# Risk Framework for Open Source Applications Using Agent Oriented Modelling

Jo Lyn Teh[1], Moshiur Bhuiyan[2], P. W. C. Prasad[1], and Aneesh Krishna[3*]

[1]School of Computing and Mathematics, Charles Sturt University, Sydney, NSW 2010, Australia.
[2] Service Consulting, Enterprise Cloud Systems Pty Limited, Sydney, NSW 2560, Australia.
[3] Department of Computing, Curtin University, Perth, WA 6102, Australia.

Manuscript submitted May 3, 2016; accepted August 12, 2016.
* Corresponding author. Email: a.krishna@curtin.edu.au
doi: 10.17706/jsw.11.9.833-847

**Abstract:** This paper aims to propose a novel risk framework utilizing agent oriented modelling in order to provide guidelines for the adoption of Open Source Software (OSS) solutions to an organization. Multiple studies have focused on user requirements and technical aspects when implementing OSS solutions. However, there is a lack of literature focusing on the social aspects of 'why' and 'how' risks of OSS adoption can impact the overall organizational goals and the OSS ecosystem. Therefore, another focus area of this work is to bridge the gap between actors (goals, softgoals, task, and resources) and risk factors when adopting OSS in the organization by proposing new risk measurement and risk prioritization methods. This study is based on literature survey and a case study on OSS adoption in an organization. The research results show that the proposed risk measurement validates the current methods while providing deeper insights into the criticality and strong relationship between actors which in turn assists in proposing appropriate risk response strategies such as delegation of risk among these actors. The result implies that taking ecosystems into consideration is important for risk mitigation.

**Key words:** Open source software (OSS), *i**, IT risk management, risk framework, organizational modelling.

## 1. Introduction

Given the constant advances in Information Technology (IT), many organizations download and install IT software applications as solution for IT infrastructure support and as means of achieving business goals, leading to an emerging trend of adopting Open Source Software (OSS) within organizations; this is particularly important in terms of data mining purpose to help organization make informed decisions on business strategies and strategic intent. OSS is defined as "software that must be distributed with source code included or be easily available" for free of cost usually via online download [1]. Despite the advantages, organizations are still hesitant to adopt OSS because the risks and challenges it poses. Some examples of risk factors that can be seen when adopting OSS solutions are OSS policies and governance, architecture incompatibility, lack of human resource capabilities and inactive OSS ecosystem support services [2]-[6].

Many studies have examined 'what' OSS solutions can bring to an organization. However, there is a lack of literature on the social aspects of 'why' and 'how' risks of OSS adoption can impact the actors in an organization and their goals [7]. To begin with, there are currently many risk frameworks that act as guidelines for effective risk management including ISO31000:2009, NIST800-39 and COBIT. NIST800-39 and COBIT are best suited for project management with a focus on governance whereas ISO31000:2009

suits mainly risk assessment [1], [8], [9]. Hence, this study will adopt ISO31000:2009 as a skeleton for our proposed novel framework. Given that ISO31000:2009 only answers the 'what' question, our study proposes to integrate *i** organization modelling for the purpose of addressing the 'why' and 'how' questions into consideration.

Shabnam *et al*. have utilized *i** modelling to assess risk in an organization and argue that there is a certain degree of vulnerability and critical risk within an ecosystem as a result of their association [10], [11]. The aforementioned authors also studied only the 1st tier of dependencies without considering 2nd tier actors that may also negatively affect the risk measurement. Therefore, the proposed method is set out to validate and enhance the authors' assessment through two new risk measurement parameters; these are frequency of association and time factors during the decision making/feedback process.

The paper is structured as follows. Section 2 reviews literature and will briefly describe the present risk frameworks, risk measurements and provide an introduction to the case study; Section 3 is the proposed framework; followed by Section 4 which provides results of the adoption of the new risk framework, and Section 5 includes discussions and section 6 concludes the paper.

## 2. Background

### 2.1. Risk Framework as Risk Management Guidelines

ISO 31000 is an international standard risk management guideline. It comprises of three elements: principles of risk management, the risk management framework and process. The focus is mainly on the risk management process which covers areas of risk identification, risk analysis and risk evaluation [12]. Therefore, this paper will adopt ISO31000 as a baseline to assess the risk of OSS adoption because the framework addresses risk management processes. There are five phases in ISO31000 framework which are establish context, risk identification, risk analysis, risk evaluation and risk treatment.

### 2.2. Risk in Adoption of OSS

Some examples of risk that organizations face when adopting OSS solutions are cost of adoption, compatibility and trialability and risk involving availability of support from a third party vendor [13]. Other risk impact factors also include project size, organizational impact, and complexity of project dependencies within an organization and technological compatibilities and architectures [2], [14]. Underestimating technical risk on integration is a major challenge when adopting OSS [21]. Culture can also be a risk factor in OSS adoption because if internal personnel are not open to the implementation of OSS, it will increase the time spent training users to adopt the system and consequently affecting the cost of implementation [15]. Tullio & Staples found that different types of communities and decision making styles play an important role in the effectiveness of the OSS project [16]. Fig. 1 below is a compilation risks and criteria facilitating the adoption of OSS.

### 2.3. Risk Measurement

In [10] the authors argued that associated actors have a degree of vulnerability and criticality as a result of their association with different actors. It is therefore, posited that the higher degree of vulnerabilities means that organizations must take stronger initiatives to mitigate the vulnerabilities. The authors also implied that the more critical actors are to the network, the more they will impact other actors [10]. However, authors did not consider time factors and frequency of dependencies among actors that could potentially impact the overall risk measurement. Vulnerabilities and Criticality in the current method are presented as [10]:

