Software re-engineering role in human computer interaction (HCI) with quality assurance

M. Muzammul, Ayesha Zafar, M Yahya Saeed, Najaf Ali

Government College University, Faisalabad, Pakistan
Department of Software Engineering

Abstract
Instead of developing new software, when we follow up re-engineering techniques in form of reliability, cost reduction, reusability and productivity come into existence as the goals of software re-engineering and human computer interaction. Software interface adoptive trend leads toward reengineering instead of new system or software development. In this paper, we proposed software re-engineering deals with the alteration, including or excluding features in existing legacy software systems that lead to quality and effectiveness. We adopted reverse as well as forward engineering to give attractive, interactive software system as actual business demand. As developing trends remain in change process, for reengineering latest trends tried to follow, which refactor code, design but don't disturb the interactive system quality. We conclude that reengineering can be effective in sense of maintainability and cost productiveness.

Key words:
Software; reengineering; HCI; Revers, forward engineering; Quality assurance

1. Introduction

Software re-engineering (SRE) process is a reverse process of software development that is also known as reverse engineering [1]. SRE work with the betterment of internal software quality in form of code refactoring, design refactoring, including or excluding modules without disturbing external software behavior [2]. Code and design is main supervisor of software, if both of these parts work accurately then our working atmosphere remain user friendly in from of user satisfaction of software throughput, productivity and output results [3]. If code does not work accurately then user interaction disturbed as the results of disturbance, the productivity of software, user interaction and in business point of view, product cost and productivity got disturbed, which can destroy whole infrastructure of business logic [4].

1.1 Real time example

In other words we can give an example of building, if the design and the material to develop a building is correct then the output look and results of builted product will be inspirational and user will got really great interaction, mental satisfaction with the validity and perfectibility of developed building, and if the design or material used for building development is not well organized then the output results will be miserable and destroyable and it will not attract the human interactions that can be in form of satisfaction and validity.
Similarly, our software environment work, if the software developing procedure adopted was good and the architecture and coding capabilities of developer and architecture engineer were skilled then he will be able to prepare such a product that will interact according to user demand of mentality and productivity. However, discussion was all about software engineering and human software interaction.

1.2 Main Focus of research (RQ)

Now, questions are that,
1) Why we need to adopt re-engineering?
2) Why re-engineering is more beneficial as compared to new software development
3) How re-engineering interact with users?
4) Is re-engineering is more quality full and risk managerial activity?
Mostly, answers will be yes, we will try to prove my hypothesis with valid theories and reports.

1.3 Solution of problematic statements

Re-engineering is defined at start up, first question's answer is that if we check history of software engineering we will get idea people adopt SE to solve their problems that really exists in our society? By the developing trends of software engineering they become software system addictive and cannot survive without software systems [5]. How this concept come into existence?
When people used software they looked many benefits of software systems in form of productivity, cost, and security of records and safety of records [6]. In short time they did more work and saved it permanently for later use that is why they adopted software infraction in form of GUI and software environment usability [7].

Manuscript received September 5, 2018
Manuscript revised September 20, 2018

Secondly, because when any programmer will develop new software then adoption of this system or becoming use to with system will take a lot of time, that will be great loss to organization. To make infraction availability we implemented re-engineering and we obtained results [8] .SRE was cost effective and validity approach to manage risks. It spread light on human computer infraction and did not disturbed the external interface of software that were legacy software and were part of firms from long years.

By re-engineering we added more modules and features in form of validity checks of design procedures and refactoring code quality result obtained in form of reliability, speed and productivity of software system. A proper system development with new requirements was a huge risky task. We implemented re-engineering approaches on existing software systems with the improvements [9].

1.4 Software re-engineering with collaboration of human computer interaction and quality assurance

Software Re-engineering (SRE) and Human computer interaction (HCI) are similar as two branches of tree with same root and seed. Both fields purpose is developing and designing suitable systems with proper requirements analysis process (RAP).Main aspects included the use of “iterative or incremental” approaches and quality assurance. For each part there may be different ways to address problem, adopting methodologies and developing process. For HCI first part is human as focused area but for SRE the main objective is developing software product at minimum level of cost at specified time with adding more and advance features in existing product. The users are involved from starting to end node of developing and reconstruction of existing product as an evaluation sign. However, the process of minimizing design as well as coding errors is little but time consuming. Even we get results at early stages of reengineering it look like stars on the ground but become possible with the HCI collaborations and SRE development specialists

