IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 3, March 2015.
www.ijiset.com
ISSN 2348 – 7968

An Efficient Requirements Engineering Framework for Software Quality Improvement using Optimization

[1]Mr.R.Ramkumar, [2]Dr. G.S AnandhaMala
[1]Research Scholar, Dept. of Computer Science and Engg, St. Joseph's Institute of Technology, Chennai, India

[2]Professor & Head, Department of Computer Science and Engineering, Eshwari Engineering College,Ramapuram,Chennai

Abstract—In recent years, reports of software security failures have become common place. One source of security problems is not considering the security requirements of the complete system. Due to the emergence of markets for off-the-shelf or packaged software, market-driven development is gaining increased interest in comparison to customer specific system development. As a consequence a shift in focus is occurring, affecting software development in general and requirements engineering in particular. Requirements Engineering (RE) lays the foundation for successful development projects regarding cost and quality. Even if a company defines and integrates an RE process for a broad use in different projects, it still does not address the various influences that engineers have to face in volatile project environments. Requirements engineering (RE) constitutes an important success factor for software development projects, since stakeholder-appropriate requirements are important determinants of quality. Incorrect or missing requirements can greatly add to the implementation or maintenance effort later.
Software can be made successfully when it proves to have better quality criteria. . The major goal is to reduce the cost required for software so that the quality can be improved efficiently. To initiate my work,I have selected the requirements for designing the software in such a way that we can improve the quality of designing.Quality improvement in requirement engineering is mainly consideredmy work where I am designing a Requirements Engineering model which uses an optimization algorithm which optimizes the Non-Functional Requirements to reduce software cost production and improve the quality.

Keywords:-Requirements Engineering, Optimization, Quality

I Introduction

Software systems are subject to security threats, which may influence organizational assets. Thus, the specifications of security requirements as well as their implementation by means of security mechanisms are of utmost importance. Many security threats are not technical; rather, they are social, as they originate from theinteractions between socialactors It is increasingly uncommon for software systems to be fully specified before implementation begins. This is because uncertainty about the right requirements is inescapable. A major concern with such lightweight requirements engineering is that non-functional requirements, such as security, are often neglected since system functionality is the focus [8]. The growth of the strategic importance of IT necessitates the need to ensure that software to be developed or procured is aligned with the strategic business objectives of the organization it will support. Attaining this alignment is a non-trivial problem; firstly, decisions in the Requirements Engineering (RE) phase are some of the most complex in the software development or procurement lifecycle .

This paper is organized as follows. In section II,adescription about the Literature survey has been presented .In section IIIthe Related work and an architecture of the proposed system are presented. In section IV experimental result and discussion on the result have been discussed andpresented. Session V presents the conclusion.

II. Related work

In the field of requirement engineering, a handful of researches have been done, with the technological improvement in software fields since it has gained more importance. Here, some of the recent researches are as mentioned below:-
Tony Gorschek et al. [4] have presented a framework of dependent variables that serves as a full range for requirements engineering quality assessment. The quality of the SRS itself was just the first level. Other higher, and more significant levels, include whether the project was successful and whether the resulting product was successful.
Alves et al.[5] have focused on RE within **SPLE** and has the following goals such as assess research quality, synthesize evidence to suggest important implications for practice, and identify research trends, open problems, and areas for improvement**.**
Daniel et al. [10] have investigated RE processes in successful project environments to discover characteristics and strategies that allow us to elaborate RE tailoring approach. They performed a field study on a set of projects at one company. They investigated content analysis which RE artefacts was produced in each project and to what extent they were produced along with this performed qualitative analysis of semi-structured interviews to discover project parameters that relate to the produced artefacts and finally they used cluster analysis to infer artefact patterns and probable RE execution strategies.