Vulnerability

VMorg = No of Outgoing Dependencies / No of Dependee Actors

Criticality
CMorg = No of Incoming Dependencies * No of Depender Actors

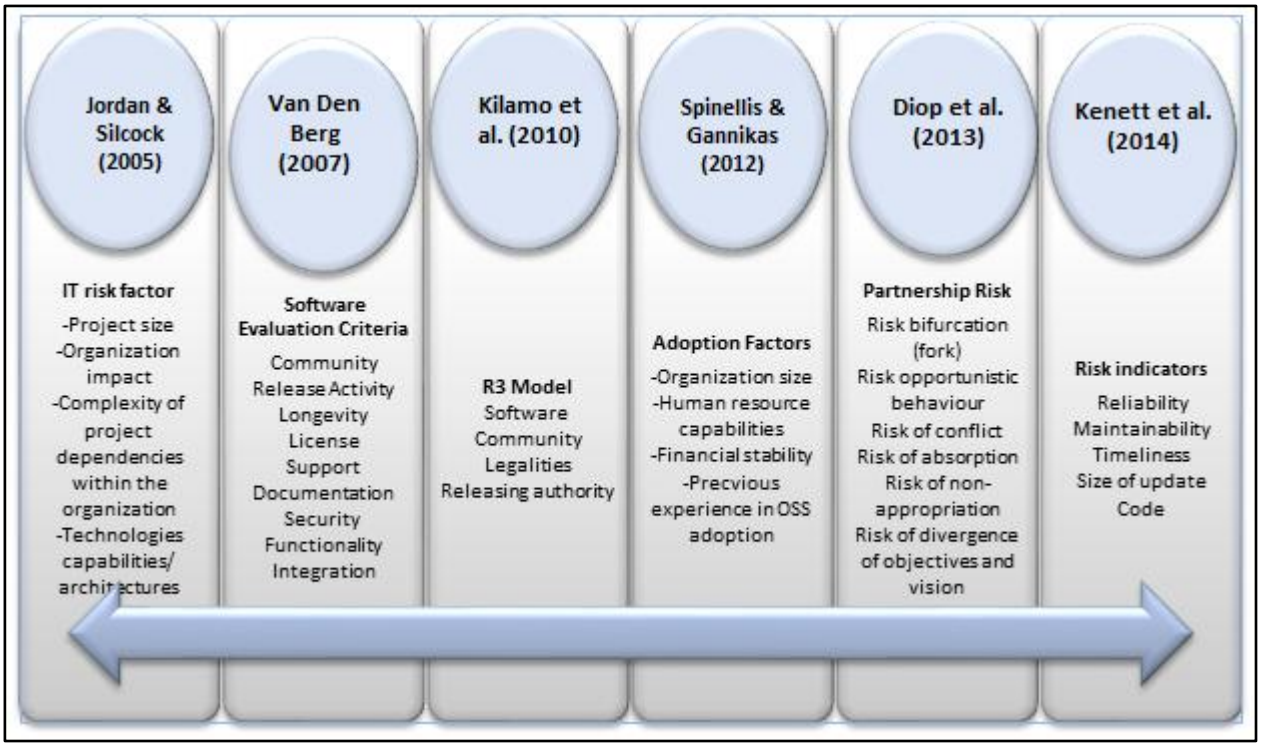



Fig. 1. OSS risk and OSS evaluation criteria.

## 2.4. OSS Evaluation Criteria

There are nine main open source evaluation criteria which are the community, release activity, longevity, license, support, documentation, security, functionality and integration (see Figure 1) [17]. Security and management of licenses seems to be one of the important factors during evaluation [18]. Additionally, a study suggests that high responsiveness in user community corresponds to effective enhancement [19]. Therefore, responsiveness can be part of the OSS evaluation criteria as it gives user a perception of project quality, activity and value.

Based on the aforementioned discussion, it is posited that organizations considering OSS solutions can utilize the Release Readiness Rating (R3) evaluation model to decide which Open Source Software is best suited for the organization [20].

OSS Ecosystem, Every OSS has its own ecosystem that is comprised of developers, the OSS community and adopters with a set of goals, task, resources and softgoals to achieve. The OSS community plays a vital role in ensuring continuity of projects in an organization [21]. Adding to that, the level of interest in the project suggests better quality and compliance [22]. There are two types of ecosystems related to OSS [21]:

- OSS community Ecosystem: This is mainly the community that maintains relationships among stakeholders.
- Adopter Ecosystem: This refers to the organization that intent to adopt OSS solutions along with other actor. This group should be aware of the range of operational risks that may impact the organization.

## 2.5. Modelling OSS Ecosystem with *i** Organizational Framework

Currently, there is substantial research on agent-oriented methodologies for requirements engineering including *i** framework, Tropos, formal Tropos, AOR, Prometheus and Gaia [23] [24]. These methodologies have similar concepts including visual modelling language and the use of agents' interaction [25] [26]. The *i** framework introduced by Eric Yu [7], models social elements of a system and can be used in the early requirements analysis stages. The Strategic Dependency (SD) model and the Strategic Rationale (SR) model are the two types of diagrams which are employed in modelling.

### 2.5.1. Strategic dependency (SD) model

The Strategic Dependency diagram represents actors' relationships. In an SD model the nodes represent the actors and the links represent the interdependency between the actors. Goal, softgoal, task and resources are the intentional elements. A dependency can be any one of the intentional elements. An SD model is a higher level of abstraction representing the actors' dependency upon each other. An SD model targets external relationships and does not disclose details of internal structure.

Fig. 2 shows SD model for this case study of an organization interested to adopt a new open source statistical software solution for data mining to track the success of marketing campaigns in order to make informed business decisions.

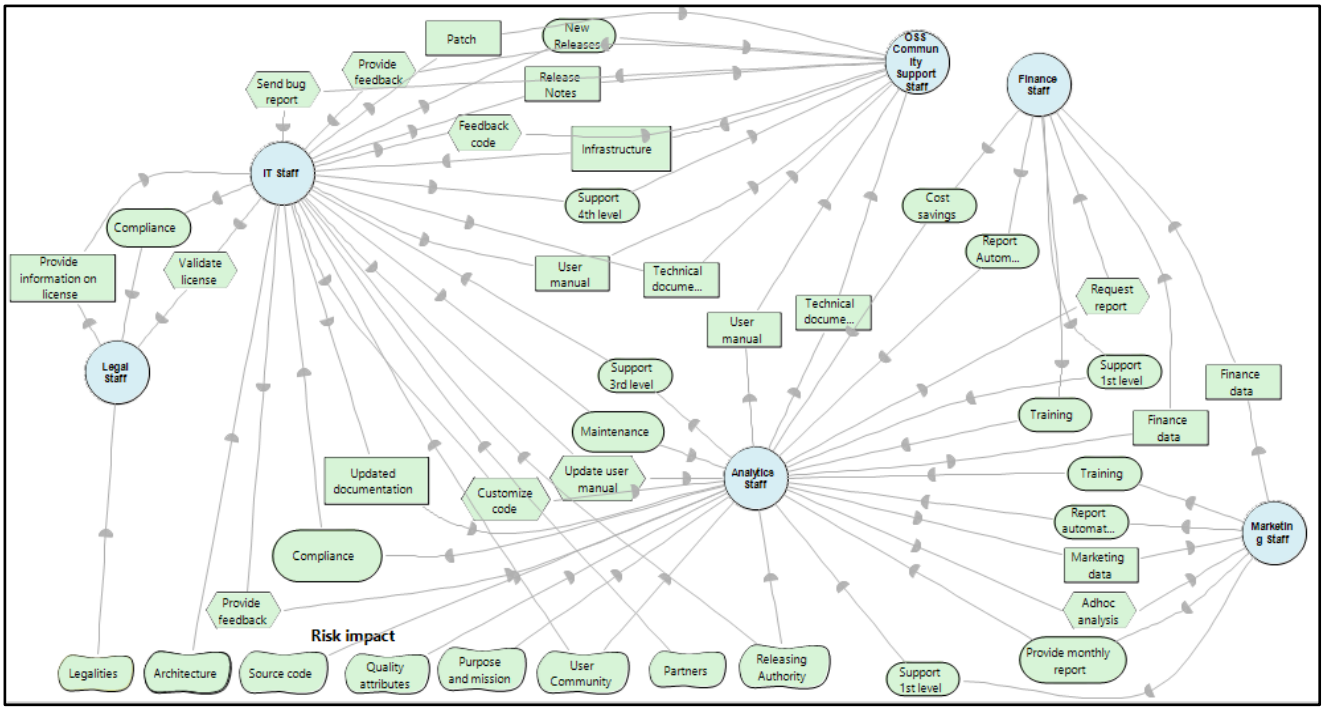



Fig. 2. Strategic dependency (SD) model.

