Risk Model For Software Development Personnel

Esiefarienrhe Michael Bukohwo

***Abstract*—Availability of adequate personnel to commence and sustain a software project is a vital component that determines the success/sustainability of the project. This paper proposed a probability model to calculate the success or failure of a software development project from the project personnel database and subsequently proposes a risk table for interpretation of the results. A further graphical tracking method was utilized to track the movement of personnel during the project development circle. Finally, the model was implemented using statistics from on-going software project and the results shows that inexperienced personnel were responsible for the project failure as far reaching solutions were proposed to prevent future project failure arising from personnel availability.**

Index Terms— **Personnel Risk, Risk Mitigation, Risk Management, Development Risk**

I. INTRODUCTION

Large software projects will never be without some risks but if risks can be brought down to acceptable levels that will be a good beginning [1].

Everyone has an intuitive understanding of risk but how can understanding risk help software to be more successful? First, we need to understand that a risk is not a problem. Rather, a risk is something that might occur in the future: a possibility, not a certainty. To be technically precise, there are two factors that comprise a risk:

1. Probability or likelihood that it will occur.
2. Loss resulting from its occurrence.

The term "risk" has been erroneously used as a synonym of "uncertainty" and "threat" [2] [3] [4]. Risk in software is viewed as a measure of the likelihood of unsatisfactory outcome and a loss affecting the software from various perspectives: project, process and product [2] [4]. However, this definition of risk is misleading because it confounds the concepts of risk and uncertainty. According to [5] most part of decision-making in software processes are under uncertainty rather than risk. Uncertainty is a situation in which the probability distribution for the possible outcome is not known.

Manuscript received August 01, 2014; revised February 2, 2015
Esiefarienrhe Michael Bukohwo is with the Department of Computer Science, North West University, South Africa; Phone: +27 (0) 838650429; e-mail: Michael.Esiefarienrhe@nwu.ac.za
The author hereby acknowledges the financial and administrative support of the Faculty Research Committee (FRC) of FAST of the North West University, Mafikeng Campus, South Africa.
esiefabukohwo@gmail.com

[5] Therefore defined risk as the product of the value of an outcome times its probability of occurrence. While risk indicates a probabilistic outcome; threat is used to identify the danger that can occur. [6] sees risk as a function of the likelihood of a given threat-source's exercising a particular potential vulnerability, and the resulting impact of that adverse event on the organization.

[7] Defined risk as the probability that an asset will suffer an event of a given negative impact is determined from various factors: the ease of executing an attack, the attacker's motivation and resources, a system's existing vulnerabilities, and the cost or impact in a particular business context. According to [7] a threat, or danger source, is invariably the danger a malicious agent poses and the agent's motivations (financial gain, prestige, and so on). Threats manifest themselves as direct attacks on system security. Vulnerability is a defect or weakness in system security procedure, design, implementation, or internal control that an attacker can compromise. It can exist in one or more components making up a system, even if those components are not necessary involved with security functionality. For every risk, a software designer can put controls in place that either prevents or (at a minimum) detects the risk when it does occur.

[8] view risks as those factors that may prevent the attainment of a set goal. He defined the goal of a software development process as, "Producing a product that meets or closely matches the needs of the people for whom it is developed". Achieving this goal is called "product integrity". He further said that product integrity must also include "the additional goals of meeting the planned cost, and meeting the planned schedule for producing that product". Therefore, project risk would involve anything that may compromise the attainment of project integrity.

From the above definitions, risk can be looked at from a number of different perspectives. First, risk concerns future happenings. Second, risk involves change. The third aspect of risk involves choice.

There are two types of risk [2] namely:

1. Dynamic risks
2. Static risks.

Software risk is a dynamic risk, as it has aspects of both gain and loss associated with them. The risks and their impact typically vary with time and circumstances. A vital part of dynamic risk that software engineers must concern themselves in software development is the various technical risk associated with a project. So what is technical risk? Technical risks are the possibility that the application of software engineering theory, principles, and techniques will fail to yield the right software product. Technical risk is comprised of the underlying technological factors that may cause the final product to be overly expensive, delivered late or unacceptable to the customer.

