

# Risk Prioritization for Software Development using Grey Wolf Optimization

B.Prakash, V.Viswanathan

**Abstract** - Risks are uncertainties that influence the project performance to greater extent. To ensure software quality and project success every organizations should enforce a proper mechanism to efficiently manage the risks irrespective of the process model they follow. Risk prioritization is a most critical step in risk management process that helps the organization to resolve the risks in shorter duration of time. This study focuses on prioritizing the risks involved in software development using Grey Wolf Optimization (GWO) algorithm. Further, the proposed Software Risk Prioritization (SRP-GWO) approach is compared with other prioritization techniques such as Analytical Hierarchy Process (AHP), Particle Swarm Optimization (PSO), Delphi, Average Ranking and Categorizing scale and the results are evaluated based on five criterion attributes such as Simplicity, Adaptability, Running time, Accuracy and Consistency. The findings show that the proposed approach outperforms other existing techniques.

**Index Terms** - Risk Management, Software Development, Grey Wolf Optimization, Project Management

## I. INTRODUCTION

Software Organizations in general uses different types of process models for developing their software applications based on project requirements and client preferences. Earlier days companies were using traditional model like waterfall model for their development activity but in recent times many organizations prefer to adopt agile process for their software development over traditional approaches [1]. Agile methodology is based on iterative and people-based approach. Agile process relies mainly on customer satisfaction, welcoming changes in requirements even in the later stage of development [2], frequent delivery of working software, continuous collaboration among developers and customers, light-weight approach [3].

The software Risk Management (RM) is a part of project management in software engineering discipline which defines a set of process, tools and methods to effectively manage the risks for any given project. Boehm proposed a set of practices and principles to efficiently handle the risks in an organization [4]. In general every software organizations have a mechanism to identify, analyse and manage the risks [5]. It is highly unrealistic to remove all the risks in a project

but the main objective of risk management is to minimize the impact of risks to improve the performance of the project [6].

The risk management process is not explicitly followed in agile based projects. The main reason is: (1) Since agile follows iterative approach the risks are automatically minimized by itself and (2) The risks if arises are dynamically handled as part of daily scrum meetings. Due to the above reasons, agile teams do not follow any kind of risk management procedure explicitly [6]. However, the study of software risk management [4][7] indicates that globally, only 16.2% of projects were completed on time and with-in budget. Among the rest, 52.7% of projects are challenged (delivered late or over the budget) and 31.1% of projects were cancelled before completion. The major reason for this large quantity of failures is due to lack of proper risk management [8]. Therefore organizations started adopting risk management procedures to ensure software are developed on-time, with-in budget and the delivered product is of high quality [4][7][9]. Irrespective of process model adopted, risks are always part of the software development process that needs to be addressed in early duration of the project.

Risk Management process has set of phases such as risk identification, risk analysis, risk prioritization, risk management planning and risk monitoring. Among these, risk prioritization is an important step that is needed for identifying and resolving the most critical risks within the stipulated timeframe. The effective risk prioritization technique is useful for organizations to support timely decision making and to take appropriate escalation measures [26]. Researchers in recent years proposed various techniques to prioritize the risks. Each technique has its own advantages and limitations and no single technique is suitable for all types of projects. Risk matrix is one of the widely used techniques for prioritizing the risks [10][11]. Rabihah [12] proposed a risk prioritization method using Analytic Hierarchy Process (AHP). Agrawal *et al.*, [13] introduced Agile based Risk Rank (AR-Rank) method for prioritization of risks and Particle Swarm Optimization (PSO) technique is used for optimization of identified risks in agile software development. This paper proposes a method to prioritize the set of risks in software development using Grey Wolf Optimization (GWO) algorithm. Mirjalili *et al.*, [14] proposed the algorithm (GWO) that mimics the hierarchical structure and hunting behaviour of grey wolves in nature. The paper is organized as follows. Section II summarizes the related work, Section III outlines about grey wolf optimization (GWO) algorithm, Section IV discusses about the proposed methodology. Section V elaborates on the results and discussions. Finally, Section VI presents conclusion with future directions.

Revised Manuscript Received on 30 March 2019.

\* Correspondence Author

**B. Prakash**, Assistant Professor, Department of Computing Science and Engineering, Vellore Institute of Technology, Chennai, India.

**V. Viswanathan**, Associate Professor, Department of Computing Science and Engineering, Vellore Institute of Technology, Chennai, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

## II. RELATED WORK

Organizations started adopting many risk management strategies as part of their software development process, but still there is noticeable lack of proactive risk management in agile based environments [25][32].