In the strategic dependency model, there are three strategic dependency types which are depender, dependum and dependee. A depender depends on a dependee to achieve a goal, task or resource. A dependum includes goal, task, resource or softgoals.

### 2.5.2. Strategic rationale (SR) model

On the other hand, the Strategic Rationale diagram represents the internal and intentional relationship of each actor. The Strategic Rationale (SR) model is another component of *i** modelling framework that helps to identify stakeholders' interests such as goals, softgoals, task and resources that can be solved with various system configurations and environments [7]. The SR model in Figure 3 below is an extended subset version of IT Staff and OSS community association with several nodes and links that represent the

structure and rationale behind the process. The additional extension is the plotting of softgoals within the actors' environment. For example, 'OSS Community Support Staff', there is an extended softgoal namely 'Technical quality' whereby the task of running unit test helps achieve good technical quality of project.

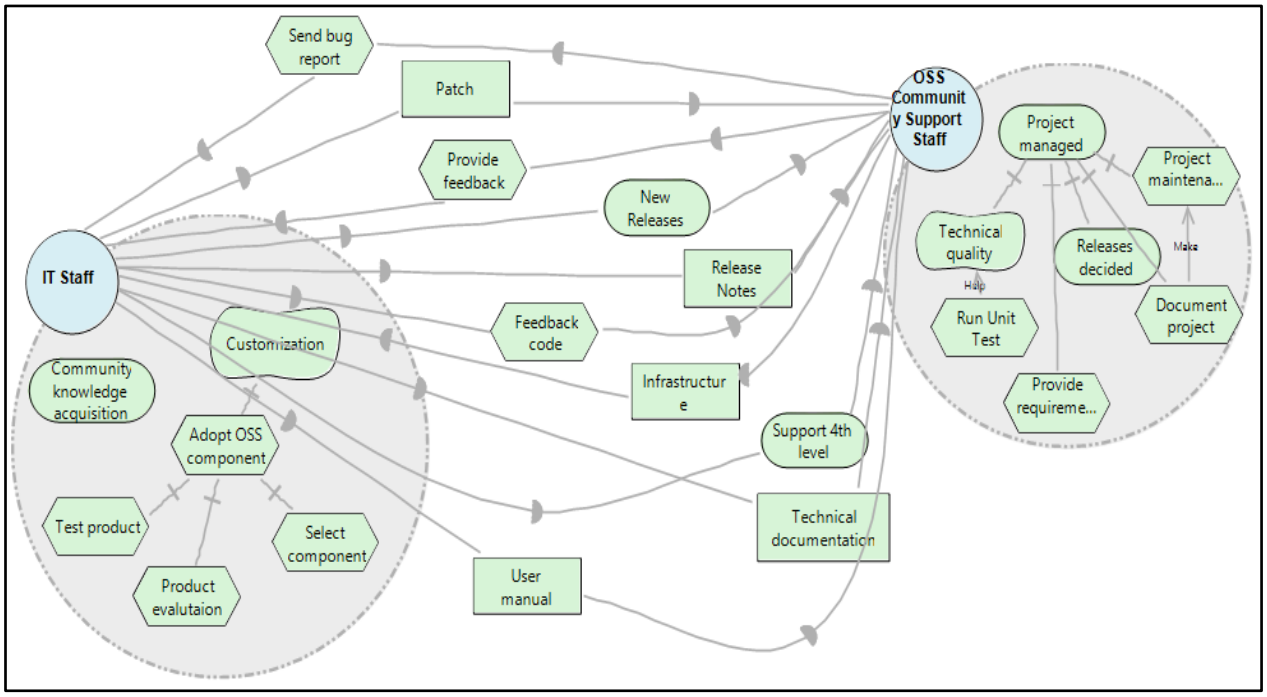



Fig. 3. Strategic rational model.

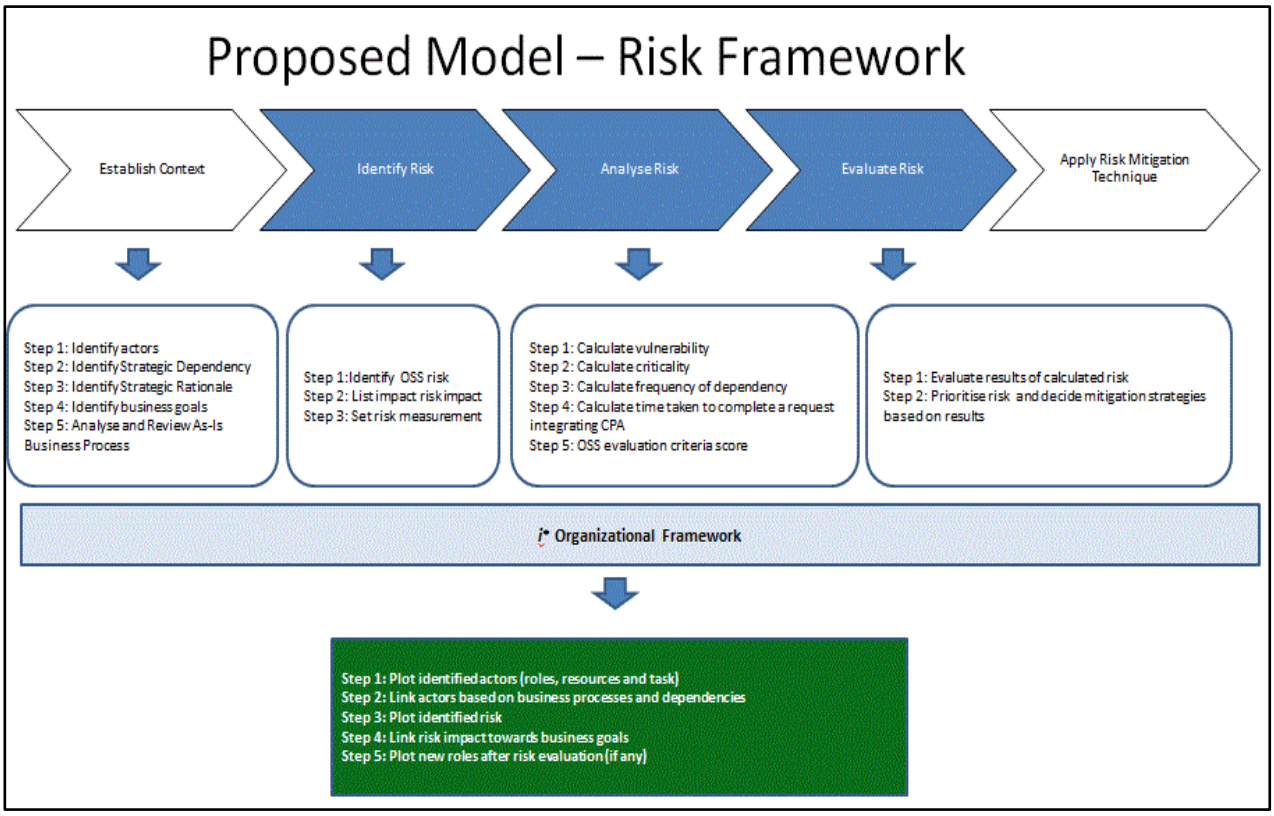



Fig. 4. Proposed risk framework.