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Technical risk lies at the heart of many problems causing the failure of software programs today. The essence of technical risk is the failure to build the right product. As the Government Accounting Office (GAO/PEMD-86) pointed out, the ultimate consequence of a risk is that the delivered system will not perform as needed. The final risk always belongs to the customer.

The various definitions and views on risk are quite elaborate. This research is geared towards investigating those risks that are inherent, undermines and contributes negatively to developing efficient, safe and reliable software products and to develop methodologies for their mitigation.

Software risk management has been recognized as one critical issue in software development since efficiently mitigating risks can directly reduce the cost of design, testing and implementing software projects [2]. Risks are viewed as potential problems which may cause software project failures either in terms of functional failure of the software, quality deficiency of the product, or cost/schedule problems of the project itself. Among all the risks, personnel risk, related to capabilities and activities of employees, is the one which tends to be ignored, but may greatly impact the product quality. Though many approaches were proposed to mitigate software risks, very few of them ever addressed this issue.

According to [10], software development risk was defined as the exposure to one or more of four types of risk:

i. Performance risk, or the failure to obtain all of the anticipated benefits of the systems and software under development
ii. Cost risk, or significantly exceeding budgeted or estimated cost
iii. Schedule risk, or the failure to deliver satisfactory software products by scheduled milestones and user need dates
iv. Support risk, or the delivery of a product that has excessive lifecycle maintenance costs due to deficiencies in maintainability, flexibility, compatibility or reliability

This paper is organized into six sections. An introduction to the concept of risk management is given section 1. Related researches on software risk management are presented in section 2. Section 3 contains the proposed model formulated for mitigating personnel related risk. A graphical method for tracking personnel risk is presented and discussed in section 4 while section 5 presents a discussion on the application of the proposed model to the actual mitigation of risk. Section 6 presents a summary and conclusion of the paper.

II. RISK ASSESSMENT

Various researchers [2] [11] [12] [13] agreed that risk assessment is the first process in the risk management methodology. It is used to determine the extent of the potential threat and the risk associated with an IT system throughout its SDLC. This process helps to identify appropriate controls for reducing or eliminating risk during the risk mitigation process.

The risk assessment methodology encompasses nine primary steps [6] namely:

i. System Characterization
ii. Threat Identification
iii. System Weakness Identification
iv. Control Analysis
v. Likelihood Determination
vi. Impact Analysis
vii. Risk Determination
viii. Control Recommendations
ix. Results Documentation

Steps 2, 3, 4, and 6 can be conducted in parallel after Step1 has been completed.

i System Characterization

This process involves the identification of the system along with the resources and the information that constitute the system. It establishes the scope of the risk assessment effort, delineates the operational authorization (or accreditation) boundaries, and provides information (e.g., hardware, software, system connectivity, and responsible division or support personnel) essential to defining the risk.

ii. Threat Identification

A threat is the potential for a particular threat-source to successfully exercise a particular vulnerability. Vulnerability is a weakness that can be accidentally triggered or intentionally exploited. A threat-source does not present a risk when there is no vulnerability that can be exercised. In determining the likelihood of a threat, one must consider threat-sources, potential vulnerabilities, and existing controls.

iii. System Weakness Identification

The analysis of the threat to an IT system must include an analysis of the weaknesses associated with the system environment. The goal of this step is to develop a list of system vulnerabilities (flaws or weaknesses) that could be exploited by the potential threat-sources.

iv. Control Analysis

The goal of this step is to analyze the controls that have been implemented, or are planned for implementation, by the organization to minimize or eliminate the likelihood (or probability) of a threat's exercising a system vulnerability.

Controls method includes both technical and non-technical controls.

v. Likelihood Determination

To derive an overall likelihood rating that indicates the probability that a potential vulnerability may be exercised within the construct of the associated threat environment, the following governing factors must be considered:

• Threat-source motivation and capability
• Nature of the vulnerability
• Existence and effectiveness of current controls.

