Global Journal of Computer Science and Technology: C
Software & Data Engineering
Volume 14 Issue 2 Version 1.0 Year 2014
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Identification of Critical Risk Phase in Commercial-off-the-Shelf Software (CBSD) using FMEA Approach

By Palak Arora & Harshpreet Singh

Lovely Professional University, India

Abstract- COTS based development is becoming a popular software development approach for building large organizational software using existing developed components. COTS based approach provides pre-developed components either as in house or commercial off the shelf components, which reduces effort and cost for developing the software. There are potential challenges, risks and complexities in using COTS components. This paper provides an analysis of risks and challenges faced during developing software using CBSD approach. The risks under various phases are identified, categorized and prioritized the risks in various phases of CBSD and provide the mitigation strategy to manage the risks.

Keywords: *CBSD, risks in CBSD, risk mitigation.*

GJCST-C Classification: *K.6.3*

Strictly as per the compliance and regulations of:

© 2014. Palak Arora & Harshpreet Singh. This is a research/review paper, distributed under the terms of the Creative Commons Attribution-Noncommercial 3.0 Unported License http://creativecommons.org/licenses/by-nc/3.0/), permitting all non-commercial use, distribution, and reproduction inany medium, provided the original work is properly cited.

Identification of Critical Risk Phase in Commercial-off-the-Shelf Software (CBSD) using FMEA Approach

Palak Arora [α] & Harshpreet Singh [σ]

Abstract- COTS based development is becoming a popular software development approach for building large organizational software using existing developed components. COTS based approach provides pre-developed components either as in house or commercial off the shelf components, which reduces effort and cost for developing the software. There are potential challenges, risks and complexities in using COTS components. This paper provides an analysis of risks and challenges faced during developing software using CBSD approach. The risks under various phases are identified, categorized and prioritized the risks in various phases of CBSD and provide the mitigation strategy to manage the risks.

General Term: commercial-off-the-shelf software development (CBSD).

Keywords: CBSD, risks in CBSD, risk mitigation.

I. Introduction

COTS-based software development aims in building the software using the existing developed components. The components can be developed in house for usage among vast projects of similar requirements. The components can also be purchased from the market as the components are also developed as small software's which intend to provide the basic functionality required for large projects.

Various components are also available in the repositories with their functionalities and Quality attributes. A target application/ software are developed by selecting the appropriate components from the component repository & then integrating the components into a target system as in Figure 1 below.

At present time, more than 60% of software are developed using component approach due to its enormous features such as:

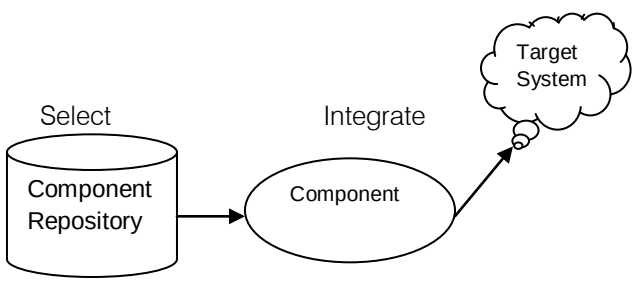

Figure 1 : Component-based Software Development

- Rapidly development.
- Accessed Immediately.
- Reduced Complexity.
- Increases efficiency of products.
- Reduced implementation, operating and maintenance cost.
- Reduced amount of time to deliver products in the market, budget and schedule saving, more than half of the software developers used component based approach. This approach has reduced the software crisis at great extent [6].

The main rationale of CBSD approach is to develop big system by integrating the pre-built components which decrease the progress time & costs. There are five main phases: Identification, Evaluation, Selection, Integration and Development of component to develop software using CBSD approach as mentioned in Figure 2 below.

