ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 2, Issue 2, February 2013

Task level Analysis of Risks within Software Engineering Projects using Graph Theory

Quist-Aphetsi Kester, MIEEE

Abstract— **The ability of any industry to survive is dependent on its ability to execute its projects successfully. However, the success of any project depends on several influencing areas, which require reliable and regular attention by the project manager. Some of these areas are schedule management, finance management, change management, conflict management, etc. It is important to understand that each of the above mentioned areas come into view as risk if not managed in a righteous way. Software projects needs a lot of time and attention during its planning phase. Risk analysis is a crucial component and activity of every project management process. Different tools and techniques in evaluation of risks within software projects at each phase do exist but they do not connect and analyze risk at each task level of a project.**

This paper seeks to propose an effective way of analyzing risks within a software project at the task level within software engineering projects. Each risks associated with each task can easily be analyze to determine its effect on neighbor tasks as well as track its chain of effects within a given project. The representation of the tasks with their associated risks will be done using pert-chart and the analysis of the risks will be performed using graph theory. At the end risk patterns can easily be traced and identified as well as the chain of effects of risks can easily be tracked within a given project.

Index Terms—**Risks, software engineering, graph theory, analysis.**

I. Introduction

With ever-increasing complexity and increasing demand for bigger, better, and faster, the software industry is a high risk business. When teams don't manage risk, they leave projects vulnerable to factors that can cause major rework, major cost or schedule over-runs, or complete project failure. Adopting a Software Risk Management Program is a vital step every software manager can take to more effectively manage software development initiatives. Risk management is an ongoing process that is implemented as part of the initial project planning activities and utilized throughout all of the phases of the software development lifecycle. Risk management requires a fear-free environment where risks can be identified and discussed openly. Based on a positive, proactive approach, risk management can greatly reduce or even eliminate the need for crisis management within our software projects. [1]

Risk assessment and analysis are very important part of planning a software project because it allows the project manager to predict potential problems that will threaten the project and take steps to mitigate those problems. Adding a risk plan to a software project plan is an effective way to keep the project from being derailed by surprises or emergencies. [2] Task level risk assessment and analysis is the effective and efficient way of evaluating risks when building a software system. This makes it easy to account for technical problems as well as monitor risk trends within the software project.

Risk is a probability for occurrence of an unlikely event, which would result with highly unacceptable consequences. Software acquisition and development are two of the most risk prone challenges of this decade. Risk factors are always present that can negatively impact the development, acquisition, or maintenance processes. If neglected, these factors can tumble an unwitting program manager into acquisition disaster. To succeed in software acquisitions and development, one need to actively assess, control, and reduce software risk on a routine basis [3].

In order to avoid risks within a project, an organized, systematic decision-making process that efficiently identifies risks, assesses or analyzes risks, tracks and communicates risk and effectively reduces or eliminates risks is vital and needed within every project. Thus, risk management is one of the critical issues that need to be addressed in a skillful and efficient manner in every project. It is therefore necessary to take precautionary procedures to reduce the probability of risk occurrences and to minimize its impact in realizing effective project management.

IT systems are at the heart of modern business and the development of new software applications and maintenance of existing systems are critical to productivity and profitability. They form the central core of every business process and activity. Advances in software technology over the last 20 years have allowed progressively more complex business solutions to be created enabling companies to offer their customers exciting new services and products. And yet, software development projects still suffer from similar problems and characteristics, regardless of the technologies being used, that they suffered from more than ten years ago. [4] This occurs most often as a result of inefficient project planning due to unforeseen risks during the project life cycle. Hence there is a need for employment of effective tools and techniques in order to efficiently analyze and track risks within the life cycle of the project.

This paper proposes an effective way of analyzing risks within a given project using graph theory. The risks analysis is done at the tasks level and can be done for the entire projects. Classification of risks can also be easily achieved as well as the chain of effects of risks can easily be traced and monitored within the project (that is at the tasks level, phase level as well as the entire project). Relationships between

***Quist-Aphetsi Kester, MIEEE**, Faculty of Informatics, Ghana Technology University College Accra, Ghana, +233 209822141*

tasks in terms of risks can also be known and traced.

The organization of this paper is as follows, section II of this paper provides details of the related works in the domain of the assessment, analysis, and management of risks within software engineering projects. Section III proposes how the analysis of risks can be done at the tasks level of a given project. Section IV provides application and results: representation of risks at tasks level as well as analysis of them was carried. The last section of this paper, section V, Concludes the paper.