1.5 Real time research implementation

We concluded that new user interactive systems come into existence with 50% low in costs and 90% decrease in risks. User felt more satisfaction and appreciated .We elaborated our concept with examples of many organizations where we conducted surveys about our study, Number of organizations was near about 100 as a sample of 2 million people related to different fields of work, organizations and locations, in which 50-60% organizations with employee range from 10 persons to 600 employees and remaining 50% were multinational as well as word wide companies. During surveys response factor remained 85% and 10-15% people did not followed our survey. We implemented software after reengineering in 25 organizations and got results 21 organizations were 95% satisfied from user interaction, quality and productivity point of view. And 4 organizations were 50% satisfied, reasons were their lack of excluding results infrastructure, thinking or we can say they were quite low level professional workers or local type organizations. We conclude that our re-engineering give lot of benefits with user mental satisfaction in form of cost, time user interface instructiveness and software usability as well as productivity.**section-1**

2. Re-engineering process with workflow

Re-engineering process domain start from the extraction of design, code from existing legacy software system and range to target user demanded system. This workflow can be performed by using some tools that extract entities and convert it into UML, state, ERD diagram in form of code, several code conversion tools also available that perform code refactoring duty like we can say convert C language to C# or JavaScript to Node.js or from one operating environment to another (Linix to DOS, or DOS to GUI).In real sense, re-engineering process is not an easy task, a lot of complexities also present in the way of work. We can say old legacy systems don’t has proper documentation or design and coding environment info. There may be system crashed and recovering of software unavailable, and extracting design from code and getting requirements may be complex task with redesigning and restructuring procedure with the use of object-oriented programming. Figure 1 explain the concept of re-engineering process from level of abstraction to software implementation.

3. Software Re-engineering focused working area

Here we will explain the key work and clear domain of software re-engineering

➢ *Reverse engineering(RE)*

It is process of analyzing the working domains of existing software by identifying code from implementation and extracting design from coding at last create data file for adding more features

➢ *Forward Engineering(FE)*

In this process, we develop new software with more and advance features as data is obtained, we specify the requirements from data, from requirements designing architecture prepared and after that we

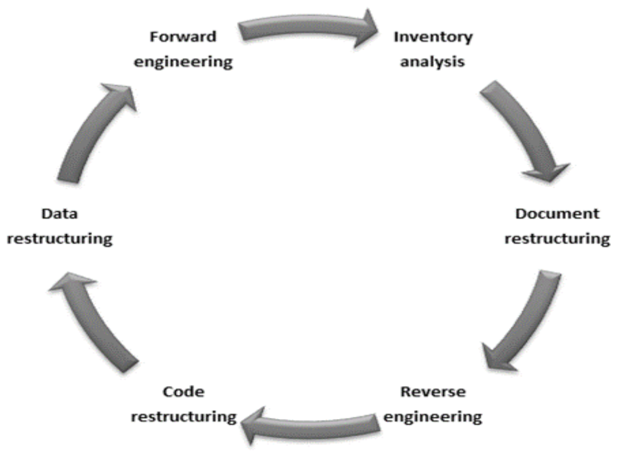

Fig. 1 Re-engineering process

Perform coding process and developed product got implemented

➢ *Programming cohesion and coupling*

We use modular programming approaches during development process. In old systems there were lot of dependencies for working that called coupling and interconnection between modules is called cohesion .In quality and UI point of view, we try to develop system that has great level of cohesion and less coupling

➢ *Extracting Requirements(FE)*

As we collected data from previous system, whole data file does not belong to the requirements of new developing system. We include or exclude features according to user demand that is also knowns as requirements specification.

➢ *Extracting architecture(FE)*

To understand requirements, the quality approach is to build design for your requirements in form of UML, Flow chart or ERD diagrams that will lead toward valid understanding of system.

➢ *Development(FE)*

Here we perform coding procedure from designing point of view. We also try to develop backend as well as front end of our system.

➢ *Implementation*

After coding we developed a product that is ready to implement as developed output.

➢ Data additions

It is process of making iterative process with many repetition or increments in system according to customer demand

➢ *Re-designing*

When we follow up iterative system, in each iteration we include or exclude some features on demand, so we prepare new design at all.

➢ *Re-coding*

When some modules are added we have to re-code the designed feature as part of re-engineering process [12].

➢ *Re-implementation(after re-engineering)*

In increments product checked by customer and all activities of redesigning,re-coding to implementation repeats until user got satisfied and new system after re-engineering implemented.

➢ *Re-documentation*

After all working activities of re-engineering we prepare new documentation as a detail description of developed product [11].

4. HCI (workflow) with quality assurance reference

Human interact with computers with different ways.