II. Review of Literature

To provide a reliable and effective software product in the market, software industry influenced by COTS development approach. In software applications CBSD is the only need to be written once and re-used multiple times than being re-written every time when a new application is developed. CBSD approach overlaps the traditional software engineering approach where existing technologies were failed to deliver project on-time and on-budget. The main reasons of these failures are: Testing -

Author α: Student, School of CSE, Lovely Professional University Phagwara, Punjab. e-mail: palakarora718@gmail.com

Author σ: Assistant Professor, School of CSE Lovely Professional University Phagwara, Punjab. e-mail: harshpreet.17478@lpu.co.in

© 2014 Global Journals Inc. (US)

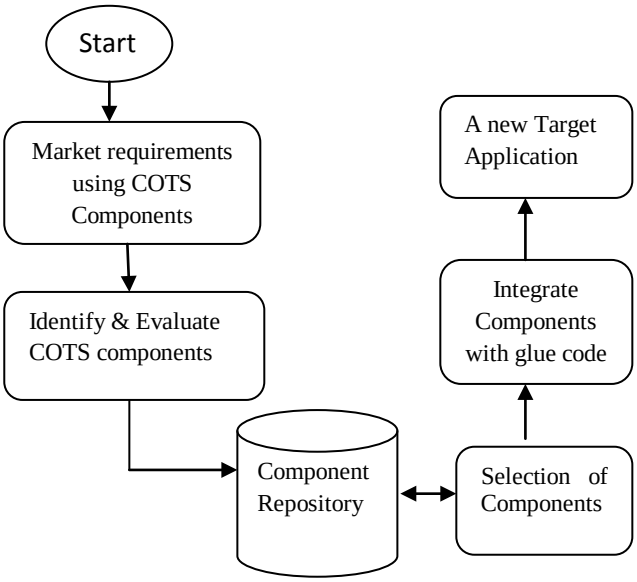

Figure 2 : COTS Development Life cycle

-efforts are not properly estimated; Team's skill is under/over estimated. However, the use of CBSD approach provides a lot of benefits, but still there are several challenges, risks, uncertainties related to this approach [6]. As the name suggested, CBSD approach means use of existing components, we are depending upon someone else (lack of trust). The main reasons of these problems are due to these factors:

- Wrong selection of components,
- Black box nature (non-availability of code) of COTS Components,
- Lack of knowledge, guidance etc.
- Unknown quality of COTS Products.

Many times, some risks are not identified in one phase and it overlaps to the second phase so in this way, it influences the whole software and fails to the organization's business. So, there is a need of proper Risk Management for using this CBSD approach from the starting phase. Failure Modes and Effects Analysis (FMEA) is a systematic method for evaluating a process to identify where risk is and how it might fail and to assess the relative impact of different failures [7]. With the help of FMEA approach, this paper provides risk management strategy for Commercial-off- The- Shelf Software development.

III. Problem Definition & Solution

In developing software using CBSD approach there is an uncertainty that there can be variations between the planned development approach and the actual software developed. A risk could cause an organization to fail to meet its approach and objectives. The main steps of this paper are as in Figure 3 below:

Figure 3 : Step-wise Problem definition

a) Challenges Faced during COTS-based Software Development life cycle

The use of commercial-off-The Shelf software Development has become an important need for developing software as they offer reduce development time and effort. Similarly there are many challenges faced such as the quality attribute of selected components may cause deviation in the quality of final product, also the cost and effort involved in integrating component during the design process may cause the product design to deviate from the actual requirement There are many challenges that start during COTS development (Identification, Selection, Evaluation, Integration, and Development) summarised as below [1]: -

1. Companies have very limited access to product's internal design and the description of commercial package is in improper format.
2. When evaluating the COTS components, customers have very few chances to verify in advance whether the desired requirements will be met in the future.
3. Selection of COTS becomes major challenge faced by requirement engineers to match the requirements with available COTS.
4. Selection of components becomes major challenge faced by requirement engineers to match the requirements with available COTS.
5. The level of quality is unknown. The COTS products will have defects, no one know where and how many will be.
6. Documentation related to component may be of inadequate quality to be used.
7. Selection of COTS components is often based on subjective judgement, so there are no additional specifications provided by vendors for COTS component's internal architecture and description.

b) Risk Management Planning

Risk management planning is a continuous process for identifying and measuring the risks continuously identifying and measuring the risks; developing mitigation options; selecting, planning, and implementing appropriate risk mitigations. It also

© 2014 Global Journals Inc. (US)

involves tracking the implementation to ensure successful risk reduction.

i. *Identification of risks during CBSD Lifecycle*

Using the COTS development approach the components are purchased from the third party vendor due to which the development of the software depends upon the customer support services provided by the vendors. So, there are several chances of arising risks on each phase of CBSD as in figure 4. The risks in CBSD life cycle are due to the factors such as the black box nature of COTS components, lack of interoperability standards, the disparity between the user & suppliers, incomplete format of requirement documentation etc. The classification of risks based on various phases is briefly defined as in [6].