Boehms [15] in 1989 proposed a risk based life cycle model named as spiral which incorporates risk analysis in each phase of life cycle. Even though this is most suitable for every project, this model requires active involvement of customers throughout the project which is difficult to undergo in real time.

Agrawal *et al.*, [13] proposed agile based risk rank (AR – Rank) method by integrating pair-wise comparison and Analytical Hierarchy Process (AHP) for prioritizing the risk items in agile based projects. Further this approach utilizes Particle Swarm Optimization (PSO) for optimizing the identified risks. However, this approach might not be suitable for large size of data since it involves large number of comparisons.

Albadarneh *et al.*, [6] discusses risk management strategies incorporated for agile based environments. The author has done a comparative study about risk management based on three agile methods namely XP, Scrum and DSDM. This approach focused only on comparative study among three agile methods and prioritization of risks is not covered as part of this approach.

Li *et al.*, [16] suggested a hybrid model called software process risk optimization model that integrates risk management into software process model. This model resembles a heavyweight approach which might not be suitable for agile based environments.

Tonella *et al.*, [20] proposed another meta-heuristic evolutionary approach known as genetic algorithm for prioritization of software requirements. In recent days, genetic algorithms are largely used in solving real life computational problems in the optimization and prioritization fields such as association rules prioritization [21], regression test case prioritization [22], test-case prioritization scenarios derived from UML diagrams [23], fault-based regression test case prioritization [24] and so on.

Elmsalmi *et al.*, [27] applied a modelling technique known as Supply chain risk management (SCRM) using MICMAC method for effective risk prioritization and assessment for global supply chain network. The risks involved in supply chain network are identified first and the model developed a reachability matrix that represents a pairwise dependency between risks. Based on the developed reachability matrix, direct and indirect risk attributes are identified. However, this approach deals only specific to supply chain management perspective and does not provide a generic solution.

RM Sum [28] demonstrates prioritization of risks using Analytic Hierarchy Process (AHP). This method structures the risks into organized framework that helps project managers to easily able to understand and perform risks assessment effectively. This approach demands more number of comparisons among risks at each levels which in turn not suitable for large size of dataset.

## III. GREY WOLF OPTIMIZATION (GWO)

Mirjalili *et al.*, [14] recently introduced a meta-heuristic Optimization algorithm called Grey Wolf Optimizer (GWO) that resembles the hierarchical organization and hunting

behaviour of grey wolves in natural life. The proposed algorithm has been widely accepted and applied to solve many real life computational problems like max-flow problem [17], 0/1 Knapsack problem [18], prioritization of software requirements [19].

The GWO algorithm replicates the hunting behaviour of grey wolves and its leadership qualities in nature. Grey wolves usually live as a group of sizes varying between 5 and 15. The grey wolves are divided into four categories namely alphas ( $\alpha$ ), betas ( $\beta$ ), deltas ( $\delta$ ) and omegas ( $\omega$ ). Alpha wolves are higher level in hierarchy who is responsible for decision making. Beta Wolves are next level to alphas who provides timely advice to his superiors (alphas). Deltas are at third level responsible for providing relevant information to higher levels while omega is at last level that performs the day to day assigned tasks. The hierarchical organization of grey wolf is represented in Fig 1.

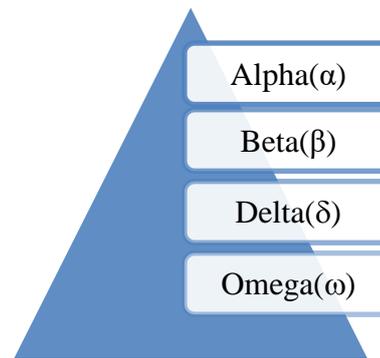


Fig 1. Hierarchical Structure of Grey Wolves

The hunting behaviour of grey wolves is mathematically represented as optimization that are guided by sequence of decisions made by controlling wolves ( $\alpha, \beta$  and  $\delta$ ). Further,  $\alpha$  represents the first best solution,  $\beta$  denotes second best solution and  $\delta$  describes third best solution whereas  $\omega$  depicts rest of the candidate solutions. The hunting process involves set of phases such as searching (chasing) for a prey, locating, surrounding or encircling the prey and finally attacking (hunting) the prey.

### A. Searching (Chasing) Prey

Usually, decision for searching a victim is made by the superior wolves such as  $\alpha, \beta$  and  $\delta$ . The position of the prey would be dynamically located by the wolves during iteration and the nearby wolves are prepared to surround the position based on the prey position.

### B. Encircling Prey

As part of the hunting process, a group of wolves encircle the prey prior to the attack. The mathematical representation of encircling prey is modelled using equations (1) and (2).

