Investigation of Risks in Reengineering Process Using Grey Wolf Optimization Algorithm

A.Cathreen Graciamary, M.Chidambaram

Abstract: ***One among the principal challenges in software frameworks construction in current days will be quality and functionality. The concept of forecasting the quality of a software product from the higher-level formulation explanation is not the fresh strategy. Software Reengineering is the process of maintaining the software to suit the requirements provided by the user. There will be lot of risks associated with the process of Reengineering.Especially the process of Reengineering encounters the quality and functionality risks. This means that while performing the process of Reengineering failure to maintain the quality and functionality of original system.The utilization of soft computing intelligence strategies to evaluate the risks is the freshly researched area. The term "risk" in developing the conclusions will be customarily utilizedinreplicateambiguitywhich might be considered probabilistically. Here in to analyze the risks involved in the process of Reengineering nature inspired Grey Wolf optimization Techniques is used. By employing the Grey wolf optimization techniques in evaluation of risks in the process of reengineering, searches effectively and identifies the quality risks effectively rather than functionality risks. Though Grey Wolf Optimization algorithm techniques have very good exploration techniques its performance in analyzing the risks associated with functionality risks in reengineering will be little be lower when compared with risks associated with quality risks. Experimental results demonstrated that evaluation of quality risks associated in the process of reengineering process is exceptionally well.***

Keywords: Reengineering, Legacy Systems, Quality Risks, Functionality Risks, Meta Heuristics, Grey Wolf Technique

I. INTRODUCTION

The spirit of software re-engineering will be focused at enhancing or conversion of theprevailing software in order to appreciate, regulate, and utilize the fresh looking one. The requirement of software re-engineering isenlargedto a huge extent,because of the traditional software frameworkis outdated with respect to the structure, the operating system over which they will be executing, and the appropriateness and steadinessin assistingdevelopmentforupdating withfluctuatingrequirements. Software re-engineering will besignificant for the purpose of recuperating and reutilizing the prevailing advantages of the software developed, keeping in control of the expenditure spent over the software upkeeping, and presenting the foundation for upcomingdevelopment of software progression [1]. Economics of Reengineering:

Amount spent for the process of upkeeping = Amount spent yearly for functioning and upkeeping for throughout the lifecycle. Amount Spent on reengineering = Estimated earning with the capital amount spent decreased by amount for employment of alterations in addition with technical parameter associated with risk Amount gained = Amount spent on Reengineering – Amount spent with the process of upkeeping

Software Reengineering is the subdivision of software developments that assists in improving maintainability of software that is going to be developed or already developed. Software Reengineering is explained as the investigation and modification of procedures and develop it into a fresh structure with improved functionality and avoiding faulty operation that existed previously. Reengineering procedure comprises the incorporation of so many sub divisions such as reverse engineering, rearrangement of structure, redocumentation, forward engineering and retargeting. The reason for utilizing the reengineering process in software development and maintenance process is it consumes least amount of expenditure for maintaining the software maintenance. It has to be used when the huge software developed requires to be maintained and updating in the functionality of developed software is needed often. It is involved in maintenance of software with least effort and easier to maintain and update the software. Legacy Frameworks were the conservative software framework that might be compulsory for supporting the business process. Software reengineering process concentrates on reemploying the legacy systems to improve the maintainability of the mentioned system. It is the possible method for assuring the persistence with the operation of legacy systems [2] Because of the fact that the software system is already developed, the effort should be concentrated on reworking certain modules. Therefor the amount of effort to be spent is far less when compared with the designing of new software system.Similarly, the amount spent will be quite least when compared with the developing the fresh software system. The significant motivations of the process of reengineering process is 1. Making the system with the improvement in overall performance 2. Enhance the conceivability 3. TheSoftware Engineering process is divided into various phases as listed below 1. Source code Alteration 2. Reverse Engineering, 3. Program Architecture Enhancement 4. Splitting the Program into various modules5. Data Reengineering.Alteration of Source code is the procedure of converting the source code from one programming language to another.

Revised Manuscript Received on May 06, 2019
A.Cathreen Graciamary, Research Scholar, Bharathiar University, Coimbatore, India.
Dr. M.Chidambaram, Asst prof, Rajah Serofiji College, Thanjavur.

The converted linguistic could be the modernized variety of the actual language or the entirely dissimilar linguistic. Requirement of Altering the source code is listed as follows 1 Improvement of background of hardware 2. In adequate expertise of the employees who are working in the current language 3. Administrational alteration in principles4.In adequate assistance from the software

Reverse engineering is the procedure of investigating software for the motivation of recuperating the formulation, description, conditions for the formulating the software. The program will be unaffected by the procedure of carrying out the reverse engineering process. The source code of the software source code will be frequently accessible. The process of Reverse engineering and reengineering will not be the identicalt one.

The difference between Reengineering and Reverse Engineering will be provided as follows Reengineering is defined as the process of Rearrangement or redrafting of portion or entire framework devoid altering the performance and functional principles. It is Appropriate while certain subdivision of the huge sized framework necessitate recurrent conservation and upkeeping. Reengineering comprises developing the attempt to construct the system with the motivation of being simple in maintaining. The reengineered framework might be modernized and have to be re-drafted.

The decision of opting the process of reengineering will be while framework alterations will be limited to one subdivision, subsequently the subdivision should be reengineered whileassistance provided to the hardware or software systems was outdated. Whileequipment forassisting therearrangementwill beeasily accessible.

Reverse engineering, termed as back engineering, is explained as the procedure of mining the information from the human developed process &replicatingdepending with the acquired knowledge.The procedureregularlycomprisestaking apart of certain hardware or the software program, biological, chemical or organic matter) &investigatingthe constituentsand operation in descriptive manner.Reverse engineering will not refer imitating or altering the manufactured article. It is nothing but merelyscrutinizing the productfor determining the formulationcharacteristics. Motivations to apply for the process of ReverseEngineering: is listed as succeeding if the 1 Credentials that were involved in certain process wasmissing.2. Forinvestigating the operation of the process.3. Toimprove the variety.4. Obtainingdelicateinformation.

Enhancing the program structure will be the procedure for enlightening the assembly of the program to determine the best ways for the memory consumption and the insufficient interpretation of software engineering. Characteristically, programs build the complicated logic framework because they will be altered while performing the task of up keeping. Fresh constraints in addition with related activities will be appended devoid of alteringthe prevailing the regulated framework.

Splitting the program into various models is the procedure of restructuring the program in order to gather the portions of program and taken into account as the solitary module. Diverse categories of modules could be generated are listed as 1. Data summarizations 2. Hardware modules 3. Functional modules 4. Procedureassisting modules.

Data Reengineering is the procedure of investigating and rearranging data assemblies in the framework to develop into the system that is easier to interpret. The reason for performing the process of Data Reengineering is 1 Reduction in the quality of the data 2. Characteristic boundaries which will be developed inside the programs.3 Development in the Structure. The disadvantage of Data Reengineering procedure will be the process is not cost effective and it takes longer duration for completing the task.

The block diagram of Software Reengineering is provided in the Figure 1.