ISSN 2348 – 7968

Juan et al. have proposed a survey answered by RE tool vendors. The purpose of the survey was to gain an insight into how current RE tools support the RE process by means of concrete capabilities, and to what degree.
Florian et al. have provided a high-level description of each of these standards and highlights their interconnection. It thus provides to the systems engineer some guidance as to the relevance of those standards. ISO/IEC 24766 defines requirements for requirements engineering tools. ISO/IEC/IEEE 29148 describes processes for requirements engineering.
Neil et al. have examined how to support lightweight, agile requirements processes which could still be systematically modeled, analyzed and changed. They proposed a framework, **RE-KOMBINE**, which was based on a propositional language for requirements modeling called Techne.
Milene et al. have proposed an approach that starts by assessing the importance of individual types of human errors in different contexts using questionnaires to extract knowledge from RE practitioners. A method for managing the priorities of solutions to improve a given process was then proposed and a prototype application was implemented for evaluation with practitioners.
Ming-Xun Zhu et al. have proposed a fuzzy qualitative and quantitative soft goal interdependency graphs (FQQSIG) model for non-functional requirements (NFRs) correlations analysis in Trustworthy Software (TS), which was considered a critical issue by academia, government, and industry. First, the FQQSIG model constructs an NFRs criteria decomposition hierarchy graph in the complex TS situation.
Unterkalmsteiner et al.had aimed to identify and characterize evaluation strategies and measurements used to assess the impact of different SPI initiatives. Potential confounding factors interfering with the evaluation of the improvement effort were assessed. Quality was the most measured attribute (62 percent), followed by Cost (41 percent), and Schedule (18 percent).
Christian Raspotnig and Andreas Opdahl [20] have proposed different techniques for identifying risk, hazard and threat of computer-supported systems were compared. This was done by assessing the techniques' ability to identify different risks in computer-supported systems in the environment where they operate. The purpose of was therefore to investigate whether and how the techniques could mutually strengthen each other.
From the literature reviews, it is clear that among all those techniques used for the requirement engineering process has certain number of drawbacks that led us to design a technique to overcome those drawbacks. The major drawbacks being the quality of the software. Software can be made successfully when it proves to have better quality criteria. Quality improvement in requirement engineering is mainly considered in our work. Along with these quality factors the execution strategy for the RE proves to be another factor that still resides in the existing techniques. Along with these factors, the execution time and the cost are some of the other drawbacks existing in some of the techniques.

.

III. Problem definition

Organizations throughout the world are attempting to improve their ability to perform requirements engineering. All of them need to determine if they have been successful in their efforts [4].However various limitations can result which may lead to ineffective requirement engineering process. Some of the problems that exist in the requirement engineering includes,

- **Requirement for Quality Improvement,**
- **Reduced time-to-market,**
- **Costs in different domains and**
- **Choice of Incorrect Requirement parameters.**
- **Different RE execution strategies are to be developed**.

IV. Proposed methodology

In recent years, the requirement engineering field has become an interesting research area due to wide spread application in different fields. In the software field to the requirement engineering plays a major role. The decisions made in the requirements phase greatly affect the value of the resulting software. Proper requirement selection can further reduce the cost of software production which can further enhance the quality of software being designed. In this proposed work we have developed an efficient requirement engineering framework on software designing inorder to improve the quality of the particular software. The major goal is to reduce the cost required for software so that the quality of the software can be improved efficiently. To initiate our work we have selected the requirements for designing the software in such a way that we can improve the quality of designing. The proposed method can be divided into three phase. In the first phase the requirements are collected. The various factors considered in the requirement phase are time and execution rate. In the second phase the selected requirements are optimized for better designing. We have utilized an optimization algorithm for selecting the requirements. The optimization algorithm utilized is the improved particle swarm optimization algorithm (IPSO). By optimizing the requirements for designing particular software the quality of the software can be improved by reducing the cost. The final phase is the cost calculation to check the quality of thesoftware. The proposed method is implemented using the working platform of JAVA.

