product to the customer [3].

2. SOFTWARE ARCHITECTURE
Software architecture is a complete set of instruction, data, documentation, software design, algorithms, data type and manual which comprise relation between these attributes to satisfy the requirements [7, 20]. Documentation provides communication between various stakeholders, which increases the usability of the system on developer and user side.

2.1 Software Design
Software design is algorithmic approach to plan and solve the software related problem with the help of various graphical representations like ER, DFD, UML diagrams etc. It provides design time usability to reuse previous components and complete view of system software with user manual and other design implementation like HCI (Human Computer Interface) [3]. The objective of software architecture is to reduce the software complexity which leads to the problem because different person use different ideas and terms to solve a particular problem.
There is no strictly defining way for development of software architecture in the literature, Due to this scenario area of interest tends to non fictional requirements of system like maintainability, quality, security, usability, fault tolerance, extensibility.

2.2 Software Usability
Usability concern with three basic term how efficient, how early user can learn and how much user is satisfied (ISO 9241, (1998)) define the usability as "The extent to which a product can be used by specified users to achieve specified goals with efficiency, effectively and satisfaction in a specified context of use". Characteristics of usability are as follow.

2.2.1 Understandability
How much particular software task is understandable and suitable for user?

2.2.2 Learnability
How easily user can learn applications.

2.2.3 Operability
How easily user can control and operate the software.

2.2.3 Attractiveness
Attractiveness is the ability of software product to attract the user, graphical user interface, colour, background etc.
According to past published definition usability attribute percentage in usability 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) [6].

2.3 Human Computer Interaction
Human Computer Interaction is a method which measures the interaction of humans with the computer machine. It deals with the usability and the comfort level of the user towards

the system. It keeps a track of how much the system is flexible for the user needs and how it responds to the user requirements. This also measures the effectiveness of the system in terms of ease of use and popularity. The study planning and design of interaction between user and computer is part of HCI.

HCI studies a human and a machine in conjunction; it draws inferences from supporting knowledge on both the machine and the human side. On the machine side specification, the consideration about operating systems, programming languages, development environments and computer graphics are relevant. On the human side, communication theory, graphic and industrial design disciplines, linguistics, social sciences, psychology, and human factors such as computer user satisfaction and usability are relevant. Attention to human-computer interaction is important, because poorly designed computer interfaces can lead to many unexpected problems.

A basic goal of HCI is to improve the relations between users and computers by making computers more usable and friendly to the user's needs. Specifically, HCI is concerned with:

i. Methodologies and processes for designing interfaces.

ii. Methods for implementing interfaces and efficient algorithms.

iii. Techniques for evaluating and comparing interfaces

iv. Developing new interfaces and techniques

v. Developing descriptive and predictive models and theories of interaction

A long term goal of HCI is to design systems that minimize the obstacle between the human's cognitive model of what they want to accomplish and the computer's understanding of the user's task.

3. RELATION BETWEEN SOFTWARE DESIGN AND SOFTWARE USABILITY

System design and architecture have impact on human how he percept the data, learn the information and how credibly judge it. Usability attributes generally deal with several design patterns.

GUI and Wizard Increase the system attractiveness that's increase the user learning desires towards system that will increase the operability and efficiency of system. If user face any problem then he provide feedback to system and system perform changes as per feedback that will increases the system usability.

There have been various empirical studies done in the field of software complexity and usability individually as well as together. Further study suggests the impact of software design architecture, which helps to predict usability attribute. These studies will help to improve software usability which increases productivity. The Various approaches in this context are defined as follows:

3.1 Impact of Usability on Software Design on Software Usability

Here, this paper specially focused on impact of usability attributes on design. Software usability depends on software design. Design complexity of any system is depends on user specified requirement, although dealing with usability during software system design contributes to improve this quality attribute, it is not the only action to be taken to achieve this purpose [6].

Software design complexity provides system metrics to developer, which plays an important role in management tool. Design metrics have three measures module design complexity, design complexity, and integration complexity. These metrics are derivatives of the well constituted cyclomatic complexity. Design decision have important impact on functional and non functional requirements, increment in complexity of interface will increase the failure of source code i.e. more design complexity likely more bugs [5][24].

3.2 Impact of Software Design Complexity on Cost

The concern with design complexity required resources are increase for maintain the system and properly utilization. So, more man power and extra management activities are required which require extra expansion so that, increment in design complexity of any software then cost of that software will automatically increase. Example: when oracle changed its navigation structure, database administrators perform their duties 20% faster [4].