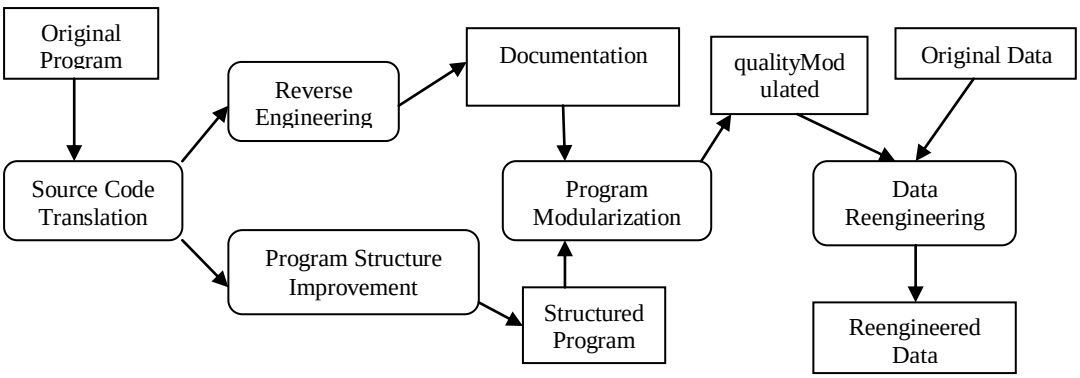

Figure 1 Overall Block Diagram of Software Reengineering

There are abundantencountersconfronted by the procedure of software reengineering. The achievement of reengineering procedure relies with the largerproportion overappropriatepreparation. It will not be the cakewalk forenhancing and transformhuge quantity of programs and databasesdevoid of unsettling data-processing facility. Itwill not be simple to confirm the authentication of such process. Administrationshould beinfluencedby which themanagementwill be actually undergone for accomplishing thenoteworthyadvantage inminimizedexpenditure and appended value.

Taking into account of the above-mentioned concept functionalityof the software keepsuntouched, the mentioned necessitatestheprofound investigation of the desiredsuperiority andenhancin the throughput. Upkeepingmetrics will becompulsoryforconsidering theenhancements [3]. There is, no doubt, a great need toimprove the quality of existing softwareproducts and processes. One wayto accomplish this is through reengineering.Reengineering poses its owntechnical challenges – transformingthe language, extracting objects, reallocatingfunctions, and proving theequivalence of functionality. Therefore, the need of the hour is to analyze the challenges faced by the process of Reengineering in an accurate manner, prior to the application of Reengineering process so that precautionary measures could be considered for avoiding the risks and challenges or tackling the issues that will be encountered while enforcing the process of Reengineering in prior.

II. LITERATURE SURVEY

For the purpose of Software Development Software Engineering is the strategy adopted.Subsequent to the finishing the construction of software, and hander over to the clients for the utilization, Software should be maintained properly for the proper functioning. Because of the fresh necessities by the clients and discovery of malfunctioning of software or letdown by the software in the achievement of goals set by the system, Software has to be modified accordingly with the reference to the short comings.By applying the consecutive upkeeping activities in the constructed software will minimize functionality of software in addition with diminishing the functioning of the system.For enhancing the upkeeping the software framework, while holding the superior functionality of the software, a strategy termed as Reengineering is adopted.On the other hand, the doubt lies in the fact time of application of reengineering process to software. question arises “when to reengineer the software”. In the work bySingh et al., (2018) [4] a background for the application of reengineering procedure utilizing decision tree approach that supports the engineers who involved in developing the software to arrive at a conclusion in such a way to choose between the upkeeping process or reengineering process in terms of various factors such as cost, time etc., For enhancing the superior functioning of the software, Investigation should be performed during the formulation stage devoid of altering the software operation. With the help of computing the difficulty with regards to the reengineering process utilizing six fundamental measurements and analyzing the complexity involved by applying decision tree approach in a swifter way and forecast are developed with respect to the necessities of reengineering or upkeeping for the purpose of retaining the software quality. The structure proposed might develop the baseline for arriving at the conclusions, the development should be continued with the reengineering or upkeeping activities.In addition, the Outcomesmight besimplified with the help of taking into considerations of greater number of software developments that is containing diverse dimensions and the amount of difficulty involved.

Utilization of Agent dependent strategy is being in the development of intellectual software which is having the capacity to determine solutions very though provoking issues was swelled to considerable extent during the previous few Several approaches are utilized in the construction of agent-based frameworks.Nevertheless,many developed techniques have grown to a considerable extent, these techniques keep stressing over the investigations, formulation and employment stage of construction procedure.But the projected techniques were devoid of comprehensive evaluation strategy for construction, assure and authenticated agent dependent framework. Client fulfillment in addition with the amount spent will the greatest significant consideration in the construction technique should be stressed.Therefore, the work by Sivakumar et al., (2011) [5]establishes the re-engineering reliant in the construction of influential agent dependent framework. Classification and Clustering (CAC) dependent data mining procedure is established and utilized inthe evaluation and re-engineer an agent-dependentframework in order to attain the fulfillment by the client. A Client fulfillment rating of 89% to satisfy the excellence anticipation in addition with ameandevelopmentcost fluctuation of only 3%. The obtainedoutcomes are better on comparison with current values maintained by the software engineeringprocess.

In Recent years the application of software in day to dayactivities has become regular part.In the era of software, software upkeeping and updating has become compulsory for making the software that fulfills the requirements by the clients.Several difficulties will be comprised in the maintenance of the developed software that supports the policycomposer should be explained by the software reengineering society. The policycomposerscompulsorily require the support from the skilled person reengineering the software. Numerousanalysis was tried in assistance for arriving at the conclusions. Nevertheless, the availability of inadequateanalysis for the consideration artificial intelligence strategies in providing assistance for arriving at the conclusions.Therefore, Kamaludeen et al., (2011) [6] established aknowledge-dependent conclusionprovisionstructure termed as REXDES which is used in the investigations of diverseconstituents ofthe software application forsimplifying the process of composing the conclusions in the reengineering process of software.The performed analysis supports in assuring the feasibility ofcombiningfreshfurther necessitiesinside the prevailingframework by offering the least amount of supportby the specific software systems engineer.The proposed system concentrates in simplification of theinitial stage reengineering for arriving at the conclusions by IT Experts.Subsequent with the arriving at the conclusion in the process of reengineering is performedin addition with the reengineering functionmight be performed for thespecific demand by the client.The employed software experts in performing the event.

Thus, REXDES reduces the duration in addition with the amount spent on the processeven before the original employment of process.

Code clones are analogousdatabasearchitectures that will be available in several software framework's format.Numerousmethodologieswereprojected that supports in the recognition of analogous code portions in software, will be termed as modest clones. Recognition in addition with followingincorporation of modest clones will beadvantageous for maintaining the software. Additionaladvantages might be accomplished with the help of improving the code clone investigational status. Basit andJarzabek (2009) [7]detectedby which repetitivestructures of modest clones frequentlyspecify the occurrence of fascinatinggreater-stageresemblanceswhich will evokeclone architectures. Structural clones demonstrate the greaterdemonstrationin analoguesconditions on comparison with modest clones only. Existence of logical clustersthat containsmodest clones, structural clones improve the issue in greater quantity of clones characteristicallydescribedwith the help of modest clone recognitionsupport systems, anissue which will be frequently handled by after recognition conceptionstrategies. Recognition of architectural clones might support the process of realization of the formulation of the framework for the purpose of improved upkeeping activity in addition with reengineering with the purpose of reutilizationbetween them.The work by, Basit and Jarzabek (2009) [7] recommends the technique which supports in the recognition of certain categories of architectural clones.Uniqueness of the recommended techniques lies in the fact that comprises the design of structural clones includes the formulation of the structural clone idea using the data mining strategies for the identification of greater stage resemblancesTheyexplained the supporting system termed as Clone Miner whichemploys the suggested strategy by them. Theyevaluated the fruitfulness in addition withextendibility of the projectedstrategies throughnumerous conditionalanalysis. Theyexplainedseveral utilization circumstancesfor explaining the manners of the expertise of structural clones appends worth for the investigation depending on modest clones only.

Software utilizationworries about the expertise of the client who is utilizing the software and the manner by which software provides feedback for the activities.Thework by Pachidi et al., (2014) [8], we present the Usage Mining Method to guide the analysis of data collected during software operation, in order to extract knowledge about how a software product is used by the end-users. Our method suggests three analysis tasks which employ data mining techniques for extracting usage knowledge from software operation data: users profiling, clickstream analysis and classification analysis. The Utilization Mining techniquewill beanalyzed via the model which will beimplementedwith respect to the condition Exact Online, the prominentreal timemonetaryadministration application in the country Netherlands. The assessmentestablished the helpfulportion for the Utilization Mining techniquefor software product administrationin addition with the procedures for construction of the software along with the suitableby recommended data mining techniques for performing the investigation over the activities of utilizations.