The likelihood that a potential vulnerability could be exercised by a given threat-source can be described as high, medium, or low.

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

vi. Impact Analysis

The next major step in measuring level of risk is to determine the adverse impact resulting from a successful threat exercise of vulnerability. Before beginning the impact analysis, it is necessary to obtain the following necessary information:

• System mission
• System and data criticality
• System and data sensitivity.

vii. Risk Determination

The purpose of this step is to assess the level of risk to the IT system. The determination of risk for a particular threat/vulnerability pair can be expressed as a function of:

• The likelihood of a given threat-source's attempting to exercise a given vulnerability
• The magnitude of the impact should a threat-source successfully exercise the vulnerability
• The adequacy of planned or existing security controls for reducing or eliminating risk.

To measure risk, a risk scale and a risk-level matrix must be developed.

viii. Control Recommendations

During this step of the process, controls that could mitigate or eliminate the identified risks, as appropriate to the organization's operations, are provided. The goal of the recommended controls is to reduce the level of risk to the IT system and its data to an acceptable level.

ix. Results Documentation

Once the risk assessment has been completed (threat-sources and vulnerabilities identified, risks assessed, and recommended controls provided), the results should be documented in an official report or briefing. A risk assessment report is a management report that helps senior management, the mission owners, make decisions on policy, procedure, budget, and system operation and management changes. Unlike an audit or investigation report, which looks for wrongdoing, a risk assessment report should not be presented in an accusatory manner but as a systematic and analytical approach to assessing risk so that senior management will understand the risks and allocate resources to reduce and correct potential losses. For this reason, some people prefer to address the threat/vulnerability pairs as observations instead of findings in the risk assessment report.

III. RISK MITIGATION

Risk mitigation, the second process of risk management, involves prioritizing, evaluating, and implementing the appropriate risk-reducing controls recommended from the risk assessment process. Because the elimination of all risk is usually impractical or close to impossible, it is the responsibility of senior management and functional and business managers to use the least-cost approach and implement the most appropriate controls to decrease mission risk to an acceptable level, with minimal adverse impact on the organization's resources and mission.

Risk mitigation is a systematic methodology used by senior management to reduce mission risk.

Risk mitigation can be achieved through any of the following risk mitigation strategies:

• Risk Assumption**.** To accept the potential risk and continue operating the IT system or to implement controls to lower the risk to an acceptable level
• Risk Avoidance**.** To avoid the risk by eliminating the risk cause and/or consequence (e.g., forgo certain functions of the system or shut down the system when risks are identified)
• **Risk Limitation.** To limit the risk by implementing controls that minimizes the adverse impact of a threat exercising a vulnerability (e.g., use of supporting, preventive, detective controls)
• **Risk Planning.** To manage risk by developing a risk mitigation plan that prioritizes, implements, and maintains controls
• **Research and Acknowledgment.** To lower the risk of loss by acknowledging the vulnerability or flaw and researching controls to correct the vulnerability.
• Risk Transference**.** To transfer the risk by using other options to compensate for the loss, such as purchasing insurance.

An examination of the above researches reveal that personnel risk was not clearly regarded as a software development risk. This is hitherto a wrong assumption as the anchor and support for any software development project begins and end with the personnel. They provide the blueprint, follow, execute and evaluate the developmental efforts to ensure that the projects do not derail. That is an enough task. Software development Personnel are central to the product development process as team members accomplish the important tasks of articulating product specifications and transforming them into the design, development, and implementation of new products. In particular, team composition has been recognized as critical to the success of product development efforts. In attempt to effectively manage project personnel, managers are often faced with various personnel related risks that posed threat to project successful completion.