II. Related works

The main objective of Risk Management is to identify potential problems before they occur so that risk handling activities can be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives. It should begin at the earliest stages of project planning and continue throughout the total life cycle of the project. [1]

Ever since the advancement of software engineering has evolved, risk management has become one of the key challenges in everyday software development processes. Management of risk in software processes is needed to minimize or eradicate risk before it can harm the productivity of a software project. [5]

Formal risk analysis and management in software engineering is still an emerging part of project management. Risk management for software projects is intended to minimize the chances of unexpected events, or more specifically to keep all possible outcomes under tight management control. [6]

Risk management is a systematic approach for minimizing exposure to potential losses [7]. It provides a disciplined environment for

- continuously assessing what could go wrong (i.e., assessing risks)
- determining which risks to address (i.e., setting mitigation priorities)
- implementing actions to address high-priority risks and bring those risks within tolerance

There are three core risk management activities core that are illustrated in Figure 1:

- assess risk—transform the concerns people have into distinct, tangible risks that are explicitly documented and analyzed
- plan for risk mitigation—determine an approach for addressing or mitigating each risk; produce a plan for implementing the approach
- mitigate risk—deal with each risk by implementing its defined mitigation plan and tracking the plan to completion

A Risk Matrix is a matrix that is used during Risk Assessment to define the various levels of risk as the product of the harm probability categories and harm severity categories. This is a simple mechanism to increase visibility of risks and assist management decision making.[8]

Many standard risk matrices exist in different contexts (US DoD, NASA, ISO), [9][10][11] individual projects and organizations may create their own or tailor an existing risk matrix to suite their need.

The harm severity can be categorized as:

- Catastrophic - Multiple Deaths
- Critical - One Death or Multiple Severe Injuries
- Marginal - One Severe Injury or Multiple Minor Injuries
- Negligible - One Minor Injury

The probability of harm occurring might be categorized as 'Certain', 'Likely', 'Possible', 'Unlikely' and 'Rare'. However it must be considered that very low probabilities may not be very reliable.

Tony Cox argues that risk matrices experience several problematic mathematical features making it harder to assess risks [12]. These are:

Poor Resolution: Typical risk matrices can correctly and unambiguously compare only a small fraction (e.g., less than 10%) of randomly selected pairs of hazards. They can assign identical ratings to quantitatively very different risks ("range compression").

Errors: Risk matrices can mistakenly assign higher qualitative ratings to quantitatively smaller risks. For risks with negatively correlated frequencies and severities, they can be "worse than useless," leading to worse-than-random decisions.

Suboptimal Resource Allocation: Effective allocation of resources to risk-reducing countermeasures cannot be based on the categories provided by risk matrices.

Ambiguous Inputs and Outputs: Categorizations of severity cannot be made objectively for uncertain consequences. Inputs to risk matrices (e.g., frequency and severity categorizations) and resulting outputs (i.e., risk ratings) require subjective interpretation, and different users may obtain opposite ratings of the same quantitative risks. He further suggests that risk matrices should be used with caution, and only with careful explanations of embedded judgments.

Kester QA proposes a new evaluation, analysis and visualization of the Assessment Evaluation of Risks Matrix in software engineering based on Formal Concept Analysis, or Galois Lattices, a data analysis technique grounded on Lattice Theory and Propositional Calculus. This method considered the set of common and distinct attributes of risks levels assessment in such a way that categorization are done based on risk types. This method helped in building a more defined and conceptual systems for evaluation and visualization of risks in software engineering projects. [13]

There exist tools and techniques that are used to evaluate risks within a software project but they do not analyze risks at the tasks level within a project and also they do not perform thorough evaluation of these risks' relationships and chain of effects on other tasks within a given project. Hence there is a need for employment of effective tools and techniques in order to efficiently analyze and track risks at all levels within the life cycle of the project. Therefore this paper address such issue by proposing a method of analyzing risks at various levels of a project by employing applying graph theory.