### 2.5.3. Benefits and limitations of the *i** modelling framework

The benefit of *i** is that the model allows analysts to identify the vulnerability and opportunity. The model is used to help understand existing configurations and the proposed configurations based on stakeholders' interest. The *i** model also provides an abstract view linking risk in adoption of OSS and business goals [17] while simplifying the analysis of semantic relations [27]. Among other benefits of *i** model is to serve as a link between technical and non-technical stakeholders to understand domain knowledge [28].

The limitation of *i** model lies in delegation of work that may complicate matters. The disadvantages of *i** model are the difficulty of visualizing the level of abstraction in the ecosystem [28]; specifically mapping softgoals and delegating responsibilities. The SR Model may be too detailed, which makes it hard to understand, so the model has to be simplified to focus on only top priority goals.

## 3. The Proposed Risk Framework based on *i** Modelling (RFiM)

The proposed hybrid framework adopts the ISO31000:2009 risk framework while integrating *i** modelling to map the ecosystem to better assess risk of the OSS adoption in an organization. There are 5 main sequential phases in the proposed risk framework which are establish context, identify risk, analyse risk, evaluate risk, and apply risk mitigation technique as presented in Fig. 4.

### 3.1. Phase 1: Establish Context

This is the first phase of risk assessment. The purpose of establishing context is to understand the organizational business processes and goals. It also provides a broad overview of the internal and external environment which comprises of stakeholders, strategic goals and technical requirements that affects the implementation of OSS solutions. There are 5 steps within this phase,

Step 1: Identify actors – identify role of actors and their responsibilities in the ecosystem

Step 2: Identify Strategic Dependency – identify actors' goal, task, resource dependencies.

Step 3: Identify Strategic Rationale - identify actors' goal, task, resource and softgoal dependencies within the actors' ecosystem

Step 4: Identify business goals – overall business goals and strategic intent

Step 5: Analyse and Review As-Is Business Process – In this step, association among actors are identified to establish the background of organizational structure, business policies, guidelines and contractual relationship are established.

### 3.2. Phase 2: Identify Risk

The second phase is identifying risk. Risk identification is a process of researching, recognizing and providing a detail description of risk. There are several ways to identify risk such as primary and secondary research. Primary research includes obtaining data through questionnaire or focus group interview whereas secondary research can be derived from online information, technical manuals, journal publication or books. There are 3 steps in this phase

1) Step 1: Identify OSS risk
2) Step 2: List impact of risk
3) Step 3: Set risk measurement based on identified risk

### 3.3. Phase 3: Analyse Risk

Risk analysis is the process that involves careful consideration of the cause and effect of risk while taking likelihood of occurrence and risk estimation into account. This process includes risk quantification to act as baseline for risk evaluation and risk management. When analysing risk, the following are definition and steps to analyse the set of risk measurement among actors in the ecosystem:

4) Step 1: Calculate vulnerability [10]
5) Step 2: Calculate criticality [10]
6) Step 3: Calculate frequency of dependency

Table 1 below illustrates definition of frequency and time risk measurements.

Table 1. Definition of Frequency and Time Risk Measurement

| Name | Definition |
|---|---|
| No. of incoming dependencies | Sum of inflow of arrows of goals, task and resources for each actors |
| No. of outgoing dependencies | Sum of outflow arrows of goals, task and resources for each actors |
| No. of depender | Sum of dependers depending on the dependee |
| No. of dependee | Sum of of dependee depending on the depender |
| Frequency | Number of request/ 365 days (or 1 year period) |
| Time | Average time taken to feedback/response to report request (days) |

The calculation of frequency of dependencies is an extension of methodology illustrated in [10] where criticality and vulnerability calculation provides a validation method to the current risk measurement. Based on step 1 and 2 calculation, the dependees are further segmented by roles in order to calculate the frequency of request more precisely. The purpose of calculating this is to assist companies in deciding the cost/budget allocation, hiring or leveraging human resources with the right skills as form of risk mitigation. The formula provided calculates the number of report requested in 365 days because the case study involves an organisation that operates each day of the year.

Frequency as risk measurement

$$\text{FreqIn} = \text{No of Incoming Dependencies} \times \text{No of Depender Actors} * \text{Frequency} \quad (1)$$

$$\text{FreqOut} = \text{No. of outgoing Dependencies} \times \text{No. of Dependee Actors} * \text{Frequency} \quad (2)$$

*Please Note: Frequency = Number of request/365 days*

Hypothesis 1:

Based on the aforementioned discussion, it is hypothesised that the higher the frequency, the more critical the weightage towards achieving organizational goals.

Step 4: Calculate time taken to complete or provide feedback [integration of critical path analysis (CPM)]

Time as risk measurement

$$\textit{TimeIn} = \textit{Time} \times \textit{FreqIn} \quad (3)$$

$$\textit{TimeOut} = \textit{Time} \times \textit{FreqOut} \quad (4)$$

*Please Note: Time = days*

Hypothesis 2:

The more time taken to respond, the more critical the weight towards achieving organizational goals.

Step 5: OSS criteria evaluation

Other risk factors can include providing scores to the type of OSS that the organization intends to adopt. In this context, the R3 evaluation model is chosen to evaluate OSS products because the model has a combination and wide coverage of overall risk factors in the OSS adoption.

## 3.4. Phase 4: Evaluate Risk

Risk evaluation is deciding on treatment or control plans to mitigate risk based on outcomes of risk analysis phase. Based on the evaluated results, risk can be prioritized and risk response strategies are

decided in this phase.

Step 1: Evaluate results of calculated risk

Step 2: Prioritise risk and decide mitigation strategies based on the results.

### 3.5. Phase 5: Apply Risk Mitigation Technique

In phase 5, once the response strategy is agreed upon, organizations can choose to avoid risk by dropping the whole implementation project, mitigate risk by changing the likelihood or sequence within the business process, sharing or transferring risk (i.e. by delegating or sharing common goals to reduce dependency) or accepting the risk [8].

### 3.6. Output: *i** Organizational Modelling Framework

The output of this framework is a *i** organizational modelling framework which consists of all the factors listed in the first four phases of the proposed framework.

Step 1: Plot identified actors (roles, resources and task)

Step 2: Link actors based on business processes.

Step 3: Plot identified risk

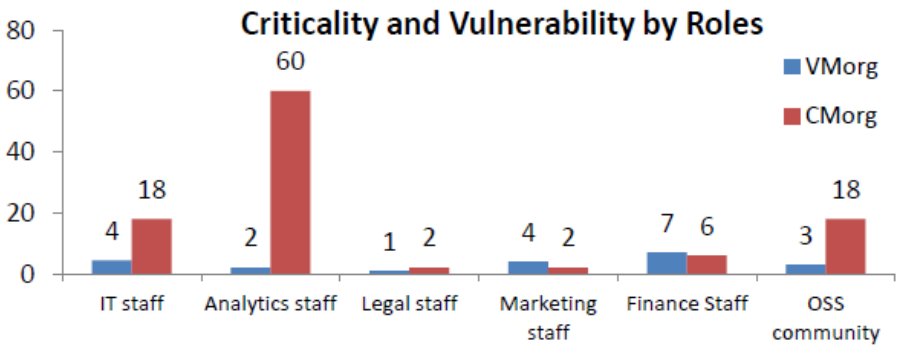
Step 4: Link risk impact towards actors.