ISSN 2348 – 7968

Architecture of Proposed System

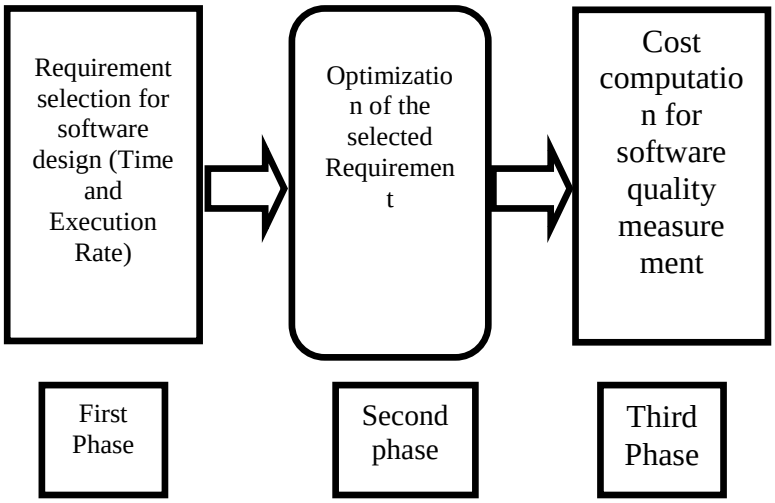

Figure 1. Architecture of Proposed System

Particle Swarm Optimization is an approach to problems whose solutions can be represented as a point in an n-dimensional solution space. A number of particles are randomly set into motion through this space. At each iteration, they observe the "fitness" of themselves and their neighbours and "emulate" successful neighbours (those whose current position represents a better solution to the problem than theirs) by moving towards them. Various schemes for grouping particles into competing, semi-independent *flocks* can be used, or all the particles can belong to a single global flock. This extremely simple approach has been surprisingly effective across a variety of problem domains.

V.The algorithm

The Particle swarm optimization algorithm(PSO) is an evolutionary computation method to that of the Genetic Algorithm in which a specific system is initialized by a population of arbitrary solutions. PSO, in addition to every latent key, randomized velocity is also allocated to fabricate a particle. Every particle goes along its coordinates in the dilemma space with reference to the superlative key. Moreover, the fitness value is also taken into account for the supplementary procedure. This fitness value is designed as *pbest.* The position of these solutions is deemed as *gbest.*

In our ambitious method, we have taken an innovative procedure by means of a tailored edition of PSO..

The PSO algorithm simulates the behaviour of birds flocking towards food. It learned from this scenario and used it to solve the optimization problems. In this PSO algorithm, each single solution is a"bird" in the search space. We call it a "particle". All the particleshave fitness values which are evaluated by the fitnessfunction to be optimized, and have velocities that direct the flying of the particles. The particles fly through the problem space by following the current optimum particles.PSO is initialized with a group of random particles (solutions) and then searches foran optima by updating generations. In every iteration, each particle is updated by followingthe two "best" values. The first one is the best solution (fitness)which it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called **gbes**t. When a particle takespart of the population as its topological neighbors, the best value is a local best and iscalled the**tBes**t.After finding the two best values, the particle updates its velocity and positions with following equations:

v[] = v[] + c1 * rand() * (pbest[] - present[]) + c2 * rand() * (gbest[] - present[]) (1)

present[]=present[]+v[] (2)

Where (v[] is the particle velocity, present[] is the current particle (solution). pbest[] and gbest[] are defined as stated before. rand () is a random number between (0,1). c1, c2 are learning factors having the value of 2.

The pseudo code for the algorithm is illustrated as follows:-

For each Particle(P(i)

InitializeParticle

End

Do

For each particle p(i)

Calculate Fitness value F(I)

If F(i)better than pBest (Best Fitness value)

Set currentvalue=new (pBest)

End

Choose the particle p(i) with the best fitness value of all the particles as the gBest.

For eachparticle p(i)

Calculate particle velocity according to equation (1)

Update the particle position according to equation (2)

ISSN 2348 – 7968

End.