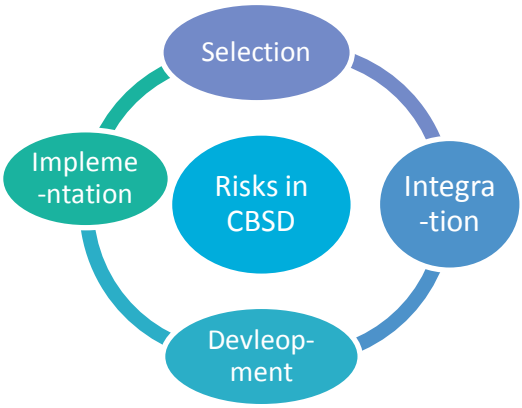

Figure 4 : Risks in CBSD life cycle

1. *Risks in COTS Selection Phase*

Risk during this phase is associated with the problems of evaluating and selecting off-the-shelf software for use in the system. The risks in this phase are due to some parameters as unavailability of source code, inflexibility of COTS components, lack of requirement document, architecture mismatches etc.

2. *Risks in COTS Integration Phase*

These risks are associated with problems of integrating systems from the existing COTS components. These risks can occur while composing of COTS components due to the lack of interoperability standards, occurrence of incompatible format among different COTS components, incomplete format of requirements etc.

3. *Risks during COTS Development*

The risks in this phase are arises when we develop the architecture from the selected COTS components. The risk arises due to the problem of using an inappropriate development process.

4. *Risks during COTS Implementation Phase*

The risks in this phase are during when we implement the final systems after selecting the appropriate components. These risks are due to the unclear design assumptions, performance factors, and security factors.

ii. *Classification of Risks during Phase-wise of CBSD*

There are three types of areas where the identified risk arises mostly:

- Functional/ Operational Requirements - The risks are which arises with the functionality and performance of the system as perceived by its operators.
- Procedural approach - The risks that are related with the technical characteristics of COTS products.
- Production strategy - Those risks which are related with the vendor of the COTS product.

1. *Risks Involving in Functional/ Operational Requirements*

Table 1 : Risks Involving in Functional/ Operational Requirements

For this Potential types of risks	Risks
Availability Risks	In the case of COTS components, it is difficult to predict that the available COTS component will meet the functional requirements, so the estimated development cost and schedule are highly uncertain
Functionality & Performance	In COTS components, the actual functionality and performance of a COTS product are not as publicized so the system may not meet its requirements.
Requirements Gap	COTS component does not match the current operational requirements or procedures.
Security and Safety Issues	It may not be possible to certify that the product meets requirements because the COTS product must be tested as a black box without its implementation

2. *Risk involving in Procedural Approach*

Table 2 : Risk involving in Procedural Approach

For the possible kinds of Risks are:	Risks
Conformance to Commercial Standards	COTS components do not conform to commercial standards so interoperability with other selected COTS products may be difficult & costly.
Integration	Contractor does not have the technical

© 2014 Global Journals Inc. (US)

Contractor's Capability	Knowledge & experience to deliver a COTS-based system so the system may not meet requirements..
Quality Requirements	COTS software components do not meet quality requirements (e.g., reliability, performance, usability).
Adaptability Risks	COTS products does not fully support initial and evolving requirements and do not have built-in flexibility.
Portability Risks	It is not necessary that COTS package will always supportable across a variety of hardware and operating system plat-forms, then hardware platform choices over a program lifecycle may be limited.
Evolution Risks	Sometimes, COTS components, hardware upgrades or replacements are not compatible with current COTS software products so COTS software components may have to be replaced at the same time.
Source code	If there is no access to source code, then it may be difficult to trace integra-tion and testing problems to COTS products
Upgrades	Sometime during upgrading COTS software, it increases the size of the programs & the size of the hardware memory in the system may be insufficient.