3.3 Impact of API Complexity on Usability

Application program interface (API) approach suggest, build a system with semi-independent work. This will allow developer minimize impact on fallows component. Various MNC's like Google, Apple, and Microsoft rely on this technology. APIs are the link between modules or components that allows for semi-independence and various development activities, therefore facilitating the coordination among developers. Study suggests that open source system accept module based approach then other software system [19]. Module based system approach require less emphasis on software testing, experimental cost and maintenance cost. Failure of one or module may put partially or fully software system in critical phase.

4 Design Issue of Software in Context of Usability

Design Issue	Usability	Approach
Design issues are solved in initial phases.	Future of usability based on developers understanding and intelligence.	Waterfall Model
Design become complex during development phase	Results high usable software.	Prototype Model
Addition in functionality may cause errors in software architecture	User get more functionality as per increments in project	Incremental Model
Any difficulty in design, coding and testing a modification should signal the need for redesign or re-coding	User reaction should be solicited and analyzed for indications of deficiencies in the current implementation.	Iterative Model
accurately reflect the software development process still produce insufficient testing methodology	No ability to respond to changes	V-Model
Supports iteration	It requires considerable experience in risk management for the project to be successful.	Spiral Model
This agile model support empirical process control strategy which opposed tradition approach	Use multiple feedback loops. It Provide speed and flexibility in commercial projects.	Scrum
Develop product Interactively [21]	No compromise with quality attributes [21].	DSDM
High productivity. Quick response to change in plan.	Lack of previous data may cause unsatisfactory output. Provide better quality. High satisfaction.	Agile

5 ANALYSIS OUTCOMES

In the figure 1 shows relation between the software design attribute and usability attributes which directly affect HCI. Software design consists of GUI, wizard, feedback as visual components. GUI and Wizard Increase the system attractiveness that's increase the user learning desires towards system that will increase the operability and efficiency of system. If user face any problem then he provide feedback to system and system perform changes as per feedback that will increases the system usability.

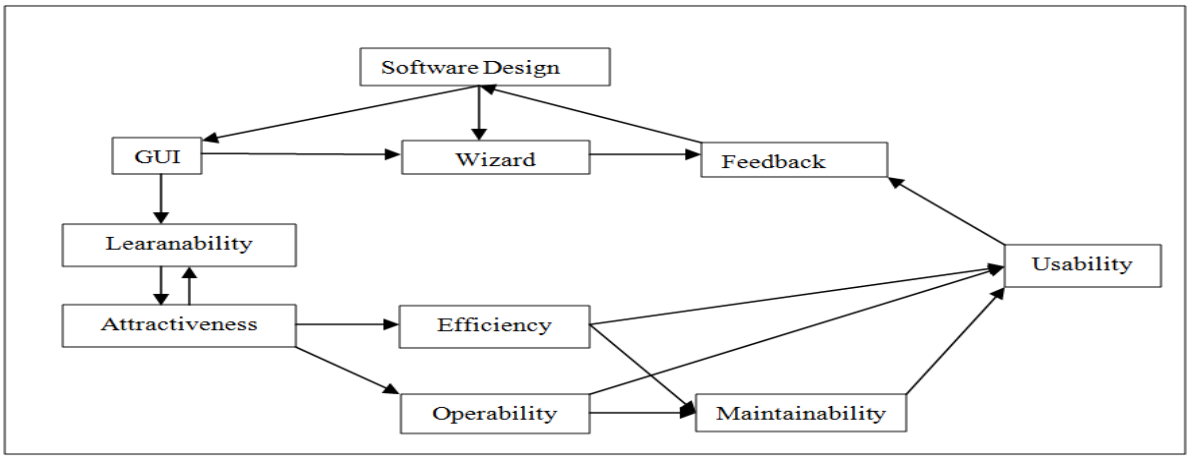

Fig 1: Communication between Various Software design Component and Usability Components in HCI

The above diagram (Fig 1) shows the relationship between Software Design and Usability Components.

5.1.1 Software Designing

Software design is a method of problem solving and developing for a software result. After the functions and conditions of software are determined, software developers then design and employ designers to build up a plan for a solution. It includes component and algorithm performance issues as well as the architectural view. The software requirements analysis of a software development process yields specifications that are used in software engineering. If the software is semi-automated, the software design may involve user experience using a story board to help determine those specifications. If the software is totally automated, then the software design may be extremely simple as a flow chart or text describing a premeditated sequence of events.