Graph Theory is now a major tool in mathematical research, electrical engineering, computer programming and networking, business administration, sociology, economics, marketing, and communications etc. In particular, many problems can be modeled with paths formed by traveling along the edges of a certain graph. For instance, problems of efficiently planning routes for mail delivery, garbage pickup, snow removal, diagnostics in computer networks, and others, can be solved using models that involve paths in graphs.[14]

Many AI problems can be cast as the problem of finding a path in a graph. A graph is made up of nodes and arcs. Arcs are ordered pairs of nodes that can have associated costs. For a finite graph without cycles, it will eventually find a solution no matter which order you select paths on the frontier. Some strategies for selecting paths from the frontier expand fewer nodes that other strategies. As part of the definition of the algorithm a solution is only found when a goal node is selected from the frontier, not when it is added. [15]

Oliver Mason and Mark Verwoerd presented a survey of the use of graph theoretical techniques in Biology. They discussed recent works on identifying and modeling the structure of bio-molecular networks, as well as the application of centrality measures to interaction networks and researched on the hierarchical structure of such networks and network motifs. They worked on the links between structural network properties and dynamics, with emphasis on synchronization and disease propagation. [16]

Understanding complex systems often requires a bottom-up analysis towards a systems biology approach. The need to investigate a system, not only as individual components but as a whole, emerges. This can be done by examining the elementary constituents individually and then how these are connected. The myriad components of a system and their interactions are best characterized as networks and they are mainly represented as graphs where thousands of nodes are connected with thousands of vertices. Pavlopoulos, Georgios A., et al. demonstrated approaches, models and methods from the graph theory universe and discussed ways in which they can be used to reveal hidden properties and features of a network. Network profiling combined with knowledge extraction helped them to better understand the biological significance of the system in their work they did on Using graph theory to analyze biological networks. [17]

The most important element in construction procurement is the contractor selection, which can result from contractor's ranking. Contractor prequalification is essential in most construction projects, and the process has been performed by many different methods in practice. In most studies of contractor selection, selection criteria are assumed to be independent of each other. However, these criteria are likely to affect each other. Maryam Darvish, Mehrdad Yasaei and Azita Saeedi showed how graph theory and matrix methods can served as a decision analysis tool for contractor selection. Their method can be used as a decision support system by project owners in order to identify an eligible contractor to be awarded the contracts. [18] Graph theory has a range of applications from analysis of systems to optimization of search engines and computer networks. [19]

This paper proposes an effective way of analyzing risks within a given project using graph theory. The risks analysis is done at the tasks level and can be done for the entire projects. Classification of risks can also be easily achieved through graph search as well as the chain of effects of risks can easily be traced and monitored within the project (that is at the tasks level, phase level as well as the entire project). Relationships between tasks in terms of risks can also be known and traced.

III. METHODOLOGY

Within a given project, a task is an activity that needs to be accomplished within a defined period of time or by a deadline. A task can be broken down into assignments which should also have a defined start and end date or a deadline for completion. One or more assignments on a task put the task under execution. Completion of all assignments on a specific task normally renders the task completed. Tasks can be linked together to create dependencies. [20] Tasks level activities are very crucial in project management. Connected task have dependencies and a delay or inefficiency in one can affect the entire project.

Using a project management tool that provides a graphical representation of a project's timeline, PERT chart (Program Evaluation Review Technique was developed by the United States Navy for the Polaris submarine missile program in the 1950s), tasks will be represented with their associated risks as well as other attributes of the tasks. PERT charts allow the tasks in a particular project to be analyzed, with particular attention to the time required to complete each task, and the minimum time required to finish the entire project [21]. But in this work additional property which is associated risks will be added to each task.

The task together with its attributes will then be the nodes within a project and the links between tasks will then be the edges. As shown in figure 1, is a PERT chart for a project with five milestones (10 through 50) and six activities (A through F). The project has two critical paths: activities B and C, or A, D, and F – giving a minimum project time of 7 months with fast tracking. Activity E is sub-critical, and has a float of 1 month.[23]

Risks can arise from any aspect of a project. Thus, a complete identification of all project risks can only be obtained traditionally by involving a sufficient number of people to ensure that in-depth competence and experience is applied to the process for all significant aspects of the project scope. Some project risks can be identified by simply deducing the defined project risks that are applicable to the project. .