Step 5: Plot new roles as per mitigation strategies (if any)

## 4. Results

### 4.1. Risk Measurement Based on As-Is View of the Organization Structure

Criticality and Vulnerability




Graph 1. Criticality and vulnerabilities by roles.

The bar graph (Graph 1) shows the criticality and vulnerabilities of each role in the OSS ecosystem. Analytics department Staff appears to be the most critical department in terms of incoming dependencies with a risk score of 60 whereas Finance Staff is most vulnerable with score of 7.

Frequency and Time

Table 2 illustrates the summary of FreqIn and TimeIn by roles and FreqOut and TimeOut by roles (Refer to Table 2.1 for snippet sample of FreqIn detail calculation). Based on the Incoming dependencies, the frequency of report requests coming in for Analytics Staff from other departments is very high. This further validates the criticality risk measurement which highlighted analytics Staff as the most critical dependency in the ecosystem. The Table 2 also shows that Finance Staff seems to be most dependent on analytics Staff with a FreqIn measurement of 1.15 and TimeIn 3.45 which suggests that Finance Staff response time is fairly slow compared to other roles in the ecosystem.

On the other hand, outgoing dependencies in terms of FreqOut and TimeOut of marketing Staff to analytics Staff are the highest compared to all other actors. This is due to the high frequency of reports

requested by Marketing Staff from Analytics Staff that has also led to an increase in TimeOut measurements.

Table 2. Summary of Incoming and Outgoing Frequency and Time Measurement by Roles

| Dependee Actor | No. of Incoming Dependencies | No. of Depender Actors | Depender Breakdown | FreqIn | TimeIn |
|---|---|---|---|---|---|
| IT Staff | 6 | 3 | Analytics Staff | 0.16 | 0.16 |
| | | | Legal Staff | 0.00 | 0.01 |
| | | | OSS community | 0.12 | 1.23 |
| Analytics Staff | 20 | 3 | IT Staff | 0.16 | 0.49 |
| | | | Marketing Staff | 3.68 | 11.05 |
| | | | Finance Staff | 1.15 | 3.45 |
| Legal Staff | 2 | 1 | IT Staff | 0.01 | 0.04 |
| Marketing Staff | 2 | 1 | Analytics Staff | 0.16 | 0.08 |
| Finance Staff | 3 | 2 | Analytics Staff | 0.07 | 0.07 |
| | | | Marketing Staff | 0.07 | 0.07 |
| OSS community | 9 | 2 | Analytics Staff | 0.00 | 0.01 |
| | | | IT Staff | 0.09 | 0.61 |

| Depender Actor | No. of Outgoing Dependencies | No. of Dependee Actors | Dependee Breakdown | FreqOut | TimeOut |
|---|---|---|---|---|---|
| IT Staff | 13 | 3 | Analytics Staff | 0.05 | 0.16 |
| | | | Legal Staff | 0.00 | 0.02 |
| | | | OSS community | 0.08 | 0.23 |
| Analytics Staff | 8 | 4 | IT Staff | 0.16 | 0.33 |
| | | | Marketing Staff | 0.02 | 0.07 |
| | | | Finance Staff | 0.02 | 0.07 |
| | | | OSS community | 0.01 | 0.02 |
| Legal Staff | 1 | 1 | IT Staff | 0.00 | 0.02 |
| Marketing Staff | 8 | 2 | Analytics Staff | 3.68 | 11.05 |
| | | | Finance Staff | 0.07 | 0.07 |
| Finance Staff | 7 | 1 | Analytics Staff | 1.15 | 3.45 |
| OSS community | 3 | 1 | IT Staff | 0.12 | 0.86 |

Table 2.1. FrequencyIn Detail Calculation

| Dependee Actor | No of Incoming Dependencies | No of Depender Actors | Depender Breakdown | No. of request | Over 1 year | Frequency | Breakdown of No. of Incoming Dependencies | Freq In |
|---|---|---|---|---|---|---|---|---|
| Analytics Staff | 20 | 3 | IT Staff | 10 | 365 | 0.03 | 6 | 0.16 |
| | | | Marketing Staff | 192 | 365 | 0.53 | 7 | 3.68 |
| | | | Finance Staff | 60 | 365 | 0.16 | 7 | 1.15 |

## 4.2. OSS Evaluation Criteria

An example of suggested compilation of statistical open source software evaluation criteria utilizing R3 model is provided in Table 3.

Table 3. Open Source Statistical Software Description and Features

| Statistical software | | | | | |
|---|---|---|---|---|---|
| Software name | A | B | C | D | E |
| Active since | 2001 | 1997 | 1997 | 1996 | 2006 |
| Still Active | Yes | Yes | Yes | Yes | Yes |
| OS | Windows 32,64 MacOS10.8, Linux | Windows, OS X, Linux | UNIX, Windows and MacOS | LINUX, Windows and MacOS | Unknown |
| Language | Java7 or Java8 | Java | C, C++ and Fortran | C++, Python | Java |
| License | APGL-3.0 | GNU GPLv3 | GNU | GPL | GNU GPL |
| Sourcecode available | Yes | Yes | Yes | Yes | Yes |
| Free | Yes | Yes | Yes | Yes | Yes |
| Paid | Yes | No | No | No | No |
| List of business partners | 40 | 0 | 0 | 0 | 14 |
| Technical partners | 9 | 0 | 0 | 0 | 18 |
| Customizable extensions | Yes | Yes | Yes | Yes | Yes |

Based on the Table 4, the top 2 open source software that meet the organizational needs are software C and software A with 0.59 and 0.58 each compared to other open source software solution. Software C appears to be on top of the list mainly because of the architecture and quality attributes category score of 0.3 and 0.21 respectively and the extensive activeness of user community 0.33.

Based on the OSS evaluation criteria results, the top 2 software with highest rating will be chosen and re-evaluated against the overall internal ecosystem. Fig. 7 below illustrates the different risk impact effecting different actors in the ecosystem; Further highlighting the roles and responsibility of each actors towards the risk factors.

Table 4. OSS Evaluation Criteria for OSS Statistical Software

| Dimension | Category | Software name | | | | |
|---|---|---|---|---|---|---|
| | | A | B | C | D | E |
| **Overall** | | **0.58** | **0.48** | **0.59** | **0.54** | **0.55** |
| Software | | 0.16 | 0.19 | 0.18 | 0.15 | 0.13 |
| | Source code | 0.24 | 0.3 | 0.21 | 0.21 | 0.21 |
| | Architecture | 0.23 | 0.26 | 0.3 | 0.19 | 0.21 |
| | Quality Attributes | 0.15 | 0.18 | 0.21 | 0.21 | 0.1 |
| Community | | 0.17 | 0.04 | 0.13 | 0.11 | 0.14 |
| | Purpose and mission | 0.24 | 0.09 | 0.18 | 0.2 | 0.24 |
| | User community | 0.28 | 0.16 | 0.33 | 0.22 | 0.19 |
| Legalities | Partners | 0.17 | 0 | 0 | 0 | 0.13 |
| | | 0.06 | 0.08 | 0.1 | 0.1 | 0.1 |
| | Copyright | 0 | 0 | 0 | 0 | 0 |
| | Licensing | 0.17 | 0.25 | 0.33 | 0.33 | 0.33 |
| | Branding | 0.08 | 0.08 | 0.08 | 0.08 | 0.08 |
| Releasing authority | | 0.19 | 0.18 | 0.18 | 0.18 | 0.18 |
| | Mindset, culture and motivation | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 |
| | Process, organization and support | 0.27 | 0.2 | 0.2 | 0.2 | 0.2 |
| | Infrastructure | 0.33 | 0.33 | 0.33 | 0.33 | 0.33 |

*Note: software evaluation is subjective and based on suitability to organization and does not in any way reflect the overall functionality of the software

## 4.3. Application of Risk Mitigation Strategies on Strategic Dependency Model (To-Be View)

Based on the risk evaluation, the following section proposes risk mitigation strategies that will lower the criticality of analytics Staff by delegating dependencies to new roles based on shared common goals, task, resources and softgoals and removing duplicated resources as depicted Figure 5 with 'x' symbol next to task, resource and goals.