3. *Risks involving in Production Strategy*

Table 3 : Risks involving in Production Strategy

For this potential kinds of Risks are:	Risks
Acquisition Alternatives Risks	During evaluation time, alternative methods of acquiring COTS products are not evaluated
Vendor Reliability Risks	Sometimes, the vendor of COTS product is financially weak or unstable & poor support.
Cost and Schedule Completeness:	The cost and schedule estimates are not considered during acquiring the COTS-based system.
Business Skills	The relationship between the contractor and vendor contractor are weak.

iii. *Risk Mitigation*

The main focus is to track, control and reduce the identified risk. A survey was conducted in various CMM level 2 companies which summarized the possibility of risk and corresponding impact of risks. Two approaches are used to calculate the risk score of identified risks in order to plan mitigation approach for the high impact risks.

a. Failure Mode and Effect Analysis (FMEA)
b. Goal-Driven software Risk Management (GSRM)

a. *Failure Mode and Effect Analysis*

A failure mode and effects analysis (FMEA) is a method for examine of potential failure modes within a system for classification by the probability and likelihood of the failures [5]. This procedure helps a team to identify potential failure modes based on past experience with similar products, enabling the team to design those failures out of the system with the minimum effort and resource expenditure. Effects analysis refers to studying the consequences of those failures. To calculate the risk score of identified risks, we are using this approach & filled the questionnaire from the 12 team member based on their past experience of using COTS components.

The probability of each risk item is measuring on likert scale ranging from low (1), moderate (3), and critical (5) as below:

Likert Scale	Probability measurement
Low	1
Moderate	3
critical	5

The impact of corresponding risk item is ranging from very low (0) to critical (5) as below:

Likert Scale	Impact values
Very low	0
Low	1
Moderate	2
High	3
Very high	4
Critical	5

Here are some assumptions of choosing these values:

- It is assuming that the impact of each risk could be different at each phase; it could be or not be same at each phase.

© 2014 Global Journals Inc. (US)

- Suppose there is a probability of arising risk is Low (1), but its impact may be moderate (2) or may be critical (5).

The working formula is:

Risk Score= $\sum_{i,j=1}^{n} Pi * Ij$

Where Pi= Probability of Risk,

Ij = Impact of risk, n= number of respondents

Results of questionnaire: The results that have been conducted from the respondents are shown as below: -

1. *Risk score of Selection Phase*

Table 4 : Risk score of Selection Phase

COTS Driver/Factor	Risk Id	Risk in Selection Phase	Risk Score
Behaviour Factors	RS1	Unavailability of source code	124
	RS2	Organizations have very limited access to product's internal design.	108
	RS 3	The Quality level of a component is unknown.	118
	RS 4	During evaluation, developers have limited chance to verify COTS behaviour.	126
Functionality Factors	RS 5	Requirement of the user and component architecture does not match.	174
	RS6	Architecture of the component is not analyzed according to the functionality.	113
	RS 7	Difficult for requirement engineers to select among different techniques of selection.	86
	RS 8	Lack of market survey.	207
Cost Factor	RS 9	Required COTS is found costly as compared to in-house Development cost.	69

Analysis of Risk Score

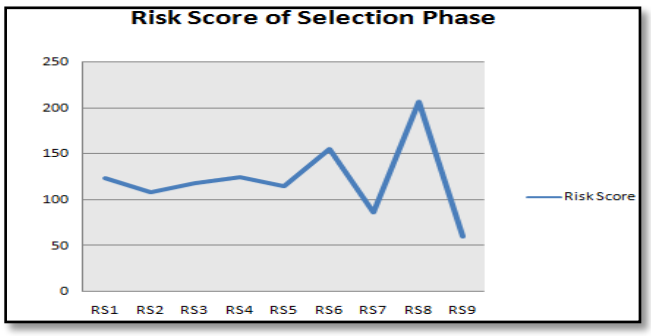

Figure 4 : Analysis of Selection Phase

© 2014 Global Journals Inc. (US)

From the above risk score, we analyzed RS5; RS 8 are critical risks because they have high impact of risks.