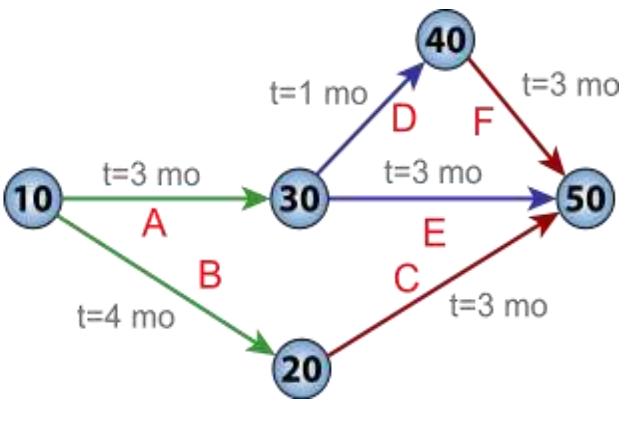

Fig 1 PERT network chart for a seven-month project with five milestones (10 through 50) and six activities (A through F).[23]

ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 2, Issue 2, February 2013

Using graph theory, analysis can now be performed in the evaluation of the risks efficiently in order to track risks at all levels within the life cycle of the project (that is at the tasks level, phase level as well as the entire project).. Therefore this paper address such issue by proposing a method of analyzing risks at various levels of a project by employing applying graph theory.

IV. APPLICATIONS AND RESULTS

A given project with nine tasks ranging from 'a' to 'i' was looked at. At each task level were associated project risks as well as the duration and their respective dependencies. Table 1 below consist of the summary of the project tasks attributes.

Table 1 Project tasks with dependencies

TASKS ID	RISKS				DURATION	DEPENDENCIES
	R1	R2	R3	R4	DT(days)	Dp
a	x		x		8	
b		x			10	
c	x		x		8	a,b
d	x			x	9	a
e			x		5	b
f		x		x	3	c,d
g	x		x		2	d
h		x		x	4	f,g
i	x			x	3	e,f

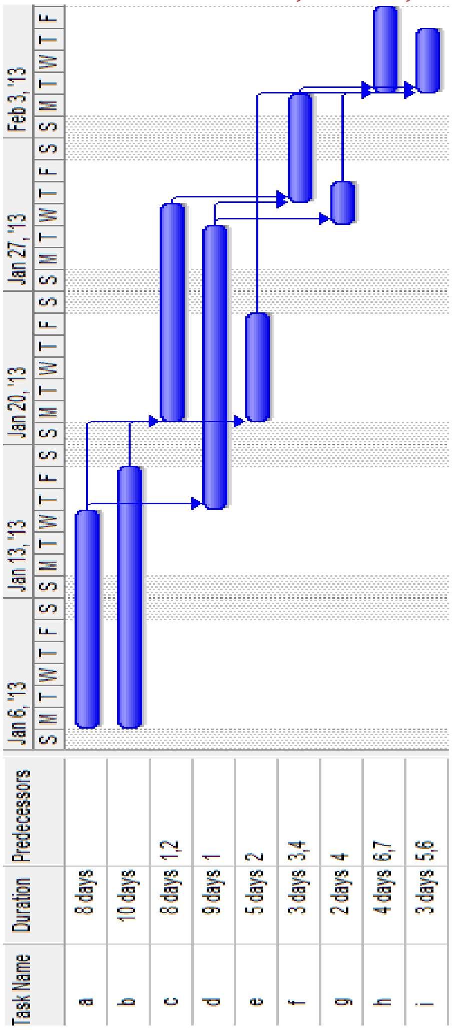

Fig 2 Gantt chart from table 1

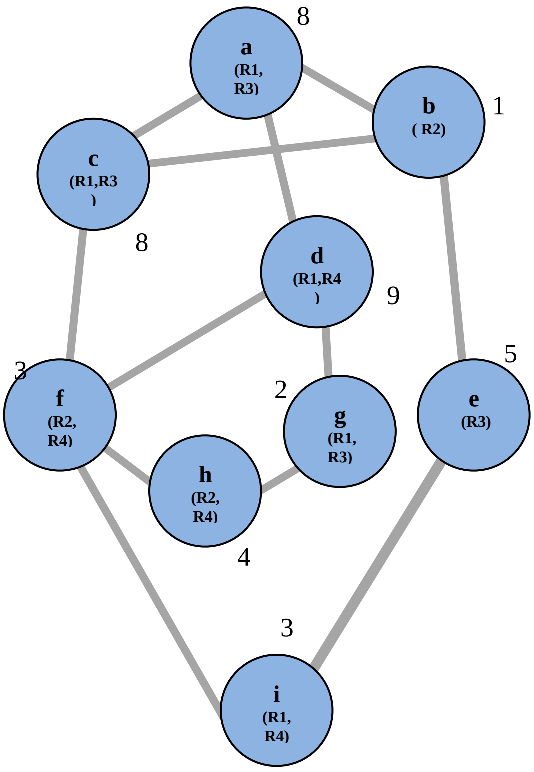

Fig 3 PERT network chart from table 1

Center Vertices
1 piece: f;