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (1)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (2)$$

where,

$D$  represents distance between location of prey ( $X_p$ ) and wolf location ( $X$ ),  $t$  denotes current iteration,  $A$  and  $C$  are coefficient vectors that are denoted by equations (3) and (4).

$$\vec{A} = 2 \vec{a} \cdot \vec{r}_1 - \vec{a} \tag{3}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{4}$$

where,  $r_1$  and  $r_2$  are vector values in the range [0,1] and a decreases from 2 to 0 as iteration proceeds.

**C. Hunting Prey**

In real world, hunting process is initiated, planned and performed under the leadership of alpha with the help of beta and delta. Also in ideal scenario, all three packs ( $\alpha$ ,  $\beta$  and  $\delta$ ) has a prior knowledge about the location of the prey. But mathematically it is not possible to implement the same idea due to the fact that we are not aware about the optimal solution (prey). So for modelling the above situation mathematically, we assume  $\alpha, \beta$  and  $\delta$  has better knowledge about the optimal search agent (location of the prey). Thus, the first three best solutions obtained so far are stored and updates other search agents locations (omega) based on the obtained locations using equations 5, 6 and 7.

$$\vec{D}_\alpha = |\vec{c}_1 \cdot \vec{X}_\alpha - \vec{X}|, \vec{D}_\beta = |\vec{c}_2 \cdot \vec{X}_\beta - \vec{X}|, \vec{D}_\delta = |\vec{c}_3 \cdot \vec{X}_\delta - \vec{X}| \tag{5}$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha), \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta), \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta) \tag{6}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{7}$$

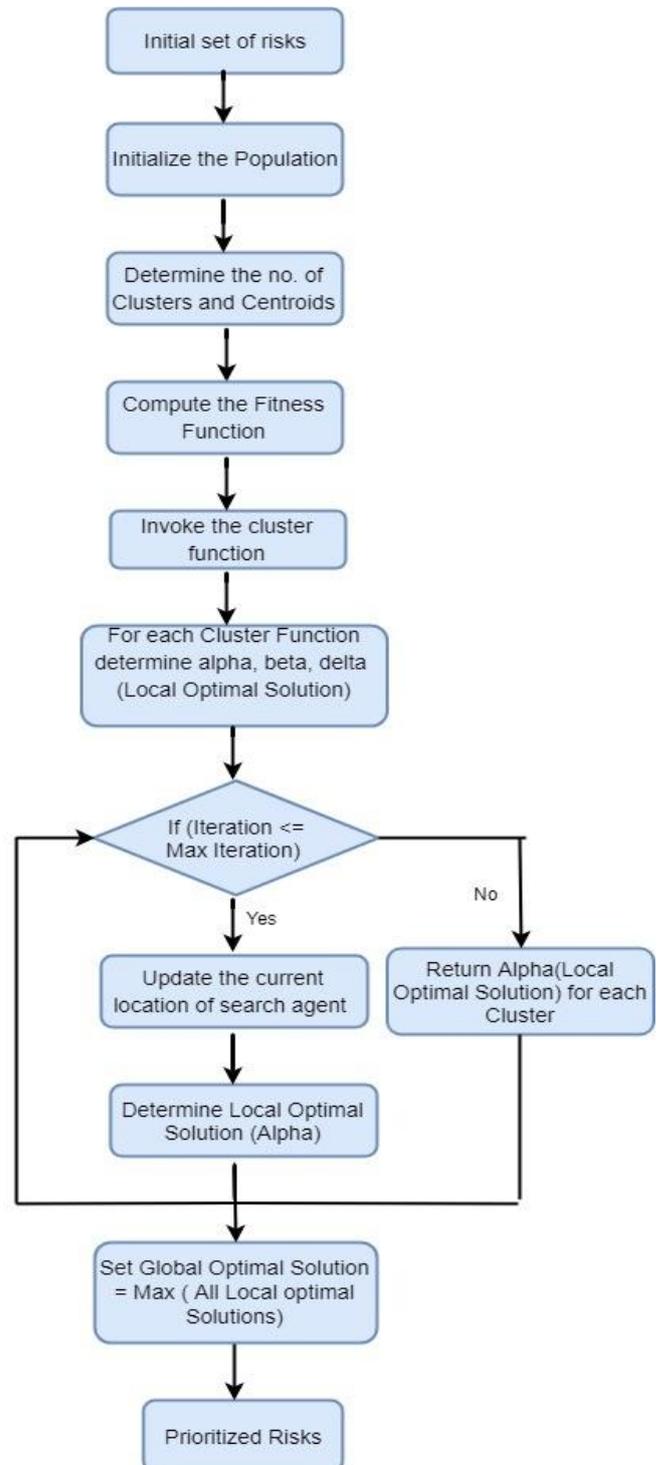
**D. Attacking the Prey (Exploitation)**