2. Risk Score of Integration Phase

Table 5 : Risk Score of Integration Phase

Risk Driver/ Factors	Risk Id	Risks in Integration Phase	Risk Score
Cost Factors	RINT1	Underestimate the development time and cost	122
	RINT2	The cost is too much to configure the components	83
	RINT3	Immature COTS components.	91
	RINT4	Lack of requirement configurations.	211
	RINT5	Lack of cost control.	112
Size Factors	RINT5	Difficult to predict the size of components.	132
Personnel shortfall factors	RINT6	Lack of knowledge.	73
	RINT7	Lack of interoperability standard.	146
	RINT8	Lack of integrator personnel.	150
Security factors	RINT9	Vulnerability risks.	140
Functionality Factors	RINT10	Unavailability of source code.	137
	RINT11	Components are not platform independent.	86

Analysis of Risk Score

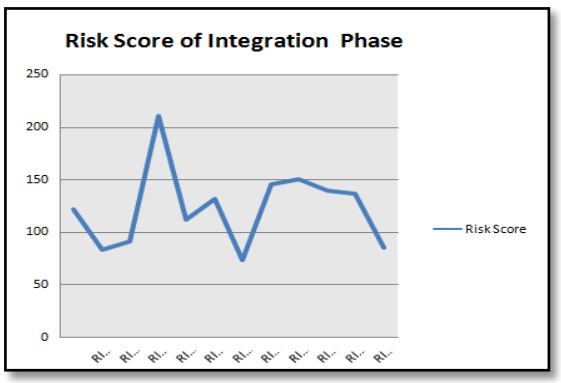

Figure 5 : Analysis of Integration Phase

From the above risk score of Integration phase, we analyzed that RINT 4, RINT 9 are critical risk; because they have high impact of risks.

ii. *Risk Score of Development Phase*

Table 6 : Risks Score in Development Phase

Risk Drivers/ Factors	Risk Id	Risks in Development Phase	Risk Score
Inappropriate Development	RD 1	Risk analysis phase is not present in CBSD.	151

© 2014 Global Journals Inc. (US)

Process	RD 2	Risks are associated due to using an inappropriate development process.	77
Functionality Factors	RD 3	A new version of COTS software may lack new updated code	144
	RD 4	Resources are insufficient.	106
	RD 5	Components are not properly supported by the vendor.	148
Behaviour Factors	RD 6	The estimation of resources {time, cost} is exceeded during development for many projects.	95

Analysis of Risk Score

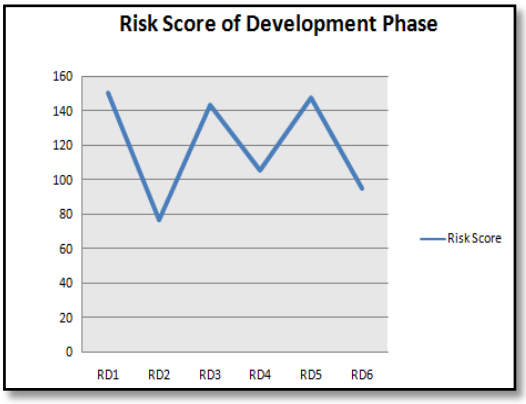

Figure 6 : Analysis of Development Phase

From the above risk score of Development phase, we analyzed that RD 1, RD 5 are critical risk; because they have high impact of risks.

iii. *Risk Score of Implementation Phase*

Table 7 : Risk Score in Implementation Phase

Risk Drivers/ Factors	Risk Id	Risks in Implementation *Phase*	Risk Score
Functionality Factors	RI 1	Unclear design assumptions.	139
Usability Factors	RI 2	Users cannot retrieve relevant & needed information.	97
Security Factors	RI 3	System can be used in unintended way.	132
	RI 4	Increase in vulnerability attack by integrating components with one another.	160
Performance Factors	RI 5	Effect on system performance.	114

Analysis of Risk Score

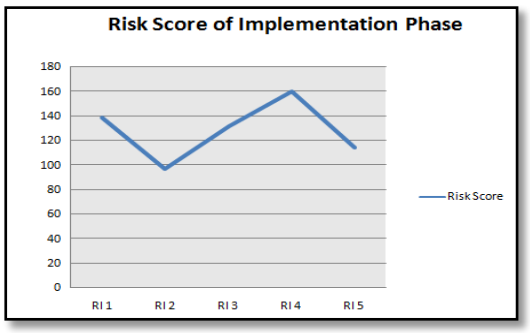

Figure 7 : Analysis of Implementation Phase

From the above risk score of Implementation Phase we analyzed that RI 1, RI 4 are critical risks because they have high impact of risks.