5.1.2 Graphical User Interface (GUI)

GUI is a type of user interface that allows users to interact with devices using images rather than text commands. GUIs can be used in computers, hand-held devices such as portable media players or gaming devices, household appliances, office and industry equipment. A GUI represents the information and actions available to a user
through graphical icons and visual indicators such as secondary notation, instead of text-based interfaces, typed command labels or text navigation in Character User Interface. The actions are usually performed through direct manipulation of the graphical elements.

5.1.3 Wizard or Setup Assistant

Wizard is a user interface type that presents a client with a sequence of input spaces that lead the user through a series of well-defined algorithm. Tasks that are complex, rarely perform, or unknown may be easier to carry out using a wizard. In expert system guides a user through a series of (usually yes/no) questions to solve a problem. Wizards had become commonplace in most consumer-oriented operating systems, although not always under the name "wizard". In Mac OS X, for example, they are called "assistants"; some examples include the "Setup Assistant", MS-Windows "New Connection Wizard".

5.1.4 Feedback

Feedback is a method in which data about the past and present influences the phenomenon in the present or future. As a part of the chain of cause-and-effect that forms a loop, the event is said to "feedback" into itself. Feedback is defined as "Information about the difference in the actual level and the ideal level of a system parameter which is to be used", meaning that the info by itself is not feedback until translated into action.
The Bottom part of the diagram has following 5 parts viz.

5.2.1 Attractiveness or attraction

It refers to a quality that causes an interest in something. When user interacts with system then how show its qualities and features so that user like to work on particular system. This characteristic of human nature comes under attraction e.g. there are lots of social networking sites still people like “Facebook” most.

5.2.2 Learnability

In software engineering, according to ISO/IEC 9126, is the capability of a software product to enable the user to learn how to use it. Learnability may be considered as an aspect of usability, and Human Computer Interaction is of major concern in the design of complex software applications. More software interaction may increase software learnability vice versa.

5.2.3 Operability

Operability is the ability to keep a system or a complete installation in a safe and reliable functioning condition, according to pre-defined operating requirements. In a computer environment with many systems this includes the capability of products, systems and business processes to work together to achieve a common task such as finding and returning availability of inventory for flight. Operability is

considered one of the entities and is closely related to reliability, supportability and maintainability. Operability also refers to whether or not an operation can be performed to debug a program with a reasonable degree of safety and chance of success.

5.2.4 Efficiency

Efficiency describes the degree to which time, effort and cost is well used for the planned task or purpose. It is used with the precise idea of relaying the ability of a detailed purpose of effort to produce a specific outcome efficiently with at least amount or quantity of waste, expense, or gratuitous effort. "Efficiency" has widely varying meanings in different aspects. The term "efficient" is very much baffling and distorted with the term "effective". In general, efficiency is a quantifiable concept, quantitatively determined by the ratio of output to input. The difference between Efficiency and Effectiveness is that, "Efficiency is making things right, while Effectiveness is making the right things" [28].

5.2.5 Maintainability

Maintainability is the simplicity with which a product can be maintained in order to:

i. Identify defects or reason of defect
ii. Correct defects
iii. Meet requirements
iv. Make maintenance easier
v. Adapt new environment

In some cases, maintainability involves a system of continuous improvement i.e. learning from the past in order to improve the ability to maintain systems, or improve reliability of systems based on maintenance experience. If a system is more efficient with high operability then maintainability of system will become easy and maintainable cost will decrease that cause high usability of system.

7. CONCLUSION

Refine of this paper to show the impact of software design complexity on software usability and to show what are impact occur due to various way of design methodology. Dealing with software design and its implementation part not only shows purpose to complete all action as per SRS or user defines requirements but software developer task is more than it to maintain software usability and quality. If a system is not able to provide proper satisfaction and fail to interact the user then software will go in "wear out" phase still it provide all desire action and task. On the other hand it is not mean that if we increase the software GUI feature than usability will also increase. When we increase the software usability than software design complexity automatically increase that cause lacking in usability. so that there is need pay special attention on software design complexity and usability in balance manner.

8. REFERENCES

[1] Akaikine,A., "Impact Of Software Design Product On Maintenance Cost And Measurement O Economic Profits Of Product Redesign", Massachusetts Institute of Technology(MIT) June 2010.