IV. RELATED WORK

[5] developed a Risk Assessment Model that addresses project risks related to schedule and budget, and focus mainly on project completion time. The model uses metrics and worked on sensitivity to requirement volatility. [14] based his work on the principles of advocated by Dr. Robert Charette risk management principles and consists of five steps namely identify risks, characterize risks, prioritize risks, avert risks, track/control risks. [15] developed METRIX to predict high risk software component based on experts' opinions and historical data. Many approaches have been tried to address software risk management. [16] proposed Software Risk Assessment Model (SRAM) which

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

considers nine critical risk elements. [17] investigated software risk control based on Capability Maturity Model (CMM). A concept to deal with risks at design level was presented by [7]. [18] incorporates logical fault trees of probabilistic risk assessment (PRA) into the "Defect Detection and Prevention" (DDP). While risk management database was used by [19] to manage risks.

V. THE PROPOSED MODEL

Given a software development project, the researchers proposed model to calculate the probability of the project success given statistics on personnel available for the project. It uses the data for planned personnel and the actual personnel for the project to calculate the probability that the project will be successful. The value obtained from the model is compared to a formulated risk table in Table 1 to determine the likelihood of the project success under personnel risk.
With this likelihood in mind, the project is actually tracked during development by monitoring the experienced personnel against inexperienced personnel and deviations that can lead to the project failure are prevented.

Where
k = Planned personnel for the project
a = Actual personnel available for the project
p = Prob. of success
q = Prob. of failure
P(p,q;a,k) is evaluated against risk table to determine the likelihood of the project under risk.

Table 1: Risk Table

Intervals	Risks Classification
0.1 - 0.24	very low risk (Project success)
0.25 - 0.49	low risk
0.50 - 0.74	high risk
0.75 - 1.0	too high risk (Failure of project)

V. TRACKING OF PERSONNEL DURING DEVELOPMENT

The researchers proposed a tracking method in conjunction with the use of equation 1.0 to mitigate personnel availability risk. Software development personnel are classified into two categories namely experienced and inexperienced staff. This classification is done with a measure of staff experience, or software personnel qualification, which deals with individual team members' proficiencies. Staff experience or qualification level can refer to several different things:

- educational level
- years of experience with the company, indicating a knowledge of company standards as well as loyalty and dedication
- years of software development experience
- years of experience in the domain
- years of experience in the language
- years of experience on similar projects
- amount of specialty training
- years of relevant specialty training

All of these measurements data should be made available to managers (from the project database) who are trying to assess whether their team has sufficient experience to complete the planned project. None of this experience necessarily guarantees capability, but these types of measurements give managers a tool to determine whether their proposed team members can perform the tasks required of them. Since risk identification is an ongoing process, measures like the above can periodically be reviewed if other indicators point to staffing problems. Managers must be careful to include any experience obtained on the current project if an interim analysis is done.

VI. APPLICATION OF MODEL TO MITIGATE PERSONNEL RISK

The data used for the simulation of the Personnel Availability Graphical metrics were collected from the software for online e-procurement system. This software was developed by a private computer firm for the e-bidding system as part of the Nigeria e-Government efforts. The data extracted from the database was used to track personnel strength for the project as shown in Fig. 1, Fig. 2 and Fig. 3. The total staff planned to handled the software development project nineteen (19) and but the project could only commenced with fourteen (14) actual staff. Using the model as presented in equation 1.0, this initial statistics about the project can be used to quickly compute the probability of the project success/failure. Applying the model to, we obtain the probability that the project will succeed as P(pr)= 0.252343536.

Evaluating the probability value against the risk table in Table 1 reveals that this project has a low risk of failure. It implies that if adequate tracking is done, the project is sure to succeed.
Having collected the personnel related data, performance was monitored looking for trends to indicate whether or not a risk item is under control, as well as to indicate that a new, previously unidentified item may be at risk.
Fig. 1 represents the total personnel available for the project. It shows the total planned experienced staff and inexperienced staff for the project. As the project progresses with time (x-axis) in Figures, the actual experienced staff and inexperienced staff available and working on the project are monitored and graphed as shown in Fig. 1, Fig. 2 and Fig. 3. Any deviation from the expected plan triggers immediate attention and this causes for further analysis.