Flow Diagram of PSO Algorithm

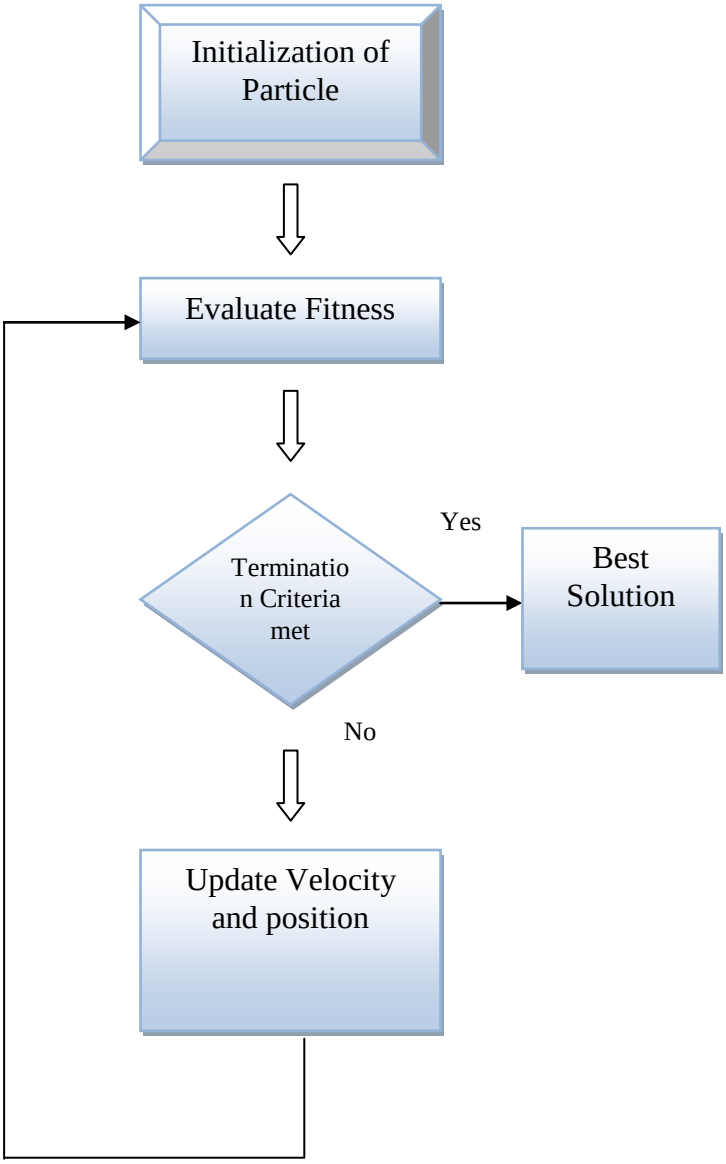

Figure 2 Flow Diagram of PSO Algorithm

From the above case, we can learn that there are two key steps when applying PSO to optimization problems: the representation of the solution and the fitness function. One of the advantages of PSO is that PSO take real numbers as particles. It is not like GA, which needs to change to binary encoding, or special genetic operators have to be used. For example, we try to find the solution for f(x) = x1^2 + x2^2+x3^2, the particle can be set as (x1, x2, x3), and fitness function is f(x). Then we can use this as the standard procedure to find the optimum. The searching is a repetitive process, and the stop criteria are that either the maximum iteration number must be reached or the minimum error condition must be satisfied.

VI. Result and Discussion

From the above case, we can learn that there are two key steps when applying PSO to optimization problems: the representation of the solution and the fitness function. One of the advantages of PSO is that PSO take real numbers as particles. It is not like Genetic Algorithm, which needs to change to binary encoding, or special genetic operators have to be used.

For example, we try to find the solution for f(x) = x1^2 + x2^2+x3^2, the particle can be set as (x1, x2, x3), and fitness function is f(x). Then we can use the standard procedure to find the optimum. The searching is a repeat process, and the stop criteria are that the maximum iteration number is reached or the minimum error condition is satisfied.

The PSO can be further modified .by allocating the worst case in addition along with the best case and furthermore cross over operation is also comprised after the fitness choice which would further raise the chance of selectingthe most excellent particle.