1) HCI after/before re-engineering of software functional point of view (Visible)

Our users interact with system as it should be available to perform different tasks in time and on time. It should not work much faster in sense of quality .Avoidance from unambiguity and remain precise, these mostly make system attractive to users in functional or visible characteristic point of view

2) HCI after/before re-engineering software non-functional point of view (Invisible)

Some of characteristics are invisible for users but relate to major quality of our system as our developed system should be reliable, secure, confident to use, cost effective, easy to use (GUI).The system should increase throughput with availability to solve problems. It show work speedily with less rate of error production with safety.

3) Quality architecture (HCI main areas of focus after re-engineering) Figer2

5. Complete methodology of work

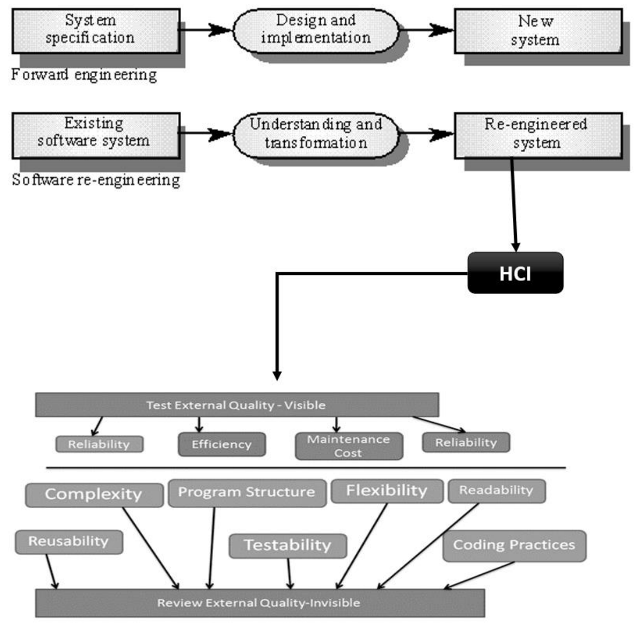

Fig. 2 HCI feature with SRE to QA

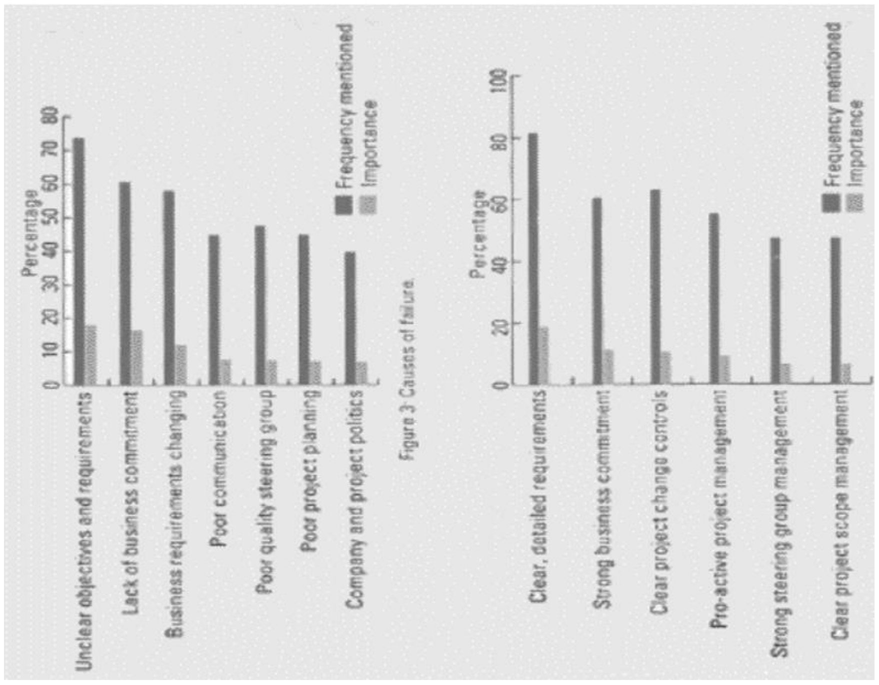

Fig. 3 Software re-engineering (HCL) to Quality attributes

6. Analysis Report of Software quality assurance when compared with re-engineering

Here we analyzed data from more than 100 organizations and the results are available in graph in section-1 of introduction similar results also given in details

7. Conclusion

We conclude that software re-engineering process enhance the quality in sense of human interaction.HCI is major root of quality as user like those products which provide real sense of good representation in form of quality attributes like reliability,reuseabity,maintenance cost and future support in form of effectiveness and risk reduction.Re-engineering process get in use existing products as by the process of re-designing,re-coding,re-publishing and then more features got engaged in existing system. Instead of preparation new product from start up, we reuse existing resources and prepare product in less time, less cost with more user satisfaction and quality demand. Our customer interacted and addicted old systems reason is that, they are working from several years in same environment ,in their point of view the system user interface is real quality that is present in their mind. So, they want to adopt new system qualities as technology, speed, reliability but they don't want to leave inner system. Then in this situation re-engineering support a lot to give quality product in form of user satisfaction (HCI).