4. *Goal-Driven Software Risk Management (GSRM)*

During study it is analyzed that if the risk in one phase is unseen or undetected, it goes to the second phase and so in this way it impacts to the whole system. If the risk in one phase is not detected, it overlaps to the second phase and increases its multiplicative impact factor [5].

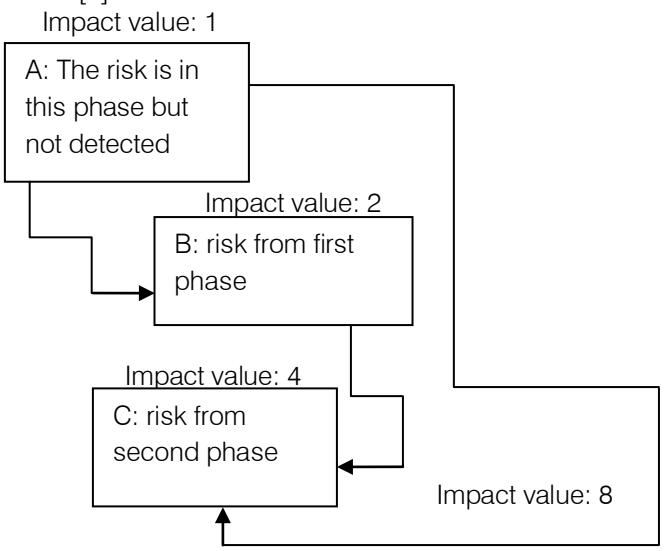

Figure 8 : Impact of Risks during phase-wise

In GSRM approach the main focus is to integrate the whole risk activities, so that we can identify those phases which have high impact of risks and then we can mitigate those risks. So we will calculate the total impact of risks as table 10.

The working formula to calculate total risk is as:

$$\text{Total Risk Score} = \sum RS_k + \sum RINT_k + \sum RD_k + \sum RI_k$$

Where RS_k = Risk in Selection Phase,

$RINT_k$ = Risk in Integration Phase,

RD_k = Risk in Development Phase, RI_k = Risk in Implementation Phase

© 2014 Global Journals Inc. (US)

i. *Total Risk Score of all CBSD (Commercial- Off-The- Shelf Development)*

Table 8 : Analysis of Total Risk Score

Total impact of risk	
CBSD phase	Total Risk
Risk in Selection phase	1098
Risk in Implementation Phase	1481
Risk in Development Phase	721
Risk in Implementation Phase	642

Analysis of Total Risk Score

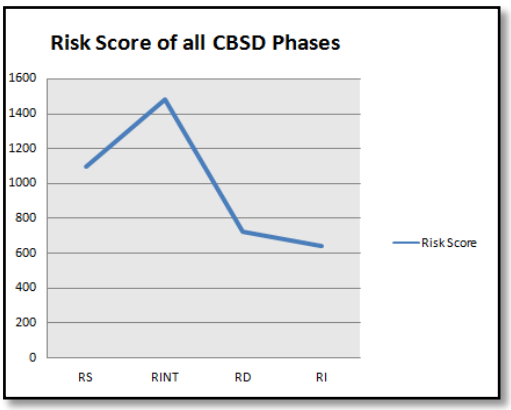

Figure 9 : Analysis of Total Risk Score

From the total risk score of all CBSD phases, we analyzed that Integration phase is more critical. So there is need to mitigate these risks.

a) Risk Mitigation Strategy for Integration phase of CBSD Development approach

From the results obtained during risk analysis, the following graph shows the risk score percentile in various COTS-based Development phases.

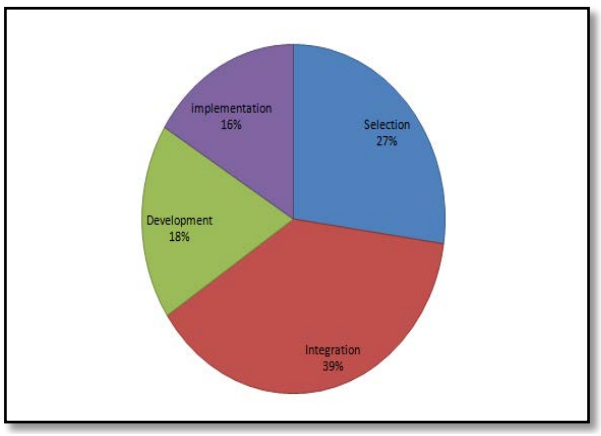

Figure 10 : Risk Score Percentile of all Phases

Now the mitigation strategy will be designed for most critical risk that is Integration Phase.