El Ramly et al., (2009) [9]established the least weighted strategy in implementing the process of reengineeringby communication between the individual persons and the CPU(HCI) and/or communication between the software frameworks that are developed.Communication process in reengineeringmight beaccomplishedinmodifying thebasis code and formulation strategyleading to the diverseUser Interface (UI), therefore restricting the explanationin the process of communication reengineering strategies whichwill not includethe process of modifying the basis code or interiorformulation of the framework. As an alternative, theyconcentrated over the techniques in addition with strategies in coveringand packing the prevailing interaction stagefor the reproduction using another manner, e.g., over the diversebackground forcombining the legacy frameworkfacilities in other application feasible with the diversestandards for structure, e.g., service-oriented architectures (SOA).Theyutilized the term communicationphaserather than establishment phasesince (1) communicationphasewill not restricted for the client connectionthat might involveadditional sub-phasesuch as Data ExplanationConditions (DEC) basis files for iSeries frameworks for example and (2) communicationwill not restricted for the individual's availablecommunicationphaseon the other hand it additionallyinvolves communicationbetween software frameworks. For instance, it will befeasible for applying the process of reengineering by the manner in which an application might be communicated between additional applications for combiningthisby using other applications on a diversebackground, e.g., Web services. The importantopinionsfor communication reengineering will be (1) the admission point of the legacy framework will be the communicationphase (2) Devoid of code or formulation changes or conversion will be considered (3) the reason for the clients in addition withframeworkcommunicate with the legacy frameworkmight be reengineered by one way or another. Additionally, the old legacy establishment will not be successfulin obtaining the accessionsince this one will be target background (e.g., using emulation), but thismight be optimized, altered, reengineered, etc. for theleast or greater degree. Huge quantity of enhancement was developed in techniques over reengineering of legacy framework in improvingof the procedure that involves therelocation. The time taken by the technique of reengineering will be considerablylarge amount because of the fact its nature of huger scale that might createan effect on assisting the business.The work bySu et al., (2009) [10]establishesconcurrentrepeated reengineering prototype for the purpose of time taken by the reengineering procedure. Proposed prototypeutilizes thetechniquethat contains the idea of RecognizedPerceptionInvestigationfor computing the complicatedadmissionassociationamongst the legacy constituents and exchanged data.

The techniques were established for construction of concurrent agenda in the procedure of reengineering is called as "Bottom up" will be segregated into two portions. The suggested frameworkredcues the processing time of the reengineering procedure was assured as quite competent to enhance the functionality of transition framework by performing the analysis.

The work by Woods et al., (1999)[11] establishes the significance in sharing of system information with the process of reengineering society.In addition, they stressed the prominence of sharing the explanationby the semantics which will be the basis ofmining, interpretation andexpanding the quantity of the materialthat will be desired for sharing by the individual. The semantics will be greater than the outline of the materialthat will be shared. They comprise the foundation for the framework for thearrangement. They also expressed that they have to compulsorily be obvious in the meaning of rebuilding of the architecture with respect to architectural semantics which will be involved in assuming theframework that has to be investigated.Theyestablished thedescription with the background of reworked group of stages depending on CORUM II model.Reliant with the facts by which the framework will be recognized by employment viathegroup of mappings, by which the framework for rebuildingwill bebasically performed with the help of obtaining the mappings again and that these mappings will be reliant over the sequence of applications that contains diverse semantics. Theyproclaim that software material sharing might be taken into consideration alongside exchanging the semantics. They additionallyrecommended that it will besignificant indetermining the causes obviously and meticulously about the realization of the term. "Architecture". Theysuggested that entire software information have to beperformed with the company of set of rules, questions, structures, combinations, etc. whichexplains the semantics of the materialthat will beshared. Theyadditionallyrecommend that a structure for software information sharingobviouslyconsider this one, as meta-data.

The identification of function clones in software frameworkwill be fruitful in consideration of the code and testing the error and upkeeping tasks. The work by Abd-El Hafiz (2012) [12] establishestheeffective measurements-dependent data mining clone identificationstrategy. Initially, standard measurements will begatheredfor the complete set of functionsthat belongs to the software framework. A data mining technique, fractal grouping, will be subsequentlyemployed insplitting the software frameworkwith thecomparativelyleastquantity of groups. Everyclusters that leads in summarizing of functionswhich will beinsidetheparticularclosenessto everyadditional place within the standard measurements space. At last, clone categories, preferred over the pairs, will besimplytaken out from the ensuinggroups. In Bigger software frameworks, the recommended technique will becompetent in space and linear in dimension of the data set. Performance Analysis will be carried out utilizingaverage and huge open source software frameworks. The recommended analysis, determines the impactby selected measurements over the accuracy of identification will be used in examination.

The development of software framework for the past several decades frequentlyresulting inneedlessly complicated and rigidformulationsthat will develop the larger quantity of exertion for the purpose of enhancing and upkeeping. Therefore, the reengineering of object-oriented software develops majority quantity of significance as the quantity, age and dimension heritageframeworksswells to a greater extent. Prominent challenge while performing the process of reengineeringwill be therecognizing and place of formulation where issues will occurthatthwarts the competentadditional progressing of theframework. Still current days the mentioned issuefieldwas not adequatelyassisted, either with the help of strategies, or throughequipment. Ciupke (1999) [13]established the strategy for the purpose of investigating the legacy code, explainingrecurrentformulationissues as questions and identifying the position themanifestations of the mentionedissues in the prototype obtained from source code. The author established hisexpertise with the help of utilizing the tool set thatheemployed for assisting the mentioned activitywith the help of involuntarilyinvestigating theprovidedframework and identifying the explained issues. The authorenforced the tools to analyze the presence of nonconformity with the amount of identifiedformulationinstructionsthat belongs to prevailing source code obtained from numerous investigations performed, in manufacturing and educationalbackground. The investigations demonstrated that the action of issue identification of reengineering processmight beperformed in an involuntary manner with the huge extent, and that the strategyestablished could be competentlyenforced to real world code.

III. PROBLEM STATEMENT

A huge proportion of the software utilized in the industry nowadays is legacy software. Legacy software frameworkswill be system which wasformulated and constructed numerous years back. Frequently the legacy framework emerges into theunsustainable system that thearchitecturecontains very few things to be considered alongsideactual commencement or the assemblypronounced in the framework's credentials.

Contemporary softwareframeworksshouldabide with the necessities, likeelasticity, compliance, and persistent with business procedure reengineering. Motivated with the mentioned necessities, the relocation and incorporation of legacy frameworkin the direction of freshbackgrounds and functioningcircumstancesdeliveringthecompetentapproach for administrationforupkeeping the spirit of the competition. Administrations, thatcombinefreshconstructionalong with the prevailing legacy frameworks, contain thebetter degree of accomplishment. Therefore, it usuallydevelops compulsorycondition for the purpose of reengineering so thatframeworks navigating them towards thefreshbackground and combine with additionalframeworks.

Upkeeping and improving the legacy frameworkswill be the task that has highest complexity, organizationsencountercurrently. Persistent alterations in the techniquesregularlydeteriorates the commercialworth of legacy frameworks that wasconstructedin the past decadesvia larger quantity offunds. Organizationsskirmish with the issue in revolutionizing the frameworks whereasholding the functional characteristicsundamaged. In spite of the outmodedness, legacy frameworkspersistent in offering the benefitviaassistingnovelcommercialprocedures and encompassingpreciousinformation and historical data.

Numerouschoicesprevail for the purpose ofrevolutionizing legacy frameworks, explainedwith massiveknowledgeframeworkthat will be highlycomplex and costlyfor performing the alteration for the purpose of satisfyingthe fresh and continuouslyfluctuatingcommercialnecessities.