[2] Basili, V.R. Qualitative software complexity models: A Summary. In "Tutorial on Models and Methods for Software Management and Engineering" IEEE Computer Society Press, Los Alamitos, Calif., 1980.

[3] Bass, Len; Clements,P., Rick Kazman (2012). "Software Architecture In Practice", Third Edition. Boston: Addison-Wesley. pp. 25–37. ISBN 0-321-81573-4.

[4] Black, J., 2002. Usability is next to profitability. BusinessWeekOnline. <http://www.businessweek.com/technology/content/dec2002/tc2002124_2181.htm>.

[5] Cataldo M., Cleidson R. B., Souza, David L.B., Miranda T. C., Nambiar,S. "The Impact of Interface Complexity on Failures: an Empirical Analysis and Implications for Tool Design" January 2010 CMU-ISR-10-100.

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

[7] Clements, Paul, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Paulo Merson, Robert Nord, Judith Stafford (2010). Documenting Software Architectures: Views and Beyond, Second Edition. Boston: Addison-Wesley ISBN 0-321-55268-7.

[8] Dubey, S. K. and Rana,A. "Analytical roadmap to usability definitions and decompositions", / International Journal of Engineering Science and Technology Vol. 2(9), 2010, 4723-4729, 2010.

[9] Dubey, S. K. "survey on impact of software metrics on software quality", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 1, 2012.

[10] Dubey, S. K. and Rana, A. "A comprehensive assessment of object-oriented software systems using metrics approach", IJCSE, Vol 2 (8), 2010, pp. 2726-2730.

[11] Jain P., Dubey, S. K. "Analysis and Performance Evaluation of Software System Usability", International Journal of Computer Applications (0975 – 8887) Volume 43– No.17, April 2012.

[12] Joseph K. Kearney, Robert L. sedlmeyer, william b. thompson, michael a. gray, and michael a. adler "Software complexity measurement" ACM OOOl-0782,/R6/1100-1044 75, 1986.

[13] Karat, C. & Lund, A. (2005). The Return on Investment in Usability of Web Applications.

[14] Bias & D. Mayhew (Eds.), Cost Justifying Usability (2nd edition).

[15] IEEE Transactions on Software Engineering (2006). "Introduction to the Special Issue on Software Architecture". Retrieved 2006-09-23.

[16] Briand, L., Wüst J., Daly, J., and Victor Porter, D. "Exploring the relationships between design measures and software quality in object-oriented systems," Journal of Systems and Software, vol. 51, 2000, pp. 245-273.

[17] Nielsen, J. Alertbox "Introduction to Usability". Retrieved 2010-06-01

[18] Juristo, N. Moreno A.M. Maria-Isabel Sanchez-Segura "Analysing the impact of usability on software design The Journal of Systems and Software" 80 (2007) 1506–1516, 2007.

[19] MacCormack, A. , Rusnak, J. Baldwin,C. "Exploring the Structure of Complex Software Designs: An Empirical

Study of Open Source and Proprietary Code," Management Science, vol. 52, pp. 1015-1030, 2006.

[20] Perry D. E., Wolf A. L. (1992). "Foundation for the Study of Software Architecture." ACM SIGSOFT Software Engi-neering Notes 17 (4), pp. 40—52.

[21] Richards,k.,"Agile Project Management:Integrating DSDM Atern into an existing PRINCE2™ environment" White Paper March 2010

[22] Schwaber, Ken (February 1, 2004). "Agile Project Management With Scrum". Microsoft Press. ISBN 978-0-7356-1993-7.

[23] ShineTechAgileSurvey2003-01-17.pdf (http://www.shinetech.com/attachments/104_ShineTech AgileSurvey2003-01-17.pdf)

[24] SoftwareArchitectures.com (2006). "Intro to Software Quality Attributes". Retrieved 2006-09-23.

[25] Situated Research "Incorporating Usability Experts with Your Software Development Lifecycle: Benefits and ROI" info@situatedresearch.com

[26] Thomas J. Mccabe and Charles W. B. "Design Complexity Measurement And Testing System Designers Can Quantify The Complexity Of A Software Design By Using A Trio Of Finely Tuned Design Metrics", Volume 32 Number 12 December 1989.

[27] Tractinsky, N., Katz, A., & Ikar, D. (2000) What is beautiful is usable [Electronic Version]. Interacting with Computers, 13(2), 127-145.

[28] www.wikpidia.org