There are not many parameters that need to be tuned in PSO. The following is the list of the parameters and their typical values:-.

The number of particles: the typical range is 20 - 40. Actually for most of the problems 10 particles is large enough to get good results. For some difficult or special problems, one can try 100 or 200 particles as well.

Dimension of particles: It is determined by the problem to be optimized,

Range of particles: It is also determined by the problem to be optimized, you can specify different ranges for different dimensions

Vmax: it determines the maximum change one particle can take during one iteration. Usually we set the range of the particle as the Vmax for example, the particle (x1, x2, x3) X1 belongs [-10, 10], then Vmax = 20

Learning factors: c1 and c2 usually equal to 2. However,

other settings were also used in different papers. But usually c1 equals to c2 and ranges from [0, 4]

The stop condition: It is the maximum number of iterations the PSO can execute and the minimum error requirement The stop condition also depends upon the problem that needs to be Optimized.

VII. CONCLUSION

Most evolutionary techniques have the following procedure:

1. Random generation of an initial population
2. Choosing a fitness value for each particle which will directly depend on the distance to the optimum.
3. Reproduction of the population based on fitness values.
4. If requirements are met, then the process is terminated or else step 2 is evaluated

From this procedure, we can learn that PSO shares many common points with GA. Both algorithms start with a group of a randomly generated population, both have fitness values to evaluate the population. Both update the population and search for the optimum with random techniques.Both systems donot guarantee success. .

We can learn that PSO does not have genetic operators like crossover and mutation. Particles update themselves with the internal velocity. They also have memory, which is important to the functionality of the algorithm.

Compared with genetic algorithms (GAs), the information sharing mechanism in PSO is significantly different. In GAs, chromosomes share information with each other. So the whole population moves like a one group towards an optimal area. In PSO, only gBest (or lBest) gives out the information to others

It is a one -way information sharing mechanism. The evolution only looks for the best solution. Compared with GA, all the particles tend to converge to the best solution quickly even in the local version in most cases.

Hence the worst case concept in Fitness value should also be considered by implementing the cross-over operation to raise the chance of selecting the most excellent particle. By doing this approach, Non-functional requirements like security requirements can be properly selected and efficiently optimized to reduce cost of software production there by enhancing software Quality.

REFERENCES

[1] Tony Gorschek and Alan M. Davis, "Requirements engineering:In search of the dependent variables", Information and Software Technology, Vol.50, pp.67–75,2008.

[2]Vander Alves, Nan Niu, Carina Alves&George Valenca",RequirementsEngineering for software product Lines: A systematic literature review", Information and Software Technology, Vol.52, pp.806–820,2010.

[3] Daniel Mendez Fernandez, Stefan Wagner, Klaus Lochmann, Andrea and Baumann and Holger de Carne," Field study on requirements Engineering: Investigation of artefacts, project parameters, and executionexecution strategies", Information and Software Technology, Vol.54, pp.162–178, 2012.

[4]Milene Elizabeth Rigolin Ferreira Lopes and Carlos Henrique Quartucci Forster, "Application of Human theories for the process improvement of Requirements Engineering", Information Sciences,Vol.250,pp,142-161,2013.

[5]Daniel Mendez Fernandez & Jonas Eckhardt,"Understanding the Impact of Artefact-based RE –Design of a Replication Study", Empirical software Engineering Measurement, pp. 267-270,2013.

[6] Daniel Mendez Fernandez and RoelWieringa ,"Improving Requirements Engineering by Artefact Orientation",LectureNotes in Computer Science,Vol.7983,pp 108-122,2013.

[7]Florian Schneidera and Brian Berenbachb,"A literature Survey on International Standards for Systems Requirements Engineering ",Conference on systems Engineering Research Vol.16, pp.796-805,2013.

[8]Neil A.Ernst,AlexanderBorgida,IvanJuerta and John Mylopoulos,"Agile requirements engineering via paraconsistentreasoning"Information Systems,2013.

.