Acknowledgement

I'm greatly thankful to Govt college university, Faisalabad, Pakistan for providing such resources. I am also really grateful to my teacher Dr.Awais, Mr. Yahya Saeed for working as supervisor of my research.

References

[1] A. Cathreen Graciamary , Dr. M.Chidambaram , "EESRM: An Effective Approach to Improve the Performance of Software Re-Engineering" ISSN 0973-4562 Volume 13, Number 6 (2018) pp. 3648-3654
[2] Fernando Szimanski,Anivaldo S. Vale," Restructuring Information Technology Area: an experience report in the public service", A Computer Socio-Technical Perspective - Volume 1, Pages 63, Goiania, Goias, Brazil — May 26 - 29, 2015, Tokyo, Japan — March 07 - 11, 2018
[3] Nathan ManeraMagalhães," An Automated Refactoring Approach to Remove Unnecessary Complexity in Source Code", Systematic and Automated Software Testing, Article No. 3,Fortaleza, Brazil — September 18 - 19, 2017
[4] Emile Swarts; "Internal vs External Quality of Software"; written on 29th September, 2015 tagged in Ruby on Rails, Software Architecture
[5] Robert Chatley; Lawrence Jones. "Diggit: Automated code review via software repository mining".IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (2018): 567 – 571
[6] Muhammad Noman Riaz. "Impact of software design patterns on the quality of software: A comparative study". International Conference on Computing, Mathematics and Engineering Technologies (2018):1-6
[7] JyothiVedurada," Refactoring opportunities for replacing type code with state and subclass", ICSE-C '17, Pages 305-307, Buenos Aires, Argentina — May 20 - 28, 2017
[8] Sivaram; "The Myth of Software Reengineering"; Posted On December 24, 2013 by GB Shah filed under Programming
[9] R. Dewar ; A. D. Lloyd ; "Identifying and communicating expertise in systems reengineering: a patterns approach" ; June 1999, p. 145 – 152
[10] NileshJadav,"How To Reverse Engineer Using Advanced Apk Tool", Mar 08 2017, https://www.c-sharpcorner.com/article/how-to-reverse-engineer-using-advanced-apk-tool/
[11] Tahvildari, L., Kontogiannis, K. On the role of design patterns in quality-driven re-engineering. in Proceedings of the IEEE 6th European Conference on Software Maintenance and Re-engineering (CSMR). 2002. Hungary.
[12] S. Ducasse, T.G.ı., and J.-M. Favre, Modeling software evolution by treating history as a first class entity, in on Software Evolution Through Transformation 2004. p. 71–82.
[13] Moghaddas, Y., & Rashidi, H. (2009). A novel approach for replacing legacy systems, Journal of Applied Sciences, 9(22), 4086–4090

Muhammad Muzammul, received his BSIT (2011-2013) from Govt. College University, Faisalabad (GCUF)-Pak. He worked as Lecturer 1.5 year in department of Software engineering GCUF. He worked as websoft trainer in Pro websoft org. He joined Fatima Jinnah college, Pak as lecturer and due to excellent performance he become Head of college in 2017. He started his MS (Software engineering) session 2017-2019 (GCUF), and he is top most student of class with CR ship and excellent in research with area of Automatic Machines learning to artificial intelligence with latest technologies trends and re-engineering.

Ayesha Zafar, Completed her BS (Computer Science) in 2016 from National College of Business Administration & Economics Lahore Pakistan. She started her MS (Software Engineering) session 2017-2019 from Government College University Faisalabad Pakistan. She excellent in research with area of "Software Re-Engineering role in human computer interaction (HCI) with quality assurance"

Muhammad Yahya Saeed received the MSc degree from UAAR, Rawalpindi, in Computer Science in 2003 and M.S degrees in Computer Science from UAF,Faisalabad in 2007. He has also done MSc from UAF, Faisalabad in Statistics in 2000.Currently he is doing PhD from GCUF,Faisalabad in Computer Science since 2015.He is faculty member in GCUF in software Engineering Department

Najaf-Ali-Prince, Completed his MSc IT session (2010-2012) from University of Sargodha, Sargodha (UOS) Pakistan. He worked as Lecturer 2.5 years in department of computer science and Information Technology (UOS) Pakistan. He also worked as Relationship Manager in Alfalah Bank Limited (Lahore) Pakistan for 1 year. He started his MS (Software Engineering) session 2017-2019 from Government College University Faisalabad Pakistan.He is excellent in research with area of Artificial Intelligence applications and sensoring devices.