Hunting process (Optimization) gets complete when grey wolves attack the victim when it stops moving. For mathematically model the attacking process, the value of ‘a’ gets decreased from 2 to 0 when iteration progress. Accordingly random value A in [-a, a] also getting reduced. When the value of ‘A’ becomes less than 1 means exploitation is achieved and the suggested candidate solution proceeds towards local optimal solution, which in turn leads to inactive state (stagnation).

In order to avoid local stagnation, the value of ‘A’ is assumed to be either greater than 1 or lesser than -1, which makes the wolves far away from the prey and achieves the exploration phase (search of victim). Another factor to be considered as part of exploration process is ‘C’ of range [0,2] and it denotes the random weight of victim with respect to distance. When  $C > 1$ , it emphasize its effect and when  $C < 1$  it reduces its influence.

**IV. THE PROPOSED “SRP-GWO” ALGORITHM**

The proposed algorithm has been developed to prioritize the risks involved in software development process. As suggested, the algorithm has been stratified in order to obtain the optimal risk prioritization results. The algorithm is implemented in ten different projects from two organizations and the results are compared with other existing algorithms. The process flow of grey wolf optimization for Software Risk Prioritization (SRP-GWO) is represented in Fig 2.



**Fig 2. Process Flow of Proposed “SRP-GWO” Algorithm**

**A. Initialization Stage**

The initial set of grey wolf population and the associated random vectors are initialized. From the initialized search agents, randomly choose one as victim and all remaining is considered as source.

**B. Fitness Function**

The main objective of the fitness function is to assign the wolves in the appropriate cluster based on the distance.

In our study, fitness function is defined as determining the distance between the wolves and all the centroids and assigns the wolves to the appropriate cluster based on the minimal distance. Fitness function is represented in Fig 3.

```

For each wolf l
Measure the distance between each wolf (i)
and all centroids Check if wolf (i) not
assigned
Assign wolf(i) to its nearest centroid
End For (if i = max. number of Wolves)
    
```

Fig 3 Fitness Function

### C. Clustering

Mirjalili *et al.*, [14] suggested that the wolves live as a group of size between 5 and 15. Thus for grouping, K-means clustering algorithm has been used such that each cluster has minimal of 5 and maximum of 15 wolves. The number of clusters, centroids and mapping of wolves into the clusters are performed using equations (8 – 10). The cluster function is denoted in Fig 4.

$$\text{No. of Clusters} = \text{Ceiling} \left( \frac{\text{Number of Wolves}}{15} \right) \quad (8)$$

$$\text{No. of Centroids} = \text{Number of Clusters} \quad (9)$$

$$\text{Distance} = ||X_i - \text{Centroid}_j|| \quad (10)$$

```

Calculate number of clusters using eq. 8
Calculate number of centroids using eq. 9
for each search agent
    Invoke Fitness Function
End for
    
```

Fig 4 Cluster Function

### D. Optimal Solution Function

Once all clusters are defined, the three best solutions ( $\alpha$ ,  $\beta$  and  $\delta$ ) are identified for each cluster. Here,  $\alpha$  denotes the best optimal solution (nearest search agent to the victim),  $\beta$  represents second best solution (second nearest search agent to the victim) and  $\delta$  shows third best solution (third nearest search agent to the victim). Each local best solution is compared with all other cluster solutions and the maximum of all the local optimal solutions are considered as best global solution as represented in Fig 5.

```

While each cluster is non-empty (1,2,3....n)
For each cluster K
    Set Global_Optimal_Sol = Max (All local
    Optimal Sol)
End for
    
```

Fig 5. Optimal Solution

## V. EXPERIMENTAL RESULTS AND DISCUSSION

This section explains the process involved in evaluating the effectiveness of the proposed algorithm with other existing approaches. The objective of this experimental study is to implement the proposed technique in a real life pilot project and analyse the performance of the proposed method with five other existing risk prioritization techniques such as Analytic Hierarchy Process (AHP), Particle Swarm

Optimization (PSO), Delphi method, Average ranking and Categorizing scale. The evaluation is based on comparing the six risk prioritization techniques to determine which one performs well upon various criterion attributes like simplicity (easy to use and understand), performance (takes less amount of time), adaptability (more number of risk factors getting added), stability (performs consistent on different situations) and accuracy. The experiment is performed using interview mode by getting a feedback from software practitioners' about their experience and understanding about each technique on these five criterion attributes. This experimental study is adopted based on the experimental practices suggested in [29][31].