Discussions overtransformation oflegacy frameworks could be followed for more than a few years, while reengineering specialistsclaimed whether it will begood enough fortransforming to thehuge, motivation-aware knowledgeframeworkfragmentary or complete one.Redraftingthe legacy framework from the commencement couldgeneratethecomparable functionalityknowledge frameworkdepending over thecontemporary software strategies and hardware. On the other hand,the hugehazard of failure related with huge software framework reduces the probabilities ofobtaining the triumph.

In the work it was recognized to consideration of risk parameters associated with quality and functional characteristic of legacy framework. The work investigates the diversity of risk mechanisms associated to quality and functional scopes in the application of legacy framework. Recognizing and construction of several measures and significant considerations in processing the impression of separate risk constituent will also be explained.

Prevailing quality and functional status of legacy framework will be taken into for the purpose of recognizing andvarious classes of risk mechanisms. The influence of numerousmethodologicalchallengeslikediscrepancyamongst ructure of legacy and target framework, unseparated framework structure, inaccessibility of mandatoryformulation records, huge proportion of associationamong various divisions of legacy frameworkwasenclosed in the work.

This paper aims at optimizing the risk associated with the software reengineering.

IV. OBJECTIVE OF REENGINEERING

The foremost motivation of the process of re-engineering as presented in Figure 2, whichsupportsin realizing the fundamental procedure of converting the legacy framework in to the contemporary form and follows the latest software structure. The procedure of rearrangement, rewriting the codes, reformulations and redraftingwill beenforced over the outdatedframeworkso as to obtain the target system in accordance with the fresh necessities and subsequently re-employ the framework with the outdated or fresh functional characteristic with modern accessible technique.

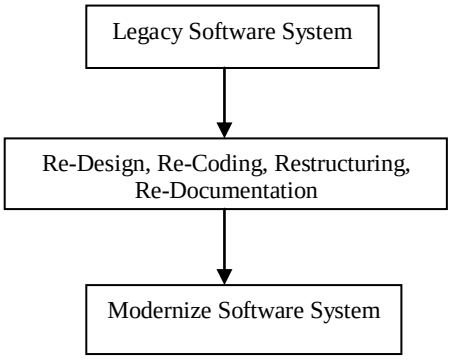

Figure 2: Basic Process of Re-Engineering

Figure 3 demonstrates the custom prototype for software re-engineering which designates the procedures for entire stages of re-engineering depending on the phases of conceptualization utilized in the construction.

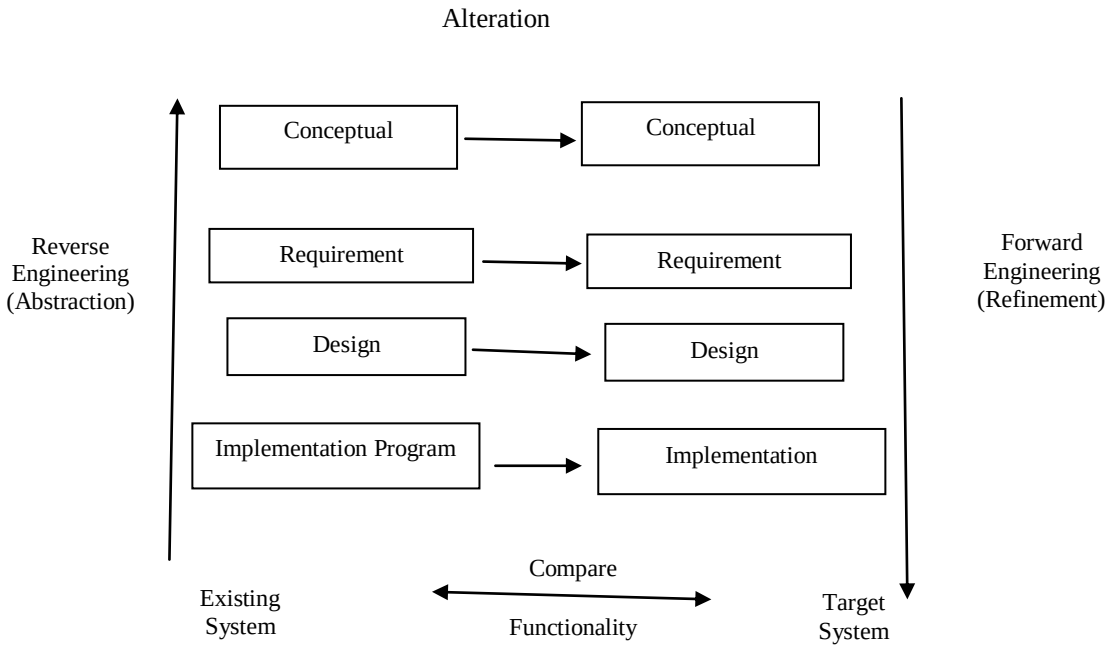

Figure 3: General Model for Software Re-Engineering

V. RISKS IN RE-ENGINEERING SOFTWARE

A) RISK:

Risk will be the probable underperformance among the expected status of functionality and the original functionality during the upcoming time.

B) Risk Evaluation

A methodicalstrategyfornoticing the functioningapproaches, anticipation of malfunctioning and determiningtheappropriatesupervisingtechniquefor theaverting thedamageto thecommercial process and nonconformitywith thefunctional locationwill betermed as evaluation. The evaluation must compulsorily comprise the regulationsneededfor removing, lessening or diminishing the risks.

C) Controlling Risks:

A Frameworkshould be awareof the risks that mighttake place during any time and at any location. Forregulating the risks the trailing valuesmust be enforced to the framework.

1. The risks must becircumvented or eradicated 2. Theconcerned portion of the structuremust beinvestigated.3. Theprecautionarystrategy for regulating the risks should be considered 4. Methodologicalapproachshould beemployed forregulating the risk[14]

Knowledge, training and making the framework to learn, and deliveringindividualdefensiveprocessmust be taken into account as the finalalternativesubsequently tomentionedregulationswas taken into account. Employers must provide the importance to the mentioned regulations whenever feasible the which will prevent the entire framework from uncovered to the risk in the better manner.

D) Risk Handling

While the risks are detected in theframework, the process of removal will be performed. With the repetition, method of determining the risks and overcoming will be the simpler task.

E) Risk Analysis

Risk analysis has become a standard requirement for legacy process transformation projects in many organizations. The basic thing is to identify, analyze, classify, and categorize the risks involved. When the risks are systematically listed out and categorized, the mitigation process become easier and economically sound in profit. Any reengineering system faces some risks when the process transformation takes places. Such risks are as follows.

- Mission and Goal
- Program management
- Decision derivers
- Organization management
- Customer/User
- Project parameters
- Product content
- Deployment
- Development Environment
- Project Management
- Project team
- Technology
- Maintenance

VI. THE RISKS ASSOCIATED IN RE-ENGINEERING SOFTWARE

Even though re-engineering will befrequentlyutilized method forminimizing the risk, diminish the amount spent and upkeeping the function of formulating the software, on other hand re-engineering could notcircumvent risk. The risk of detectingthe programs at the premature stageforassisting and project administration in the groundwork of constructing the approximations, risk evaluation of software re-engineering and delivertheappliedsystem for the opportunities. Detecting the risks will be compulsory for competent risk evaluation, risk investigation and risk administration. Whensupervising the project in theincessantstyle, the detection of risks might beperformed during the primaryphases. The risk will bedetected the mitigation procedurewill besimple with the skilledgroup and thereforecircumventing the issue in purposefulprocedures. The upkeeping portionwill besimple. Thehandling of risk is presented in Fig.4 comprises of the succeedingphases as listed. Classification, Identification, Assessment, Mitigation, Monitoring.The process can be done in the following manner,