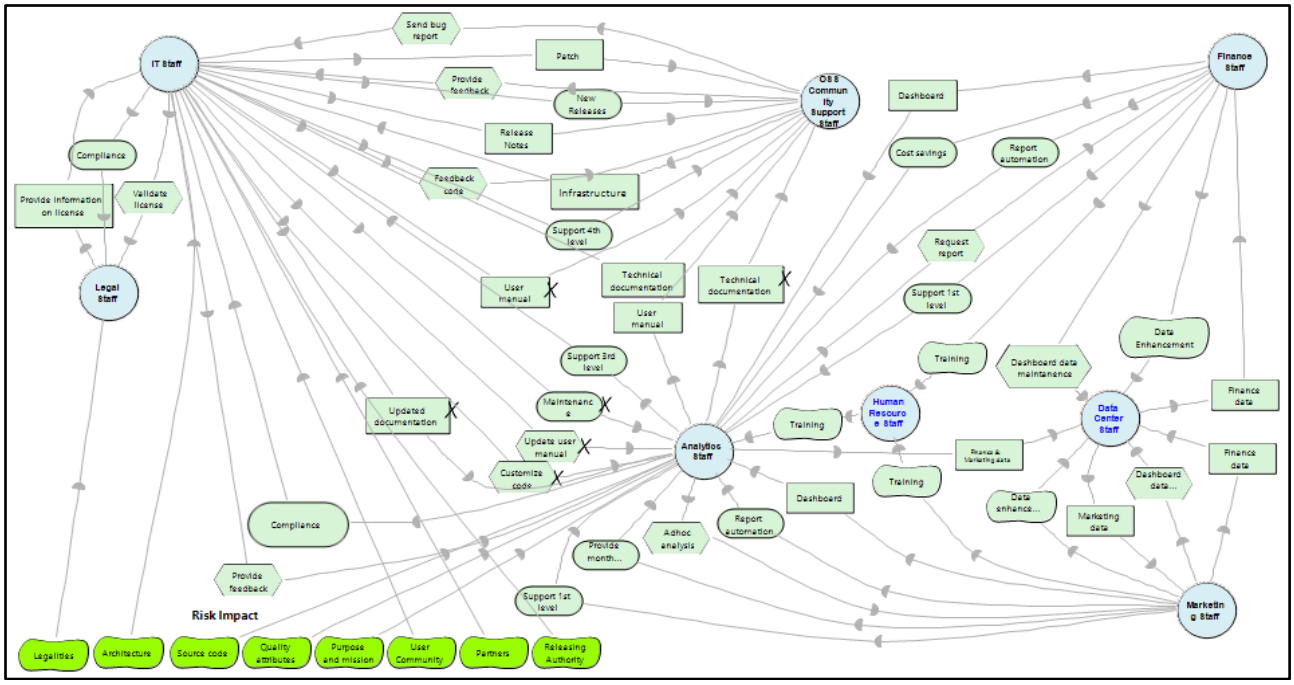


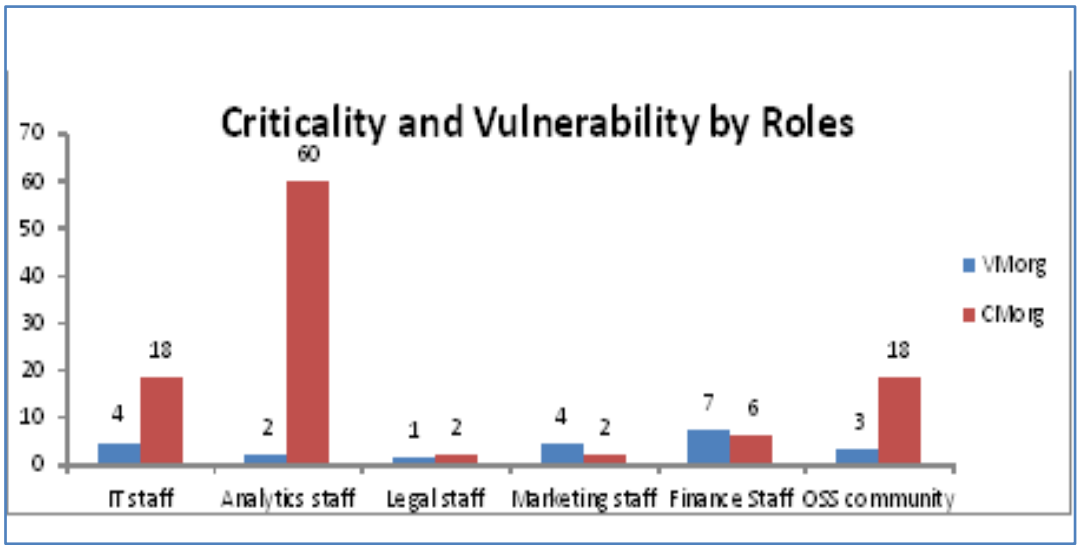

Fig. 5. Strategic dependency model with risk impact and proposed roles.

The two new proposed roles are HR Staff and Data Center Staff. HR Staff is in charge of handling arrangement of training; this helps to reduce dependency on Analytics Staff. Data Center Staff is in charge of data governance.

Duplicated resources are assigned according to the roles relevance; Analytics Staff no longer needs to depend on OSS community to obtain technical documentation since they do not need to look at architectural aspect of the OSS compared to IT Staff. Currently, IT Staff depend on Analytics Staff to customize source code, update user manual and documentation for back up purposes. This will no longer be needed if the tasks of backing up these documents are handed over to the Analytics Staff.

## 4.4. Risk Measurement based on To-Be View of Organizational Structure

Criticality and Vulnerability




Graph 2. Graph illustrating criticality and vulnerability by roles.

The criticality measure has now decreased from 60 to 40. Vulnerabilities measures are now spread out across all actors as opposed to the as-is model whereby finance Staff are most vulnerable.