### A. Experimental Design

The proposed "SRP-GWO" algorithm is implemented on a real life pilot project in an organization in order to evaluate the performance of proposed algorithm with other existing risk prioritization techniques. The project has been selected carefully with the following characteristics in mind: widely used web based e-commerce application, medium in size with a six member team. The evaluation is based on the perspective of eight member team comprises of six developers, one project manager and one customer. For this study, there are total of thirty five software risks factors were identified for the selected pilot project based on expert judgment method. The project manager, customer and the team practitioners were actively involved as part of risk identification process. The same set of identified thirty five risks are assigned to each member and asked them to prioritize the risks independently.

In order to achieve the accurate and consistent results, the test was conducted with specific intervals over a time period by focusing on evaluating only one technique in a day. Further, to get the knowledge and clarity among the practitioners' on the techniques to be tested, thirty minutes brainstorming session was scheduled each day in which the participants discusses about the technique which they are going to evaluate on the particular day. Following the brainstorming session, two hours are allotted for each participant to complete the evaluation process.

### B. Threats to Validity

By performing the experimental analysis based on one project, immediate question one can think of: How valid the results are? Thus, we have carefully selected the project from an organization by keeping this question in mind and ensured that the selected project satisfies the set of characteristics as highlighted earlier. The objective of this study is to evaluate the performance of the proposed SRP-GWO algorithm by comparison with other existing risk prioritization algorithm on the selected pilot project. This experimental evaluation process also involves various threats that are identified as follows:

**Risk Identification:** Expert judgement technique has been adopted to identify the risk factors for the selected project. The team comprises of scrum master, product owner and group of developers are involved in the risk identification process. The number of risk factors selected for the validation is limited to thirty five by keeping in mind the time duration to perform the experiment.

**Nature of Risk Factors:** The risk factors are selected based on the influence it had on the project outcomes. There are different categories of risks such as technical risks, project management risks, customer related risks, organizational management related risks and so on. Eventhough the risks belong to different categories, the impact it makes on the project might be same. Thus, the team involved in the risk selection process are considered the above factors are the risks are identified from all possible categories.

**Team involved in validation:** In general, adding more people in the software development process leads to more confusion and delay in the project deliverables [30]. Hence, the number of people involved in this evaluation process is limited to eight who are part of the development team.

**Risk Characteristics:** The identified risk factors are carefully chosen based on the influence it had on the project outcomes with respect to cost, duration, scope or quality. However, dependent risks and external risks are not considered as part of this experimental process.

**C. Experimental Analysis**

A questionnaire has been prepared as part of the interviewing process in which a set of questions are asked to each participants on the selected technique. The participants are expected to provide answers in the range of values and the average value has been considered for our evaluation. The following questions are as part of the questionnaire used in the interview process.

- 1) The experimental evaluation starts with the first question to development team on how simple the risk prioritization techniques to understand and to adapt in the project. The results are highlighted in Fig 6. The feedback from the participants suggests that SRP-GWO were simple to understand and adapt in their project followed by categorizing scale technique. Also team feedback suggests PSO followed by AHP are most difficult techniques to follow.

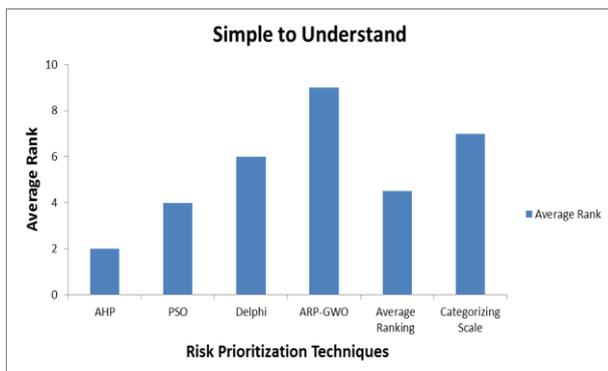


Fig 6 Comparisons among Risk Prioritization Techniques for attribute "Simple to Adapt / Understand"

- 2) The next question was how the prioritization techniques reacts when the number of risks or the work environment changes. The lesser value signifies least adaptability and higher value denotes the particular technique is highly adaptable with respect to the changes. The result as shown in Fig 7 proves that most of the participants felt Delphi is highly adaptable technique followed by Average Ranking. Our approach (SRP-GWO) ranks

third among all and PSO is the least flexible approach related to changes.