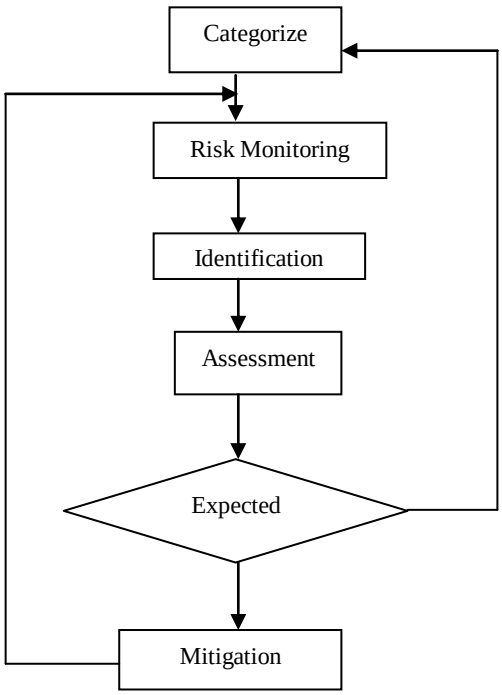

Fig 4. Risk Monitoring Methodology

Whatsoever the risks mightdevelop, risks can be classified into the inadequateseparations. Throughincessantsupervising of risk,it is possible to detect the risk while it transpires. The evaluation of might be performed and examineif the risk is recognized or unidentified risk. If the risk is recognized and anticipated, handled it with skilledsquadthen, the unidentified risk must be classified and navigated back to risk supervising framework.

Software re-engineering will be focused on overcoming the risks and minimizing the amount involved in the process of functional and upkeeping on other hand legacy conversion contain numerous risks encompassed in the construction of framework. The influence of the risks will be much worser in comparison with the traditional construction strategies since development because re-engineering the prevailingstructures comprisemodification in purposefulprocedureswhichwill be essential partalongsidepresent tendency and currentobtainable techniqueforcommercialfunctions. Risk Detection at the primary stagesupports programmers and project managers in formulating for approximation and assessment of software re-engineering risks and deliverstheachievable and representative framework for prospects. Risk Detectionwill becompulsory for efficient risk evaluation, risk investigation and handling of risks.[15]The motivation of the analysis will be in recommending the technique for supervising the reengineering software, for detecting the risks as soon as it transpires and classify them and handle with suitable solution. This will result in the reengineering software for minimizing the risk and preparing it with the commercial way i.e. cost-effective manner

VII. QUALITY AND FUNCTIONAL RISK ISSUES IN REENGINEERING PROCESS

A legacy frameworkconsistingfundamentalpurposefulcompetence and essential quality statuswill be thebetter contender for reengineering. Reengineering mightenhance quality and functional competence of such legacy framework in aneconomicalway. Butif the legacy frameworkcontaining underprivileged functional and quality competence the risk of reengineering intensifies. The reconstruction choice will benormally utilized for such category of legacy frameworks.

It is possible to detect and consider the effect of riskbecause of the functional capability and quality parametersthatcomprisesconsistency, obtainability, usefulness, abiity to split and integrate separate modules and functionality of the legacy Framework. Likelihood of reengineering accomplishmentintensifiesalongside legacy frameworkcontainingsubstantial quality stages. Effect of discrepancybetween legacy and target frameworkstructure, hugeproportion of connection, complicatedclient interaction along with thestage of consistency and obtainabilitywill beconsideredinside the background of technologicalarea. Detected risk mechanisms and comparative risk parameterssupportsininvestigation of quality and functional competence of prevailing legacy framework. [16].

VIII. QUALITY ISSUES IN LEGACY SYSTEM REENGINEERING

Measurable quality procedureswill be very muchsupporting in the accomplishment of triumph in reengineering procedure. Realizing the parameterswhichencourages software quality and several quality motivationsmightassist in software engineers and executivesfor considering thebetterknowledgeableconclusion which helps in enhancing the software outcomevia reengineering [17]. It is possible to detect and construct risk mechanisms and effect consideration strategy for quality portion will be as trails as follows:

A. *Reliability Risk Component*

Dependability will be the complicated perception which must be measured at the system instead of separate individual component level, since the mechanisms in the framework will be codependent [18]. The stage of dependability risk impacts the complete amount spent and timetable for the development of legacy framework thru reengineering.

B. *Usability Risk Component*

The usability risk component characterizes the damage accompanying with disappointment of client because of incompetent and complicated framework provision and communication with the clients.The mentioned risk componentexplainsinopportuneness and clumsinessinutilizations [19].

C. *Performance Risk Component*

Functionality of the legacy framework will be explained with thesatisfactory conduct by theframework which motivates atdevoid ofusage of excess resources, like memory storage and consumption of processor effort, network bandwidth, durationetc. [20]. Performance risk constituentexplains the damagerelatedalongsideinappropriate resource consumption and enormousduration of answering. Leastproportion of computingswiftness, throughput and competencewillinfluence the of performance risk constituent. Functionality of the frameworkwill be the significantchallenge in accomplishment of triumphof the software framework. Nevertheless, numerous legacy software frameworks will fall back in satisfying thenecessities their functionalitymotivations while the mentioned will bedeveloped during the starting time.

D. *Modularity Risk Component*

The division characterizes thegroup of associatedapprehensions. Combinationwill be theconsideration of interrelationshipbetween division with the architecture of program. Connectionrelies with the communication complicatednessamong the divisions, the place at which admission or situationwill bedeveloped by the division, and data navigation around the interface. [21].Modularity risk constituentcharacterizes the risk of damagerelatedalongside the unsplitted legacy framework into module containing the huge proportion of connection among the diverse divisions.

E. *Availability Risk Component*

To reengineer the legacy framework successfully, the fundamental structure that depends over the administration arrangement must be compulsorily utilized for supporting the organized reengineering.Further, the subsequentlylisted ones should be utilized: investigation and formulation of records, amount spent over the data for the purpose of upkeeping the system, satisfactoryarrangement ofadministration, and project handlingpreparationcompetences. If the mentioned proficiencieswill not be obtainable, the reengineering attemptdevelopsinto damaged and confusing. Availability risk constituentcharacterizes the damageaccompanying with inaccessibility of obligatoryformulation and modification in the administrationof record of legacyframework [22].

IX. FUNCTIONAL ISSUES IN LEGACY SYSTEM REENGINEERING

Huge stageinmethodical risk influenceconsiderationcontains the significanteffect over the triumph or defeat of the reengineering strategy. Nevertheless, the mentionedapproaches might be criticallydefective or partialbecause of inadequate focusin functional risk characteristics of legacy and target system. It is possible to investigate the communication relianceamongseveraloperations of the legacy framework [23].

Measurement of riskeffectbecause of the diverseoperational challengessuch asstage of learning for prevailing workforce, concentration of numerousoperationalcharacteristics comprising shortcomings, rate of error, rate of duration taken for the answer and duration spent for upkeeping. For taking out of the mentioned challenges, it is explained about the five diverse functional risk constituents as trails.

A. *Software Architecture Risk Component*

The software structure of theprocessing system will be the architectures of the framework that encompasses software constituents, the superficiallyobservablecharacteristics of the mentioned constituents, and the associationamong them. Letdowninassessing the prevailingstructure and discrepancyamongcurrent and target structuremightupsurge the influence of software structure risk constituent. Software structure risk constituentcharacterizes the risk of damageaccompanying with discrepancyamongprevailingstructure of legacy framework and anticipatedstructure of target system [24].

B. *Complexity Risk Component*

Difficulty risk constituentwill be the risk of damageaccompanyingalongside complicated structure of legacy framework with hugestatistics of reliableconnectionsinside the framework along with the peripheralobjects. Existence of numerouscharacteristics of theframeworkwhich mightspecifythe difficult nature like thequantity of administrativeblockscontained, quantity of participantscontained, quantity of classifiedstagesengaged by the clients, ambiguousnecessities, fluctuatingopportunity, quantity of complicated implementation with complicatedassociation.

C. *Maintainability Risk Component*

Maintainability risk constituent characterizes the influence of compressed legacy framework or legacy framework containing inappropriate upkeeping events that might intensify difficulty of architecture of legacy framework. Maintainability effectconsiderationcomponentinvestigatesstructure of legacy frameworkfordetecting theamount of divisions appendedremoved and modified subsequent to theinitial release of the legacy framework.