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

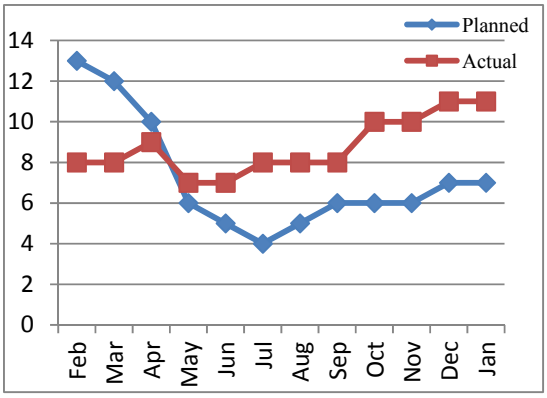

Fig 1: Planned Experienced Staff Vs Actual Experienced staff

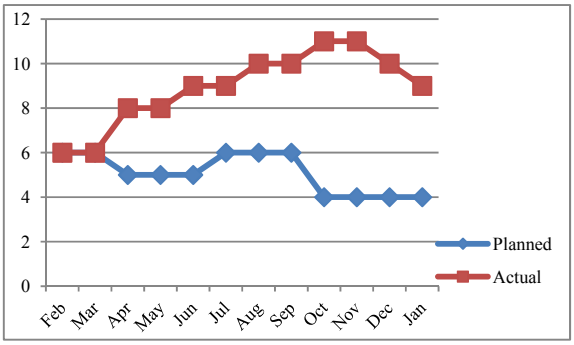

Fig 2: Planned Inexperienced Staff Vs Actual Inexperienced

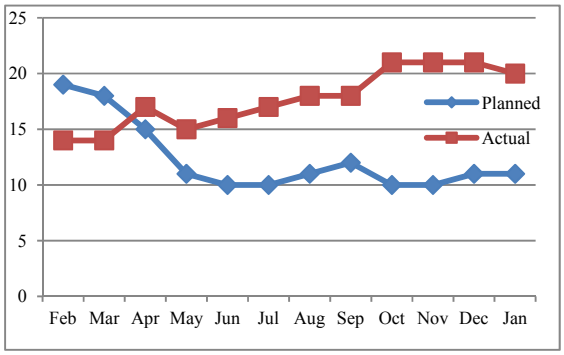

Fig 3: Total Planned Staff Vs Total Actual Staff

From Fig. 3, it is evident that the project manager could not get adequate team ready to commence the project. The project was estimated to commence with 19 staff made up of 13 experienced (Fig. 1) and 6 inexperienced staff (Fig. 2). The project commenced with unexpected staff loss of (-5) which were experienced staff. This is an indication that the project might be at risk. Further tracking revealed that 3months into the project, the experienced staff were leaving the project and in a bid to sustain the development, more of inexperienced staff was brought in far above the planned (Fig. 2). The consequence of this is inadequate requirement elicitation, poor problem analysis, and unstructured database design. The project derailed and almost crumpled. A combined effort from management coupled with pressure from client saved the project from failing as this lead to conceding vital tradeoff of cost and time to get the project on track. In this bid to sustain and complete the project, management went into desperate recruiting of experienced staff (unexpected gain of (+4). This effect of this is a reduction in the staff strength of inexperienced staff below what was planned for.

It should interest analyst to know that a project with this characteristics cannot stand the test of time. The project will be error-prone, unspecified requirement, missing functions, wrong data structure implementation, and even wrong use of objects. The vital stages of the project (Requirement elicitation and feasibility study) were handled by staff with no experience. A lot of things could have gone wrong. When finally the project was completed, the organization overshot its planned staff strength by (+4) with over 60% of this as experienced staff. A project that was planned to be completed within 9 months was completed in 12 months.

A vital lesson to learn here is: a project that does not have enough experienced personnel or that tries to bring too many into the project towards the end of the schedule is a project at risk.

From Fig. 3, the total staff curve should grow through the requirements and design phases, peak in code and early test, and begin to fall as acceptance and integration tests are completed. This was not the case with this project.

As seen in Figures 1, 2 and 3, personnel leaving the project and adding new personnel introduces delay in the project due to the learning curve. Such project may also stand the risk of non-completion due to impairing project knowledge, and eroding the knowledge-based.