Vertices Degree
a, degree: 3
b, degree: 3
c, degree: 3
d, degree: 3
e, degree: 2
f, degree: 4
g, degree: 2
h, degree: 2
i, degree: 2

Adjacent Vertices For Vertex
a, adjacent vertices (3 pieces): b, c, d,
b, adjacent vertices (3 pieces): a, c, e,
c, adjacent vertices (3 pieces): a, b, f,
d, adjacent vertices (3 pieces): a, f, g,
e, adjacent vertices (2 pieces): b, i,
f, adjacent vertices (4 pieces): c, d, h, i,
g, adjacent vertices (2 pieces): d, h,
h, adjacent vertices (2 pieces): f, g,
i, adjacent vertices (2 pieces): e, f,

A vertex v in a connected graph G has eccentricity e if the maximum of the lengths of the shortest paths to the other vertices of G is e. That is the eccentricity ∈ (v) of a graph vertex v in a connected graph G is the maximum graph distance between v and any other vertex u of G. For a disconnected graph, all vertices are defined to have infinite eccentricity.

Eccentricity
From vertex 'a' to vertex 'a': 0
From vertex 'a' to vertex 'b': 1
From vertex 'a' to vertex 'c': 1
From vertex 'a' to vertex 'd': 1
From vertex 'a' to vertex 'e': 2
From vertex 'a' to vertex 'f': 2
From vertex 'a' to vertex 'g': 2
From vertex 'a' to vertex 'h': 3
From vertex 'a' to vertex 'i': 3
From vertex 'b' to vertex 'a': 1
From vertex 'b' to vertex 'b': 0
From vertex 'b' to vertex 'c': 1
From vertex 'b' to vertex 'd': 2
From vertex 'b' to vertex 'e': 1
From vertex 'b' to vertex 'f': 2
From vertex 'b' to vertex 'g': 3
From vertex 'b' to vertex 'h': 3
From vertex 'b' to vertex 'i': 2
From vertex 'c' to vertex 'a': 1
From vertex 'c' to vertex 'b': 1
From vertex 'c' to vertex 'c': 0
From vertex 'c' to vertex 'd': 2
From vertex 'c' to vertex 'e': 2
From vertex 'c' to vertex 'f': 1
From vertex 'c' to vertex 'g': 3
From vertex 'c' to vertex 'h': 2
From vertex 'c' to vertex 'i': 2
From vertex’d’ to vertex 'a': 1
From vertex’d’ to vertex 'b': 2
From vertex 'd' to vertex 'c': 2
From vertex 'd' to vertex 'd': 0
From vertex 'd' to vertex 'e': 3
From vertex 'd' to vertex 'f': 1
From vertex 'd' to vertex 'g': 1
From vertex 'd' to vertex 'h': 2
From vertex 'd' to vertex 'i': 2
From vertex 'e' to vertex 'a': 2
From vertex 'e' to vertex 'b': 1
From vertex 'e' to vertex 'c': 2
From vertex 'e' to vertex 'd': 3
From vertex 'e' to vertex 'e': 0
From vertex 'e' to vertex 'f': 2
From vertex 'e' to vertex 'g': 4
From vertex 'e' to vertex 'h': 3
From vertex 'e' to vertex 'i': 1
From vertex 'f' to vertex 'a': 2
From vertex 'f' to vertex 'b': 2
From vertex 'f' to vertex 'c': 1
From vertex 'f' to vertex’d’: 1
From vertex 'f' to vertex 'e': 2
From vertex 'f' to vertex 'f': 0
From vertex 'f' to vertex 'g': 2
From vertex 'f' to vertex 'h': 1
From vertex 'f' to vertex 'i': 1
From vertex 'g' to vertex 'a': 2
From vertex 'g' to vertex 'b': 3
From vertex 'g' to vertex 'c': 3
From vertex 'g' to vertex’d’: 1
From vertex 'g' to vertex 'e': 4
From vertex 'g' to vertex 'f': 2
From vertex 'g' to vertex 'g': 0
From vertex 'g' to vertex 'h': 1
From vertex 'g' to vertex 'i': 3
From vertex 'h' to vertex 'a': 3
From vertex 'h' to vertex 'b': 3
From vertex 'h' to vertex 'c': 2
From vertex 'h' to vertex’d’: 2
From vertex 'h' to vertex 'e': 3
From vertex 'h' to vertex 'f': 1
From vertex 'h' to vertex 'g': 1
From vertex 'h' to vertex 'h': 0
From vertex 'h' to vertex 'i': 2
From vertex 'i' to vertex 'a': 3
From vertex 'i' to vertex 'b': 2
From vertex 'i' to vertex 'c': 2
From vertex 'i' to vertex’d’: 2
From vertex 'i' to vertex 'e': 1
From vertex 'i' to vertex 'f': 1
From vertex 'i' to vertex 'g': 3
From vertex 'i' to vertex 'h': 2
From vertex 'i' to vertex 'i': 0