D. Training Risk Component

Training risk constituent explains the risk of damage accompanying with strangeness of the prevailing work force on cutting-edge equipment and strategies utilized in accomplishment target system objectives. Training risk consideration strategy investigates the necessity of personalized and dedicated learning programs and distinct accessing facilities for prevailing workforce [25].

E. Technology Risk Component

Reengineering encompass methodical conversion of the legacy framework to thefreshstructureforunderstanding, enhancement of quality enhancement in functioning, system competence, functionality, performance, or improvability with the least amount spent, agenda, or risk for the customer. Technology risk constituentharmonize with the opportunityby which the current status of legacy framework and administrationmight fall backinassisting thesophisticated techniques and equipment utilizedinsatisfying the necessities of target system. The stage of worsening and outmodedness in legacy frameworkimpact the completeinfluentialworth of technique risk constituent.

X. REASON FOR USING THE NATURE INSPIRED TECHNIQUES IN THE RISK MANAGEMENT

Particularly, subsequent with the accomplishment of genetic algorithms, that was inclined in the direction of inherent choice mechanism, for the past few years was observed with progressing prominence in the field of computer science and engineering societies over the analysis about nature-influenced processing. Substantial quantity of techniques copying the certainstrategies from the nature provided thebroad range of arenas.

A) Portfolio Optimizations

Portfolio will be explained as distribution of research between numerous functions. Portfolio optimization, that establishes the superlative distribution of resources for prevailing functions, was considered to be the significant investigation areas in contemporary risk administration.

B) Nature-Influenced Optimization Techniques

Nature-influenced optimization techniqueswill be under the category of metaheuristics.The metaheuristicswill be explained with certain approaches inspired withprevailing conducts/mechanisms for the purpose of accomplishment of the solution of an optimization-likeissue in nature.In computer science, the metaheuristic refers in the explanation of the processingtechniquethatdetermines the best solutions with the help of repetitivelyattempting inenhancing the solutionin accordance with theprovided consideration in quality. These techniques will be methodical trial-and-error strategies.

Metaheuristics (certain times explained as devoid of derivative, straightawayexploration, black-box, or certainlymerely heuristicapproaches)develop a very little or devoid of presumptions about the problem that will be aimed to determine the best solutions and might explore huge space membersolutions. In addition, with majority of thementioned techniques will be inherently flexible for concurrent processing, thatdevelops them into suitable to in very huge level problems. Nevertheless, it has to be observed that metaheuristics will not assure the provision of best solution can be identified. But, for everytechnique, abundantquantity of analysis wasperformed so as torealize the manner the technical characteristics must befine - tuned for the purpose of enhancing the likelihood ofaccomplishment of the triumph.

Actually, metaheuristics technique wassuggested for conjunctional optimization by whichthe best solution will be identified with the independentexploration-space. Widespread metaheuristics utilizedconjunctional problems comprise simulated annealing,tabu search, genetic algorithms, and Ant Colony Optimization.In real valued problems, the traditionalstrategy will be for obtaining the gradientof the function to be optimized, and subsequently implementation of gradient descent or a quasi-Newton technique.

Metaheuristics will not utilize the gradient or Hessian matrix; therefore, the benefit will be the function to be optimized not required to be continuous ordistinguishable in addition with, it might contain conditions. Popular metaheuristicoptimizers for real-valued spaces for exploration compriseparticle swarmoptimization differential evolution andevolution strategies

Entire technique of the mentioned category wasprimarilysuggestedfor single-objective problems.Nevertheless, during the years, multi-objective prolongations of thementioned techniques wasprojected. Initial Efforts will be the prolongation of simulatedannealing,tabu search. On the other hand,the analysis with respect to population-dependentmetaheuristics should be taken into account.

In this paper, we will focus on the nature-inspired metaheuristics of Greywolf technique and theirapplications to the risk management that involves in analysis of risks and risk optimization problem.

XI. GREY WOLF OPTIMIZATION TECHNIQUE

A. Overview

The area of optimization will be thesignificantinvestigationfieldobligatory by all studyarea. It relies with the human intellectin investigation and demonstrating the conduct of nature, insects, animalsand numerouscategories of living being and phenomena's, in theadaptable andeasier expressions. Lately, aninnovative meta-heuristic termed as Grey Wolf Optimizer projected by Mirjalili et al. [62] in 2014.TheGrey wolftechniqueencouraged from the actualhunting mechanism of the structured group of Grey Wolf by investigating the performance and interactionfor hunting in nature.

B. Search Mechanism of GWO

Usually thefamily of grey wolves will begeneratedamong 5 and 12 individuals.The collection of wolves will be well structured and typicallyframed with the help of trailing synchronized sub groups.

1. Alphas agent: The group is directed by thesuperior leader (male orfemale), called alphas. The foremostactivity of the alphas will be to developthe most influentialchoices for hunting, selectingsetting fornapping, to protect the group.

2. Beta agents: in nature and in numerouscommunal lives, one can determine theconsensus, in the arrangement of grey wolves; a second groupingtermed as betas operates in cooperation with the initial group, agents from this sub group support the superior (alpha) in development of conclusions for the accomplishment of the anticipated motivations of the group. Consequently, the expertise of superiorwill beenhanced by the response transported from related agents of this second grouping.

3. Delta agents: termed asassistant, wolves inside thegroupingshould be motivated in accomplishment ofnumerousactivitieslike: scouts, sentinels, elders and hunters, they should exchange the concepts to alpha and beta to assure the protection of the group.

4. Omega agents: the omega takes thelowermostposition Grey Wolf in the grouping; they will be the final category of wolveswhich will bepermitted forconsumption. The omega takes the role of scapegoat, the wolves of thegroupingshould obey the instruction from the all the other individuals from superiorgrouping.

In unfamiliar nature, hunting will be the significant communal activities of Grey Wolf families for endure inside the diverse conditions and significant circumstances. The trailing explanations abridge the foremoststages of Grey Wolf hunting mechanism 1 Following, hurtling and impending the prey.2Trailing, surrounding, and irritating the prey still it terminatesthe navigation.3 Attack in the direction of the prey.

C. Mathematical Model of the Grey Wolf TechniqueSearching for Prey

Exploration procedure started with any arbitrary way of beginning of candidate solutions from the examination space. They deviate from one another to probe the target and congregate after they identify it.

i) Encircling Prey

Subsequent to identification the target, grey wolves enclose the target. Encircling conduct might be signified as specified below.

$$\vec{E} = |\vec{O}.\overrightarrow{X_p}(i) - \vec{X}(i)| \quad (1)$$

$$\vec{X}(i+1) = \vec{X}_p(i) - \vec{B}.\vec{E} \quad (2)$$

Here recent iteration is characterized by $\vec{B}$ *and* $\vec{O}$. are the coefficient vectors. $\vec{B}$is utilized for upholding the distance between search agents' grey wolves (GW) and the target. $\vec{O}$signifies hindrances in the attacking route of the grey wolves whereas tactics to reach the prey. $\vec{X}$signifies the position vector of the grey wolf in addition $\overrightarrow{X_p}$designates the position vector of the prey.