Frequency and Time

Table 5. Summary Table of Mitigated Risk FreqIn, TimeIn, FreqOut and TimeOut Measurement

| Dependee Actor | No of Incoming Dependencies | No of Depender Actors | Depender Breakdown | FreqIn | TimeIn | Depender Actor | No of Outgoing Dependencies | No of Dependee Actors | Dependee Breakdown | FreqOut | TimeOut |
|---|---|---|---|---|---|---|---|---|---|---|---|
| IT Staff | 6 | 3 | Analytics Staff | 0.16 | 0.16 | IT Staff | 8 | 3 | Analytics Staff | 0.01 | 0.03 |
| | | | Legal Staff | 0.00 | 0.01 | | | | Legal Staff | 0.00 | 0.00 |
| | | | OSS community | 0.12 | 1.23 | | | | OSS community | 0.07 | 0.20 |
| Analytics Staff | 10 | 4 | IT Staff | 0.03 | 0.08 | Analytics Staff | 6 | 5 | IT Staff | 0.16 | 0.16 |
| | | | Marketing Staff | 1.03 | 3.09 | | | | Marketing Staff | 0.01 | 0.03 |
| | | | Finance Staff | 0.33 | 0.99 | | | | Finance Staff | 0.01 | 0.03 |
| | | | HR Staff | 0.01 | 0.08 | | | | OSS community | 0.00 | 0.00 |
| Legal Staff | 2 | 1 | IT Staff | 0.01 | 0.00 | | | | Data Center Staff | 1.00 | 1.00 |
| Marketing Staff | 2 | 2 | Analytics Staff | 0.08 | 0.25 | Legal Staff | 1 | 1 | IT Staff | 0.00 | 0.01 |
| | | | Data Center Staff | 0.03 | 0.03 | Marketing Staff | 8 | 3 | Analytics Staff | 1.01 | 3.02 |
| Finance Staff | 2 | 2 | Analytics Staff | 0.03 | 0.10 | | | | HR Staff | 0.01 | 0.08 |
| | | | Data Center Staff | 0.03 | 0.03 | | | | Data center Staff | 0.10 | 0.10 |
| OSS community | 7 | 2 | Analytics Staff | 0.00 | 0.00 | Finance Staff | 7 | 3 | Analytics Staff | 0.33 | 0.99 |
| | | | IT Staff | 0.07 | 0.20 | | | | HR Staff | 0.01 | 0.08 |
| HR Staff | 2 | 2 | Marketing Staff | 0.01 | 0.08 | | | | Data Center Staff | 0.16 | 0.16 |
| | | | Finance Staff | 0.01 | 0.08 | OSS community | 3 | 1 | IT Staff | 0.12 | 1.23 |
| Data Center Staff | 6 | 3 | Marketing Staff | 0.81 | 0.81 | HR Staff | 1 | 1 | Analytics Staff | 0.01 | 0.08 |
| | | | Finance Staff | 0.16 | 0.16 | Data Center Staff | 2 | 2 | Marketing Staff | 0.03 | 0.03 |
| | | | Analytics Staff | 1.00 | 1.00 | | | | Finance Staff | 0.03 | 0.03 |

Based on Table 5, initial FreqIn of Marketing Staff to analytics Staff was 3.68. This has now decreased to 1.03 because 50% of the request has been channelled to data center Staff to reduce criticality. Also, TimeIn

has reduced from 11.05 to 3.09. However, a new criticality has emerged from this proposal whereby Data Center Staff are now in a critical role supplying data to Analytics Staff.

## 5. Discussion

Based on the findings in the previous sections, Analytics Staff appears to have the most critical role in terms of the consequence of impact whereas Finance Staff is most vulnerable; this suggests a stronger implementation of risk mitigation for an alternative effective organizational structure.

The findings are further validated by the calculation of frequency and responsiveness among actors whereby Analytics Department Staff criticality has a strong relationship with marketing Staff based on the number of reports requested by the Marketing Staff. These findings are particularly useful when considering resources allocation and design implementation to manage services and support, given that new systems will often impact the operational process in the organization.

An interesting observation, it takes the Analytics Staff fairly long time to provide feedback to Marketing Staff (risk TimeOut=11.05, refer to Table 2) in part due to the frequency of requests made by marketing Staff. This implies that analytics department may want to consider delegating and training staff from other departments to better utilize the software to reduce request time and frequency of requests.

Next, based on Table 5 the OSS evaluation criteria (R3) model, the top two OSS software most relevant to the chosen organizational goals are software C and A. The main difference between software C and software A is the latter has a partnership score associated with it. Since the organization is not entirely prepared to operate in a transparent manner due to the nature of its business, partnership will not be a priority. Since the organization has existing resources and prior experience in utilizing software C and possesses C++ skills, the organization has decided to adopt software C into the organization.

The next step of this framework is to prioritise risk impact. Naturally, 'Software' dimension (refer to Table 4) will be the first priority because if the software cannot be installed and the architecture does not support the current system, the IT department will have to decide whether to upgrade its platform to meet the software requirements or to abandon the project. 'Releasing authority' dimension is second on the priority list since its adoption needs to be supported by the relevant stakeholders and top management. Third priority is 'Community' dimension because as mentioned, partnership will not be taken into consideration in this context because the organization is not willing to form partnership and therefore third dimension is irrelevant. Finally, the 'Legalities' dimension score is similar across all software and software licenses evaluated in this case study are similarly compliant to open source software licenses therefore this is rated as the lowest priority.

Finally, once the listed measurements are analysed and evaluated, an application of risk mitigation techniques are proposed in Fig. 5. The two new proposed roles are HR and Data Center Staff. HR Staff is in charge of scheduling training for the new software; this helps to reduce dependency on Analytics Staff on training coordination. Data Center Staff is in charge of data governance. The findings show that the Introduction of these additional methods has proven that criticality and vulnerability has been mitigated.

## 6. Conclusion

In conclusion, the aim of this paper is to propose a risk framework for OSS adoption. The benefits of the proposed measurement of frequency and time are to enhance and validate the current methods [10]. It also considers 2nd tier actor dependencies to improve the accuracy of measurement and provide a broader perspective of the organization's network dependencies. This approach can be used to mitigate risk by proposing that other departments absorb some goals or task from the critical actor and merge duplicated tasks to reduce dependencies. The proposed framework also integrates impact of risk towards different

actors so organizations are aware of their roles in mitigating these risks. This paper also provided suggestions on risk approaches and mitigation strategies. Future plans include proposing a framework in greater details and possibly drilling into business process management (BPM) to address organizations in various domains. Further studies on alternative risk mitigation strategies are required, combining shared goals and possibly exploring business process management. Subsequently, implementing the framework in an organization and comparing effectiveness of this methodology.

## References


[1] Agrawal, M., Campoe, A., & Pierce, E. (2014) *Information Security and IT Risks Management*. Wiley. p.1.

[2] Jordan, E., & Silcock, L. (2005). *Beating IT Risks*. West Sussex, England: J. Wiley.

[3] Van Loon, A., & Toshkov, D. (2015). Adopting open source software in public administration: The importance of boundary spanners and political commitment. *Government Information Quarterly*, *32(2)*, 207-215.

[4] Affleck, A., Krishna, A., & Achuthan N. (2013). Optimal selection of operationalizations for non-functional requirements. *Proceedings of the 9th Asia-Pacific Conference on Conceptual Modelling* (pp. 69-78).

[5] Affleck, A., & Krishna, A. (2012). Supporting quantitative reasoning of non-functional requirements: A process-oriented approach. *Proceedings of the 2012 International Conference on Software and Systems Process* (pp. 88-92).

[6] Burgess, C., Krishna, A., & Jiang, L. (2009). Towards optimising non-functional requirements. *Proceedings of the 9th International Conference on Quality Software* (pp. 269-277).