From table 1 a Gantt chart was generated using Microsoft office project with the tasks duration and dependencies as shown in figure 2. But risks cannot be factored in at the tasks level within the Gantt chart. Using graph theory and PERT chart with associated risks types at the tasks level, nodes as vertices with their links as edges was obtained as shown in figure 3 above. Visualization of risks at all levels can be seen and their effects can easily be analyzed. The centre of the graph, adjacent vertices, and eccentricity of vertices was obtained from the graph (figure 3) as well as vertices degrees.

A search was then initialize based on graph search algorithm to categorize the risks with the most occurring risks and their associated tasks. The risks with more tasks were arranged at the top (from left to right) and the least at the bottom as shown in figure 4 below.

The graph searching algorithm has the following steps:
Repeat

- Select a path on the frontier that is having risks type and count distinct risks available. Let's call the path selected P.
- if P is a path to a goal node, stop and return risks types and tasks with their relationships,
- remove P from the frontier
- for each neighbor of the node at the end of P, extend P to that neighbor and add the extended path to the frontier.

Continue until the frontier is empty. When it is empty there are no more solutions based on the search.
At the end the results was analyzed and the graph below was obtained for the risks.

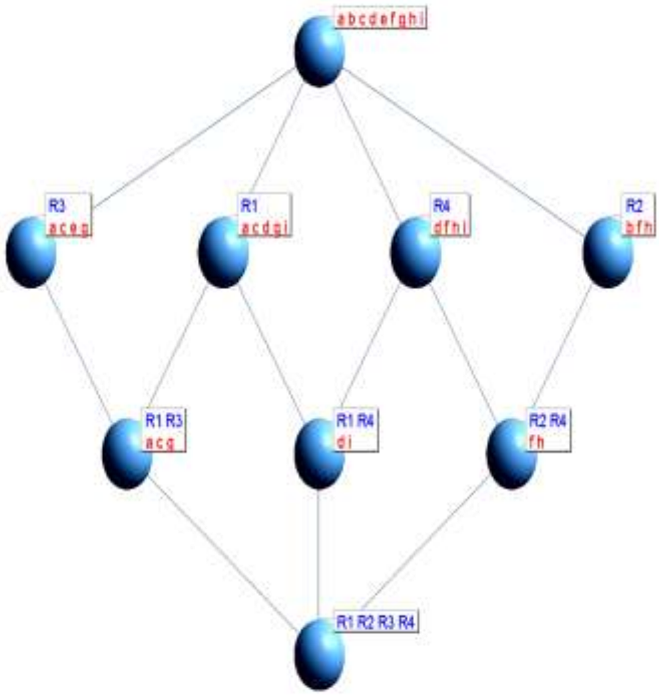

Fig 4 relationships between risks types and tasks

Figure 4 indicated how tasks share common risks types within the project with the most occurring risks appearing at the top of the graph with their associated tasks. This makes it easy for common risks identification within projects.

V. CONCLUSION

With the proposed method, an effective way of analyzing risks within a software project at the task level within software engineering projects can now be achieved and also the approach can now be integrated into existing project management software for effective risks analysis as well as visualization. Each risks associated with each task can easily be analyze to determine its effect on neighbor tasks as well as tracking of the chain of effects of risks within a given project.