The coefficient vectors $\vec{B}$ and $\vec{O}$ expressed as given in the equations respectively in (3) and (4):

$$\vec{B} = 2 * \vec{l} * \overrightarrow{r_1} - \vec{l} \quad (3)$$

$$\vec{O} = 2 * \overrightarrow{r_2} \quad (4)$$

where the component $\vec{l}$shrinkages proportionally from 2 to 0 at the time of iterations. $\vec{r}_1$1 and 2 are random vectors in the interval [0, 1].

ii) Hunting

Subsequent to the surrounding of the prey, they are focused in hunting the target. The process of hunting normally directed with the help of α, β & γ categories wolves. Out of these, a delivers the finest candidate answer. Mathematically, hunting performance of grey wolves is expressed by

$$\vec{E}_\alpha = \left|\vec{O}_1\vec{X}_\alpha - \vec{X}_i\right| \quad (5)$$

$$\vec{E}_\beta = \left|\vec{O}_2\vec{X}_\beta - \vec{X}_i\right| \quad (6)$$

$$\vec{E}_\gamma = \left|\vec{O}_3\vec{X}_\gamma - \vec{X}_i\right| \quad (7)$$

$$\overrightarrow{X_1} = \vec{X}_\alpha(i)\text{-}\overrightarrow{B_1}.\overrightarrow{E_\alpha} \quad (8)$$

$$\overrightarrow{X_2} = \vec{X}_\beta(i)\text{-}\overrightarrow{B_2}.\overrightarrow{E_\beta}) \quad (9)$$

$$\overrightarrow{X_3} = \vec{X}_\gamma(i)\text{-}\overrightarrow{B_3}.\overrightarrow{E_\gamma}) \quad (10)$$

$$\vec{X}(\text{i+1}) = \frac{(X_1+X_2+X_3}{3} \quad (11)$$

iii) Attacking prey

Subsequent to the accomplishment of hunting, they try to ambush the target. Permission of the search agents by the GWO techniques i.e.., wolves try to improve the locations by itself for attaching the prey is provided as per the position of α, β and γ class of wolves. It supports in generating the dependable solution. For explaining the model for reaching the target, two vectors, $\vec{a}$ and $\vec{A}$ are taken into consideration. Therefore, $\vec{a}$proportionally shrinkages from 2 to 0 while the iteration upsurges in addition to the oscillations of $\vec{A}$is also diminished in accordance with the $\vec{a}$. But, $\vec{A}$is an arbitrary value between [-a, a] When arbitrary value of $\vec{A}$is between [-1, 1], the subsequent location of the agent might indicate between its present location and target location.

XII. EXPERIMENTAL RESULTS

Grey wolf algorithm is employed in the analysis of risks involved in the reengineering process of modernizing the legacy systems.Grey wolf techniques will be the population-dependentdevelopment-supervised stochasticexploration strategy. A population of probablesolutions is prepared—generally atarbitraryon the other handoccasionallywith the help of integratingareaexpertise to keeppossibilityof certain conditions. This is trailedby a recurrentarrangement of assortment andrecombination functions.

Nature-inspired techniques contain a little quantity of certainexclusive parameters thatdevelopsintothe appreciated strategy utilized in risk analysis:They are adopting multipurpose strategy. It will notbeneeded to contain the comprehensive strategy of the area they are utilized for. They do not require adjustment. In severalcircumstances of risk analysischallenges, comprehensive knowledge of the circumstance is not available. This will not result incomplexity for nature-inspired strategies.

Further,the mentionedmethodswill bestrong in such a way they might alter according to the modifications in the circumstances. The mentioned techniquewill beespeciallyvital in the risk analysis.

Risk detectionwill becompulsory for competent risk evaluation, risk investigation and risk administration.The probable risks might be classified in the subsequent areasas represented in figure 5 and are deliberated in the result analysis for generating the evaluation report.

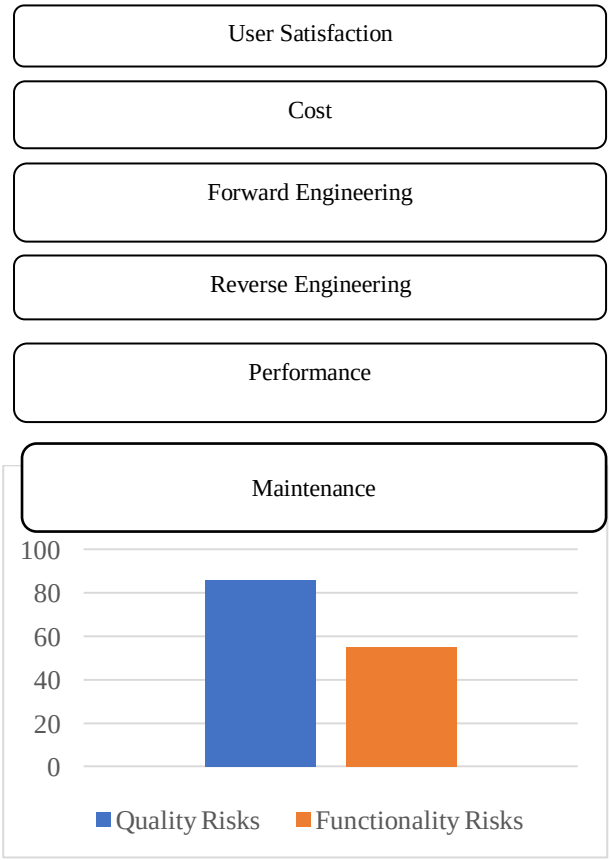

Figure 6. RiskEvaluation of Quality and Functionality Risks Analyzed with the Grey Wolf Technique

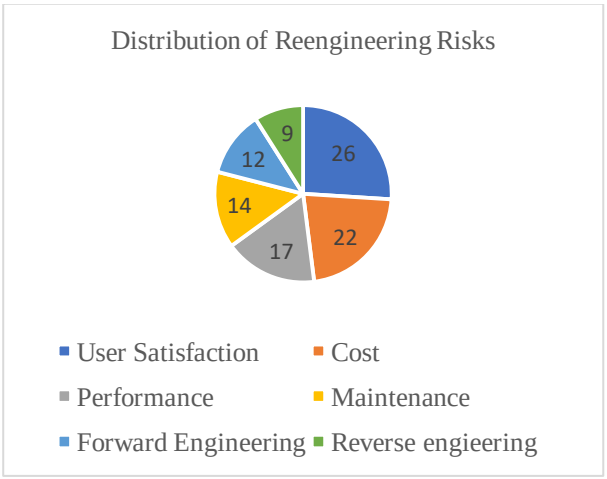

Figure 7 Distribution of Reengineering Risks

Figure 6 demonstrates analysis of Risks associated with the Reengineering process in modernizing the legacy systems. From the figure 6, using grey wolf optimization technique the probability analyzing the quality risks will be much more than the functionality risks.

Figure 7 will be detailed in the subsequent secion.

User Satisfaction: Clientfulfillmentwill bethe essential significantcomponent for any commercial approach, consequently it will be indispensable forany commercial to efficiently handle consistent considerations forclient fulfillment. In software re-engineering the clientfulfillment risks will be listed as below [29].1Inadequate of client compatibility 2.Excess cost Budget in un-administered procedures 3.Unanticipatedoutcome of the target system4. Unsubstantiated to referential prototype.

Cost: Legacy software will be reengineered so as to encounter the commercial competition with UpToDateknowledge and tacklesfordeveloping it into cost effective. Risks comprised in cost advantagewill bespecifiedas listed:1 Leastadvantage from the cost of re-engineering 2 Huge amount spent withupkeeping subsequent to reengineering3. Affluent backup 4. Huge amount will be spent for the generation of commercial report5Underprivileged quality procedures for re-engineering and discrepancy of commercialprocess6Damage of reserves on legacy conversion.

Forward Engineering: Forward Engineering will be theconservative strategy fornavigating from high-level summarizations and logical, employment-independent formulations in the physical employment of the system. Risks comprised in FE are provided as trails.1Heldsubjectswill not combine to fresh framework 2. Complexity in driftingprevailing data to thefreshframework 3Extent of planning for conversion and reverse engineering will not be adequate.

Reverse Engineering: Reverse engineering (RE) will be the procedure of investigatingthe subject software framework to (1) detectthediversecomponents and the intersocietyand (2) and to characterize the framework in additional manner that is visible orgreaterstatus of thought. The RE risk parameters are:1. Summarizedknowledgecould not be articulated in the considered language for necessities and strategystipulations.2 It will befairlyproblematic to apprehendcompetentstrategy and limitednecessities from the source code. 3Prevailingcommercial expertisebuilt in source code will bemisplacedbecause ofunsuitableprocedures. 4ImprovedKnowledgewill not fruitful or not utilized.