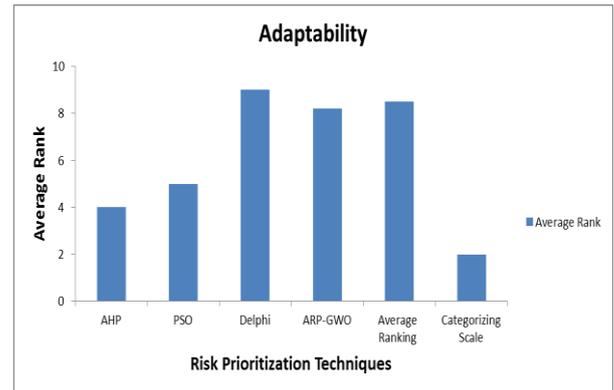


Fig 7. Comparisons among Risk Prioritization Techniques for attribute "Adaptability"

- 3) The third question to team was how much time they spent to perform the prioritization. The higher value denotes the particular technique consumes least time to complete the prioritization process and the team was happy about the performance and lower value shows the team was not happy due to the more time consumption. The results are highlighted in Fig 8. The result indicates SRP-GWO is the fastest technique among all followed by Average Ranking and categorizing scale. Most participants suggest Delphi method is the slowest among all other techniques which took more time to complete the task since it involves more amount of manual effort.

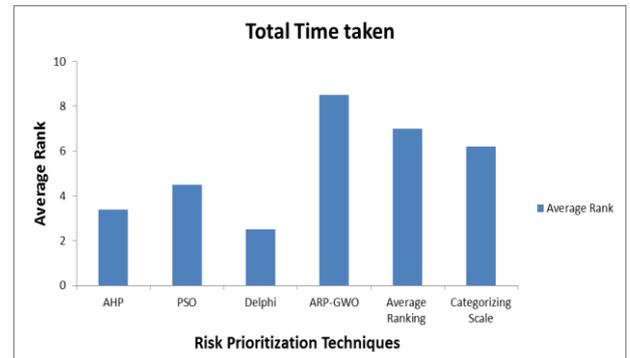


Fig 8. Comparisons among Risk Prioritization Techniques for attribute "Total Time Taken"

- 4) The next question was that the team to rank the technique based on the consistency level of the results over a period of time. The Fig 9 suggests most of the participant prefers SRP-GWO was the most stable technique followed by PSO and Delphi. Also most of team members felt AHP was least consistent technique with respect to risk prioritization among all other methods.

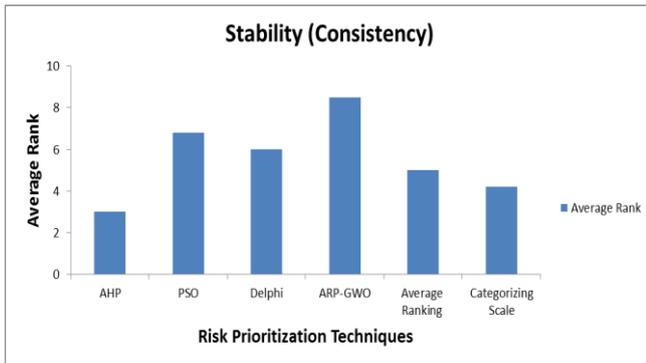


Fig 9. Comparisons among Risk Prioritization Techniques for attribute “Stability (Consistency)”

- Finally the teams were asked to provide feedback about the accuracy level on the derived results of the techniques. The result in Fig 10 clearly indicates SRP-GWO was the most accurate risk prioritization technique followed by PSO and AHP. Most participants indicate Delphi was the least accurate technique among other existing methods.

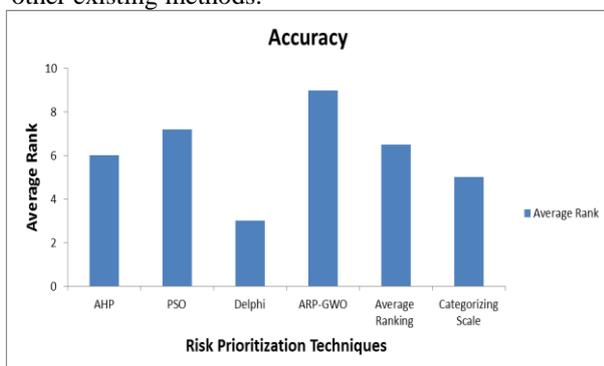


Fig 10. Comparisons among Risk Prioritization Techniques for attribute “Accuracy”

Based on the team inputs, evaluation of SRP-GWO has been performed against other existing prioritization techniques and the results are derived for individual contributing attributes. Further, it is necessary to determine the overall best risk prioritization technique for by considering the weights for five criterion attributes. The weights of the individual attributes as highlighted in Table I are necessary for stakeholders to define the importance of criterion attributes.