[7] Yu, E. (1997). Towards modelling and reasoning support for early-phase requirements engineering. *IEEE*, 226 – 235.

[8] Harrast, S., & Weirich, T. (2009). New IT risk framework. *J. Corp. Acct. Fin. Journal of Corporate Accounting & Finance*, 49-54.

[9] Bhuiyan, M., & Krishna, A. (2010). Business modeling with the support of multiple notations in requirements engineering. *Proceedings of the 14th Pacific Asia Conference on Information Systems*.

[10] Shabnam, L., Haque, F., Bhuiyan, M., & Krishna, A. (2014). Risk measure propagation through organisational network.

[11] Zahidul Islam, M., Bhuiyan, M., Krishna, A., & Ghose, A. (2009). An integrated approach to managing business process risk using rich organizational models.

[12] Ernawati, T. & Suhardi, Nugroho, D.R. (2012). IT risk management framework based on ISO 31000:2009. System Engineering and Technology (ICSET), 2012 International Conference on , vol., no., pp.1,8, 11-12 Sept. 2012.

[13] Diop, B., Pascot, D., Mbibi, S., & Banag, L. (2013). Risk factors of the partner relationship between open source ERP editors and IT services companies. *IBR, 6(9)*.

[14] López, L., Costal, D., Ayala, C. P., Franch, X., Annosi, M. C., Glott, R., & Haaland, K. (2015). Adoption of OSS components: A goal-oriented approach. *Data & Knowledge Engineering*, *99*, 17-38.

[15] Qu, W., Yang, Z., & Wang, Z. (2011). Multi-level framework of open source software adoption. *Journal of Business Research*, 997-1003.

[16] Tullio, D., & Staples, S. (2013). The governance and control of open source software projects. *Journal of Management Information Systems*, *30(3)*, 49–80.

[17] Van, D. B. K. (2007). Open source software evaluation. *Handbook of Research on Open Source Software Technological, Economic, and Social Perspectives*, 197-209.

[18] Erlich, Z., & Aviv, R. (2007). Open source software: Strength and weaknesses. *Handbook of Research on*

*Open Source Software Technological, Economic, and Social Perspectives*, 184-196.

[19] Ghapanchi, A. H., & Aurum, A. (2011). An evaluation criterion for open source software projects: enhancement process effectiveness. *International Journal of Information Systems and Change Management*, *5(3)*, 193-208.

[20] Kilamo, T. (2010). Evaluating the readiness of priopriety software for open source development. *Open Source Software*, New Horizons. Springer.

[21] Kenett, R., Franch, X., Susi, A., & Galanis, N. (2014). Adoption of free libre open source software (FLOSS): A risks management perspective. *Proceedings of the 2014 IEEE 38th Annual Computer Software and Applications Conference*.

[22] Van, D. B. K. (2007). Open source software evaluation. *Handbook of Research on Open Source Software Technological, Economic, and Social Perspectives*, 197-209.

[23] Bhuiyan, M. (2012). Managing process design in a dynamic organisational context. University of Wollonggong Australia. PhD Thesis.

[24] Fortuito, A., Bhuiyan, M., Haque, F., Shabnam, L., Krishna, A., & Withana., C. (2015). Citizen's charter driven service area improvement. *Proceedings of the 22nd Asia-Pacific Software Engineering Conference* (pp. 401-408).

[25] Miller, T., Pedell, S., Lopez-Lorca, A. A., Mendoza, A., Sterling, L., & Keirnan, A. (2015). Emotion-led modelling for people-oriented requirements engineering: The case study of emergency systems. *Journal of Systems and Software*, *105*, 54-71.

[26] Duran - Faundez, C., Ramos, M., & Rodriguez, P. (2015). Applying Gaia and AUML for the development of multiagent — Based control software for flexible manufacturing systems: Addressing methodological and implementation issues. *Software: Practice and Experience*, *45(12)*, 1719-1737.

[27] Ruiz, M., Costal, D., España, S., Franch, X., & Pastor, Ó. (2015). GoBIS: An integrated framework to analyse the goal and businessprocess perspectives in information systems. *Information Systems*, *53*, 330-345.

[28] Norta, A., Mahunnah, M. Tenso, & Taveter, K. (2014). An agent-oriented method for designing large socio-technical service-ecosystems. *Proceedings of the IEEE 10th World Congress on Services*.

[29] Costal, D, Gross, D. *et al*., (2014). Quantifying the impact of OSS adoption Risks with the help of *i** model. *GESSI Research Group*.

**Jo Lyn Teh** is a recipient of dean's list for academic excellence in master of IT from the Department of Computing at Charles Sturt University, Australia. She also has received a dean's award for postgraduate diploma of IT from Macquarie University, Australia and a BA (Hons) international business management degree from Northumbria University, UK. She has vast experience as an Intelligence Analyst specialising in data analytics. Her research interests include risk management, data mining and machine learning.

**Moshiur Bhuiyan** is an experienced IT management consultant, who possesses extensive expertise in management consulting, business analysis, BPM, change management and enterprise architecture. He has significant passion in research and teaching. His research areas include but are not limited to business process discovery & modelling, process rules and policy integration, process execution, process reengineering and optimization, process lifecycle management, change management, software requirement engineering, cloud computing, ICT governance & architecture. He has published his works in reputed international conferences and journals. He has served as program committee member and reviewer in several conferences and workshops. He is also the founder member of a technology entrepreneurship

company named Enterprise Cloud Systems (www.ecloudsys.com) which develops innovative cloud applications.

**P. W. C. Prasad** is an adjunct associate professor with the School of Computing and Mathematics at Charles Sturt University, Australia. Prior to this, he was a lecturer at the United Arab Emirates University in UAE, Multimedia University in Malaysia and also the Informatics Institute of Technology (IIT), Sri Lanka. He gained his undergraduate and postgraduate degrees from St Petersburg State Electrotechnical University in the early 90s and completed his PhD studies at the Multimedia University in Malaysia. He is an active researcher in the areas of computer architecture, digital systems, modelling and simulation. He has published more than 100 research articles in computing and engineering journals and conferences proceedings. He has co-authored two books entitled 'Digital Systems Fundamentals' and 'Computer Systems Organization and Architecture' published by Prentice Hall. He is a senior member of the IEEE Computer Society.

**Aneesh Krishna** is currently senior lecturer of software engineering with the Department of Computing, Curtin University, Australia. He holds a PhD in computer science from the University of Wollongong, Australia, an M.Sc (Engg.) in electronics engineering from Aligarh Muslim University, India and a B.E. degree in electronics engineering from Bangalore University, India. He was a lecturer in software engineering at the School of Computer Science and Software Engineering, University of Wollongong, Australia (from February 2006 - June 2009). His research interests include software engineering, requirements engineering, conceptual modelling, agent systems, formal methods and service-oriented computing. His research is (or has been) funded by the Australian Research Council and various Australian government agencies as well as companies such as Woodside Energy, Amristar Solutions, Andrew Corporation, NSW State Emergency Service, Western Australia Dementia Study Centre and Autism West. He serves as assessor (Ozreader) for the Australian Research Council. He has been on the organising committee, served as invited technical program committee member of many conferences and workshop in the areas related to his research.