REFERENCES

[1] Westfall, Linda. "Software risk management." ANNUAL QUALITY CONGRESS PROCEEDINGS-AMERICAN SOCIETY FOR QUALITY CONTROL. ASQ; 1999, 2000.

[2] O'reilly "Applied Software Project Management : Software Project Planning Practices arrow :Risk Plan " Retrieved from: http://www.stellman-greene.com/aspm/content/view/19/38/

[3] Risk Management Guide for DoD Acquisition Defense Acquisition Management College (DSMC) Press P6-5 retrieved from http:www.stsc.hill.af.mil/resources/tech_docs/gsam3/chap6.pdf

[4] Mike Griffiths, "The Top Five Software Project Risks" Retrieved from: http://www.projectsmart.co.uk/top-five-software-project-risks.html.

[5] Software Engineering. Risk Management. retrieved from : http://openseminar.org/se/modules/21/index/screen.do

[6] Roy, G.G.; , "A risk management framework for software engineering practice," Software Engineering Conference, 2004. Proceedings. 2004 Australian , vol., no., pp. 60- 67, 2004

[7] C. Alberts, and A. Dorofee, "Risk Management Framework," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Report CMU/SEI-2010-TR-017, 2010. http://www.sei.cmu.edu/library/abstracts/reports/10tr017.cfm

[8] Risk matrix retrieved from : http://en.wikipedia.org/wiki/Risk_Matrix

[9] United States Department of Defense, Risk Management Guide for DoD Acquisition, August 2006

[10] Goddard Space Flight Center, NASA, Risk Management Reporting, GSFC-STD-0002, 8 May 2009

[11] International Standards Organization, Space Systems Risk Management, ISO 17666,

[12] Cox, L.A. Jr., 'What's Wrong with Risk Matrices?', Risk Analysis, Vol. 28, No. 2, 2008, doi:10.1111/j.1539-6924.2008.01030.x

[13] Kester, Quist-Aphetsi. "Application of Formal Concept Analysis to Visualization of the Evaluation of Risks Matrix in Software Engineering Projects." International Journal of Science, Engineering and Technology Research [Online], 2.1 (2013): pp:220-225.

[14] Application to Graph theory Retrieved from : http://aix1.uottawa.ca /~jkhoury/graph.htm

[15] Wille.R, 1989."Knowledge acquisition by methods of formal concept analysis", in E.Diday, editor, Data analysis, Learning Symbolic and numeric knowledge, pages 365-380.1989.

[16] Mason, Oliver, and Mark Verwoerd. "Graph theory and networks in biology." Systems Biology, IET 1.2 (2007): 89-119.

[17] Pavlopoulos, Georgios A., et al. "Using graph theory to analyze biological networks." BioData mining 4.1 (2011): 10.

[18] Darvish, Maryam, Mehrdad Yasaei, and Azita Saeedi. "Application of the graph theory and matrix methods to contractor ranking." International Journal of Project Management 27.6 (2009): 610-619.

[19] Kester, Quist-Aphetsi. "Optimization of Searches on Social Networks using Graph theory and Formal Concepts Analysis." International Journal of Advanced Research in Computer Engineering & Technology(IJARCET) [Online], 2.1 (2013): pp:293-303.

[20] Task (project management) Retrieved from : http://en.wikipedia.org/ wiki/Task_%28project_management%29

[21] Investopedia, Organizational Theory Retrieved from : http://www.investopedia.com/terms/p/pert-chart.asp#axzz2JsffTtB6

[22] Program Evaluation and Review Technique Retrieved from : http://en.wikipedia.org/wiki/Program_Evaluation_and_Review_Technique

Quist-Aphetsi Kester, MIEEE: is a global award winner 2010 (First place Winner with Gold), in Canada Toronto, of the NSBE's Consulting Design Olympiad Awards and has been recognized as a Global Consulting Design Engineer. He is a law student at the University of London UK. He is a PhD student in Computer Science. The PhD program is in collaboration between the AWBC/ Canada and the Department of Computer Science and Information Technology (DCSIT), University of Cape Coast. He had a Master of Software Engineering degree from the OUM, Malaysia and BSC in Physics from the University of Cape Coast-UCC Ghana.

He has worked in various capacities as a peer reviewer for IEEE ICAST Conference, IET-Software Journal, lecturer, Head of Digital Forensic Laboratory Department at the Ghana Technology University and Head of Computer science department. He is currently a lecturer at the Ghana Technology University College.