TABLE I  
ASSOCIATED WEIGHTS OF EACH CRITERION ATTRIBUTE

<b>Simplicity</b>	4
<b>Adaptability</b>	6
<b>Running Time</b>	5
<b>Consistency</b>	8
<b>Accuracy</b>	7

In this study, for the selected pilot project the weights are derived based on expert judgment by conducting a brainstorming session among development team members as part of the sprint planning. The eq. (11) and (12) are used to determine the overall best risk prioritization technique.

$$R_{xy} = W(A_x) * (N + S_x(M_y)) \quad (11)$$

$$OR(M_y) = \frac{\sum_{x=1}^{NA} S_{x,y}}{NA} \quad (12)$$

where,

$R_{xy}$  denotes Ranking of technique  $y$  in criterion attribute  $x$

$W(A_x)$  represent weights of attribute  $x$

$N$  is number of techniques compared

$S_x(M_y)$  suggest score of technique  $y$  in criterion attribute  $x$

$OR(M_y)$  is overall ranking of technique  $y$

$NA$  represent number of criterion attribute

Fig 11 provides the comparison of proposed SRP-GWO algorithm among all other existing methods based on weighted criterion attribute. The result proves that SRP-GWO outperforms all other techniques on the basis of evaluation criteria.

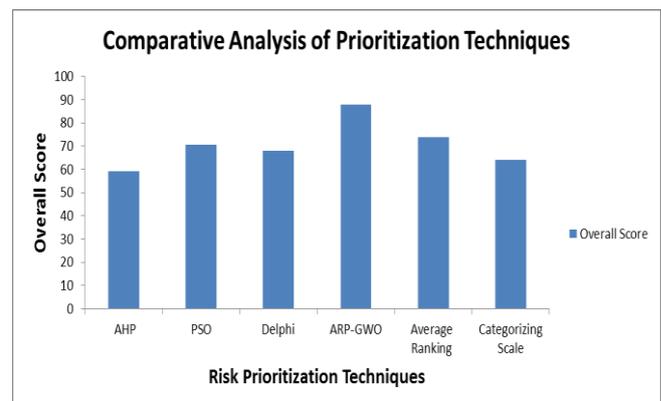


Fig 11. Comparisons of Risk Prioritization Techniques

## VI. CONCLUSION AND FUTURE ENHANCEMENTS

This study proposed a risk prioritization algorithm using grey wolf optimization technique. The suggested approach ranks the critical risk factors and helps to optimize the risks which in turn enhance the successful delivery of the project. This approach has been implemented in a real life pilot project and the performance was evaluated against other existing techniques such as AHP, PSO, Delphi, Average Ranking and Categorizing scale based on five criterion attribute measures namely Simplicity, Adaptability, Running time, Consistency and Accuracy. Based on the analysis, the result proves that the suggested SRP-GWO algorithm outperforms all other existing optimization algorithms with respect to criterion attribute measures. The limitations of this study are: (i) Although the evaluation of this study is based on pilot project and with the closed set of eight member team, the results are closely confined to this environment. (ii) The derived overall score among the prioritization techniques is relative to the weights assigned to the criterion attributes. Thus, for any changes in the weight in turn modify the results obtained. Future work might be focussed on validation using different industrial projects, incorporating metrics based evaluation, integrating mutation and other existing evolutionary attributes with grey wolf optimization algorithm for achieving the better results.