Performance: The re-engineering procedurereliesprofoundlywith the functionality of revolutionizeframework. It will be the extent of ambiguitywhich mightretain the frameworkforsatisfying theproceduralstipulations or which mightlead the framework not fulfilling the fundamentalobjectives. Functionality of the freshframework must undeniably be improvedin comparison with the legacy framework. The Performance risks parameterswill beprovided in the followings.1 Non transportabilityoffresh framework 2Outcome not suited with the precedingframework 3. Consistencydiscrepancy 4Unsuitable Re-engineering method and data re-arrangement

Maintenance: Maintenance will be the importantparameter to be taken into account in software re-engineering. Conclusionswill besupportedwith the help of realizing materializesfor software frameworkswithdurationin accordance with the freshnecessities. The significant software upkeepingchallenges could be either administrable ormethodological or both. The upkeep risks will bepresented asbelow.1Timed Backup 2Retrieval of legacy frameworks 3.Inappropriate Re-documentation and data rearrangement.

XIII. CONCLUSION

Currently, alongsideprogressing development in the comprehensive techniques, handlingthe much complicated and ambiguous challengeswill beunavoidable. So, the requirement of techniques thatrealizes complicated and abstrusechallengesto be identified.Soft computing will be the one among the competentassistances for investigating risks involved in the process of software engineering andreengineering procedures.Risk management is the prominent challenge containing difficulty and ambiguity in the assessment and examination of issuesbecause of the impact of diverse parameters.Currently, simultaneousdevelopment ofambiguous and indeterminatechallenges the employment of soft computing techniquesaugmented. Significanttechniquesin soft computing is the nature inspired optimization techniques. Here Grey wolf optimization technique is used in analysisof risks that is associated in the reengineering process applied for legacy systems. From performing the experiments, it is observed that the analysis of risks associated with the quality of legacy systems will be better performed than analysis of risks associated with functionality. In future risk optimization work can be carried out using evolutionary techniques.

REFERENCES

1. Rosenberg, L. H., & Hyatt, L. E. (1996). Software re-engineering. Software Assurance Technology Center, 2-3.
2. Almonaies, A. A., Cordy, J. R., & Dean, T. R. (2010, March). Legacy system evolution towards service-oriented architecture. In International Workshop on SOA Migration and Evolution (pp. 53-62).
3. Sneed, H. M. (1995). Planning the reengineering of legacy systems. IEEE software, 12(1), 24-34.
4. Jaswinder Singh, Kanwalvir Singh Jaiteg Singh, (June 2018) Reengineering framework for open source software using decision tree approach, International Journal of Electrical and Computer Engineering (IJECE) Vol. 9, No. 3,
5. Sivakumar, N., Kalimuthu, V., & Gunasekaran, A. (2011, December). CAC Based Data Mining Workflow to Test and Re-engineer Software Agents. In International Conference on Computing and Communication Systems (pp. 139-148). Springer, Berlin, Heidelberg.
6. Kamaludeen, R. A., Sulaiman, S., & Cheah, Y. N. (2011, December). Expert-based decision support framework for software reengineering. In 2011 Malaysian Conference in Software Engineering (pp. 325-329). IEEE.
7. Basit, H. A., & Jarzabek, S. (2009). A data mining approach for detecting higher-level clones in software. IEEE Transactions on Software engineering, 35(4), 497-514.
8. Pachidi, S., Spruit, M., & Van De Weerd, I. (2014). Understanding users' behavior with software operation data mining. Computers in Human Behavior, 30, 583-594.
9. El-Ramly, M., Stroulia, E., & Samir, H. (2009). Legacy systems interaction reengineering. In Human-centered software engineering (pp. 316-333). Springer, London.
10. Su, X., Yang, X., Li, J., & Wu, D. (2009, October). Parallel iterative reengineering model of legacy systems. In 2009 IEEE International Conference on Systems, Man and Cybernetics (pp. 4054-4058). IEEE.
11. Woods, S., Carriere, S. J., & Kazman, R. (1999). A semantic foundation for architectural reengineering and interchange. In Proceedings IEEE International Conference on Software Maintenance-1999 (ICSM'99).'Software Maintenance for Business Change'(Cat. No. 99CB36360) (pp. 391-398). IEEE.
12. Abd-El-Hafiz, S. K. (2012, July). A metrics-based data mining approach for software clone detection. In 2012 IEEE 36th Annual Computer Software and Applications Conference (pp. 35-41). IEEE.
13. Ciupke, O. (1999, August). Automatic detection of design problems in object-oriented reengineering. In Proceedings of technology of object-oriented languages and systems-TOOLS 30 (Cat. No. PR00278) (pp. 18-32). IEEE.
14. Joshphin Jasaline Anitha.X M. Karthika K. Alagarsamy (2012), The Classification and Analysis of Risks in Reengineering System, International Journal of Computer Applications (0975 – 8887) Volume 39– No.18, February 2012
15. Rashid, N., Salam, M., Sani, R. K. S., & Alam, F. (2013). Analysis of risks in re-engineering software systems. International Journal of Computer Applications, 73(11).
16. Rajavat, A., & Tokekar, V. (2012, April). RrMm-A Measurement Model to Quantify the Effect of Reengineering Risk in Quality Perspective of Legacy System. In International Conference on Advances in Information Technology and Mobile Communication (pp. 9-16). Springer, Berlin, Heidelberg.
17. Rajavat, E. A. (2011). An impact-based analysis of software reengineering risk in quality perspective of legacy system. International Journal of Computer Applications, 975, 8887.
18. Gui, G., & Scott, P. D. (2008, November). New coupling and cohesion metrics for evaluation of software component reusability. In 2008 The 9th International Conference for Young Computer Scientists (pp. 1181-1186). IEEE.
19. Kan, S. H. (2002). Metrics and models in software quality engineering. Addison-Wesley Longman Publishing Co., Inc.
20. Cortellessa, V., Goseva-Popstojanova, K., Appukkutty, K., Guedem, A. R., Hassan, A., Elnaggar, R., ... & Ammar, H. H. (2005). Model-based performance risk analysis. IEEE Transactions on Software Engineering, 31(1), 3-20.
21. Gui, G., & Scott, P. D. (2008, November). New coupling and cohesion metrics for evaluation of software component reusability. In 2008 The 9th International Conference for Young Computer Scientists (pp. 1181-1186). IEEE.
22. Reliability Hot Wire eMagazine, (2003) "Relationship between Availability and Reliability," Reliability Engineering Newsletter, Issue 26, April 2003.
23. Er. Anand Rajavat, Dr. (Mrs.) Vrinda Tokekar, (2011), "Identification and Measurement of Functional Risk Components in Reengineering Process of Legacy System", Published in International Journal of Advanced Software Engineering (IJASE), ISSN 2249-3069 Volume 1, Number 1 © Research India Publications, pp. 11-25.
24. McCabe, T. J., & Butler, C. W. (1989). Design complexity measurement and testing. Communications of the ACM, 32(12), 1415-1426.
25. Brinkerhoff, R. O., & Dressler, D. (2015). Using evaluation to build organizational performance and learning capability: A strategy and a method. Performance improvement, 54(7), 37-44.
26. Rajavat, A., & Tokekar, V. (2014) Investigation of Quality and Functional Risk Issues in Reengineering Process of Legacy Software System.International Journal of Programming Languages and Applications (IJPLA) Vol.4, No.1
27. Yilmaz, A. E., & Weber, G. W. (2011). Why you should consider nature-inspired optimization methods in financial mathematics. In Nonlinear and Complex Dynamics (pp. 241-255). Springer, New York, NY.
28. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in engineering software, 69, 46-61.
29. Chiang, C. C. (2007, August). Software stability in software reengineering. In 2007 IEEE International Conference on Information Reuse and Integration (pp. 719-723). IEEE.