COTS Integration means when different COTS packages are combine into a system with "glue code". For ex, Office Automation Software, email, messaging system, where the components are bundled as a procedural library [1]. But in this phase many risk arises as:

- Lack of interoperability standard.
- Lack of tools, methods to integrate components.
- Effort for integration may increase from what was estimated.
- When developers try to integrate incompatible COTS components etc.

This integration phase becomes a most challenging phase in Component-based Software Development. The main failures in software arise due to wrong integration of components. As in [4], the recent computer screen upgrade in the British Government caused nearly 80,000 desktop computers to crash The crash halted the United Kingdom's pension and benefits agency that provides benefits to about 24 million people. The crash delayed the process of new claims and forced employees to fax and fill out some payment checks by hand. The problem occurred during an upgrade across the network of computers. So there is need to improve Integration techniques of COTS components.

Mitigation guidelines for Integration of COTS Components:

1.	A proper understanding of component's capabilities is must how components are packaged and evaluated.
2.	A developer should avoid general modifications to COTS components.
3.	Modifications that add the complexity to the project of COTS components should be avoided.

© 2014 Global Journals Inc. (US)

4.	When a developer add or replace a component, it should be integrated system testing.
5.	A proper documentation should be there before buying or developing components from third-party vendors.
6.	A developer should use the components that fulfil with well-known component standards.
7.	A developer, vendor or customer must have knowledge of integration tools.
8.	A developer should use reliable and trustworthy components so that it can minimise the risk of COTS system and provide quality to the system.
9.	The main risk in component system are due to the reason that components are not platform dependent with the system, a developer should provide components that supports adaption to the system
10.	While integrating the components, a developer should choose exact match of COTS components with system requirements instead of approximate match of COTS components.
11.	A developer should use open Standard technologies that are freely distributed among different data models or software infrastructure which provide basis for communication and enable consistency among different COTS components [6].
12.	A proper estimation of time and cost should be estimated, before integrating COTS Components.
13.	All drivers should be considered before measuring component behaviour. For ex, ACIEP-used for COTS Integrator Experience with the product, ACIPC - used for COTS Integrator Personnel Capability.

IV. Conlcusion

Commercial-off-The-Shelf Software Development has become a great need for large organizations as it saves development time and money. It is belief that COTS components fulfill everyone's needs and can be used as-is. In reality, the risk arises in each phase of CBSD as, COTS selection, Integration, Development and on maintenance phase. In this paper, the main focus is to provide risk identification strategy for COTS based software Development. The risk adds on each phase of CBSD was identified and risk score is calculated to examine the critical risk phase.

References Références Referencias

1. Dr. Sohail Asghar, Mahrukh Umar, " Requirements Engineering Challenges in Development of Software Applications and selection of Customer-off-The-Shelf (COTS) components", in International Journal of Software Engineering(IJSE), 2010, (pp 32-50).
2. "Risk Management Guide for DOD Acquisition", in OUSD (AT&L) Systems and Software Engineering/Enterprise Development.
3. James Everett Tollerson, Hisham M. Haddad, "Conceptual Model for Integration of COTS Components" in Department of Computer science &IT (pp 1-7).
4. Amandeep Kaur & Shivani Goel, "Designing of RIMCOTS model for Risk identification and mitigation for COTS-based Software Development" in Research Journal of Computer Systems Engineering- an International Journal.
5. Saima Amber, Narmeen Shawoo & Saira Begum, "Determination of Risk During Requirement Engineering Process" in Journal of Emerging Trends in Computing and Information Sciences (pp 358-364).
6. Palak Arora, Amandeep kaur, "Improving COTS-based Software Development Process by Identification and Mitigation of Component Risks" in International Journal of Advanced Research in Computer Science and Software Engineering, 2013, (pp 219-225).
7. "Failure Effect Mode Analysis (FMEA) "in Institute for healthcare Improvements.

© 2014 Global Journals Inc. (US)

This page is intentionally left blank

© 2014 Global Journals Inc. (US)