## REFERENCES

1. K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, "Manifesto for agile software development," 2001. [Online]. Available: <http://www.agilemanifesto.org/>.
2. VersionOne, "8th annual state of agile survey," 2013. [Online]. Available: <http://www.versionone.com/pdf/2013-state-ofagile-survey.pdf>.
3. A. Cockburn, Agile software development: the cooperative game, 2nd ed. South Asia: Pearson Education, 2007.
4. Barry W. Boehm, Software Risk Management: Principles and Practices, IEEE Software, vol. 8(1), pp. 32–41, (1991).
5. DOORS. (2014) Web site: <http://www.sei.cmu.edu/risk>. [Online]. Available: <http://www.sei.cmu.edu/risk>.
6. A Albadameh, I Albadameh, A Qusef, "Risk Management in Agile Software Development: a Comparative Study", IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), 2015.
7. Michael Dedolph, The Neglected Management Activity: Software Risk Management, Bell Labs Technical Journal, vol. 8(3), pp. 91–95, (2003).
8. C Kumar, D K Yadav, "A Probabilistic Software Risk Assessment and Estimation Model for Software Projects", Eleventh International Multi-Conference on Information Processing (IMCIP), pp. 353 - 361, 2015.
9. Bernd Freimut, Susanne Hartkopf, Peter Kaiser, Jyrki Kontio and Werner Kobitzsch, An Industrial Case Study of Implementing Software Risk Management, In Proceedings of the 8th European Software Engineering Conference Held Jointly with 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering, ESEC/FSE-9, pp. 277–287, (2001).
10. APM (2004) Project Risk Analysis and Management Guide, 2nd edition, APM Publishing, High Wycombe, ISBN 1-903494-12-5.
11. PMI (2004) A Guide to the Project Management Body of Knowledge (PMBOK), 3rd edition, Project Management Institute, Pennsylvania.
12. R Md. Sum, "Risk Prioritisation Using The Analytic Hierarchy Process", Innovation and Analytics Conference and Exhibition (IACE 2015), AIP Conference Proceedings, 2015.
13. R Agrawal, Deepali Singh, Ashish Sharma, "Prioritizing and Optimizing Risk Factors in Agile Software Development", Ninth International Conference on Contemporary Computing (IC3), 2016.
14. S Mirjalili, S M Mirjalili, A Lewis, "Grey Wolf Optimizer", Advances in Engineering Software, vol. 69, pp. 46 – 61, 2014.
15. B. W. Boehm, Ed., Software Risk Management. Piscataway, NJ, USA: IEEE Press, 1989.
16. J Li, M Li, D Wu, H Song, "An integrated risk measurement and optimization model for trustworthy software process management", Information Sciences, vol. 191, pp. 47 – 60, 2012.
17. R Masadeh, A Sharieh, A Sliet, "Grey wolf optimization applied to the maximum flow problem", International Journal of Advanced and Applied Sciences, vol. 4, pp. 95 – 100, 2017.
18. E Yassien, R Masadeh, A Alzaqebah, A Shaheen, "Grey wolf optimization applied to the 0/1 Knapsack problem", International Journal of Computer Applications, vol. 169, no. 5, pp. 11 – 15, 2017.
19. R Masadeh, A Alzaqebah, A Hudaib, "Grey wolf Algorithm for Requirements Prioritization", Modern Applied Science, vol. 12, no. 2, pp. 54 – 61, 2018.
20. P Tonella, A Susi, F Palma, "Interactive requirements prioritization using a genetic algorithm", Information and Software Technology, vol. 55, pp. 173 – 187, 2013.
21. M Ramesh Kumar, K Iyakutti, "Application of Genetic algorithms for the prioritization of Association Rules", IJCA Special issue on "Artificial Intelligence Techniques – Novel Approaches & Practical Applications", AIT, pp. 35 – 38, 2011.
22. E Ashraf, A Rauf, K Mahmood, "Value based Regression Test Case Prioritization", Proceedings of the World Congress on Engineering and Computer Science (WCECS), 2012.
23. S Sabharwal, R Sibal, C Sharma, "Applying Genetic Algorithm for Prioritization of Test Case Scenarios Derived from UML Diagrams", International Journal of Computer Science Issues (IJCSI), vol. 8, issue. 3, no. 2, pp. 433 - 444, 2011.
24. A Kaur, S Goyal, "A Genetic Algorithm for Fault based Regression Test Case Prioritization", International Journal of Computer Applications, vol. 32, no.8, pp. 30 – 37, 2011.
25. B Derfer, "Introducing the Agile Risk Management (ARM) Framework", Agile Six Applications, Inc.
26. M Elmsalmi, W Hachicha, "Risk Prioritization in Global Supply Networks Using MICMAC Method: A Real Case Study", International Conference on Advanced Logistics and Transport, Tunisia, July 2013.
27. RM Sum, "Risk Prioritisation using the analytic hierarchy process", AIP Conference Proceedings, vol. 1691, no. 1, December 2015.
28. V R Basili, R W Selby, D H Hutchens, "Experimentation in Software Engineering", IEEE Transactions in Software Engineering, vol 12, no. 7, pp. 733 – 743, 1986.
29. M Cohn, Succeeding with Agile: Software Development Using Scrum. Pearson Education in South Asia, 2012.
30. M Khari, N Kumar, "Comparison of Six Prioritization Techniques for Software Requirements", Journal of Global Research in Computer Science, vol. 4, no. 1, 2013.
31. B G Tavares, C E S da Silva, A D de Souza, "Risk Management in Scrum Project: A Bibliometric Study", Journal of Communications Software and System s, vol. 13, No. 1, 2017.