VII. CONCLUSION

Software project analysts need to manage risk and use every tool available to them for this purpose. If they can use the tools that are already being used on their project for other purposes, they save themselves time and money. Most analysts use some form of metrics program to track their project for cost, schedule, effort, and quality. Many of the measures that are used to assist them with project management can also serve additional purpose of identifying and tracking risk.

The software development personnel are instrumental to the development of sound and effective software. Non-availability or inadequate supply of software personnel will constitute risk that may lead to the failure of the project. This paper proposed a model for a common software risk, personnel shortfall, to show how analysts can use measures and metrics to help identify and track risk items. The model is capable of monitoring personnel on a project by charting the total estimated (planned) personnel for the project against the personnel available for the project. The variances from the curves were monitored and analyzed using the graphical method. It was emphasized that project that does not have enough experienced personnel or that tries to bring too many into the project towards the end of the schedule is a project

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

that is at risk. The model can also be applied to other common risk items, such as requirements changes and unrealistic schedules and budgets, to help analysts have visibility into and control over their overall projects in addition to identifying and monitoring their risk items.

.

REFERENCES

[1] C. Jones, "Assessment and Control of software Risks", Yourdon Press, Prentice Hall, Englewood Cliffs, NJ, 1994.

[2] R. P. Higuera and Y. Y. Haimes, "Software Risk Management", Technical Report CMU/SEI-96-TR-012, ESC-TR-96-012, 1996.

[3] D. Karolak, "Software Engineering Management", IEEE Computer Society Press, Washington DC. 1996.

[4] E. Hall, "Managing Risk, Methods for SoftwareSystems Development, Addison Wesley, 1977.

[5] J. Nogueira, "A Formal Risk Assessment Model for Software Projects". Ph.D Dissertation. Naval Postgraduate School, 2000

[6] G. Stoneburner, A. Goguen and A. Feringa, "Risk Management Guide for Information Technology Systems", NIST Special Publication 800-30, 2002.

[7] G. McGraw, "Risk Analysis in Software Design", *IEEE Security & Privacy*, pp. 32-37, 2004.

[8] L. Bersoff, "Elements of Software Configuration Management", *IEEE Transactions on Software Engineering*, 10(1): 79-87, 1984.

[9] GAO/PEMD-86, Government Accounting Office. "*Technical Risk Assessment: The Current Status of DOD Efforts",* GAO/PEMD-86-5, Government Accounting Office, Washington, D.C., 1986.

[10] Software Management Guide Volume II, "Software Technology Service Center", Hill AFB, Ut, 1993.

[11] B. Boehm, "A Spiral Model of Software Development and Enhancement", *IEEE Computer Society* 21(5): 61-72, 1988.

[12] R. Fairley, "Risk Management for Software Projects", *IEEE Software*, 11(3): Pp. 57-67, 1994.

[13] M. Lyu, "Software Reliability Engineering", IEEE Computer Society Press, 1995.

[14] J. Rockwell, "Risk Management", Rockwell Job Aid, Rockwell, 1995.

[15] B. E. Gayet and L. C. Briand, "METRIX: a tool for software-risk analysis and management", *Annual Reliability and Maintainability Symposium*, pp. 310-314, 1994.

[16] S. Foo and A. Muruganantham, "Software Risk Assessment Model", *IEEE International Conference on Management of Innovation and Technology* (ICMIT), vol. 2, pp. 536-544, 2000.

[17] R. Xu, L. Qian and X. Jing, "CMM-based software risk control optimization", *IEEE International Conference on Information Reuse and Integration* (IRI), pp. 499-503, 2003.

[18] M. S. Feather, "Towards a Unified Approach to the Representation of, and Reasoning with, Probabilistic Risk Information about Software and its System Interface", *15th International Symposium on Software Reliability Engineering* (ISSRE), Pp. 391-402, 2004.

[19] L. Fussell and S. Field, S. "The Role of the Risk Management Database in the Risk Management Process", *18th International Conference on System Engineering* (ICSEng), Pp. 364-369, 2005.

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)