Application of the Contribution Acceptance Process Model in the OSS Environment

Shiva Tyagi[1], Rajendra Kumar Bharti[2], Sachin Kumar[3]

[1]Research Scholar, Uttarakhand Technical University, Dehradun, Uttarakhand, India.
tyagishiva1@gmail.com
[2]Bipin Tripathi Kumaon Institute of Technology, Dwarahat, Uttarakhand, India.
rajendramail1980@gmail.com
[3]Ajay Kumar Garg Engineering College, Ghaziabad, Uttar Pradesh, India.
imsachingupta@rediffmail.com
Corresponding author E-mail: tyagishiva1@gmail.com

Corresponding Author:
Shiva Tyagi
Research Scholar,
Uttarakhand Technical University,
Dehradun, Uttarakhand, India.
Email: tyagishiva1@gmail.com

ABSTRACT

The OSS ecosystems have changed the ways the software industry has been functioning. The firm owners have a lot of options to choose from in terms of the contribution for their product development is concerned. The open innovation (OI) is another important terminology that this paper ventures into. The study of the OSS in the backdrop of the open innovation and its effects on the planning and contribution strategies of a firm are the critical factors discussed in this paper. The earlier proposed CAP model has been studied in detail and has been superimposed on the information meta model to arrive at a workable solution for the software firms. Considering all these factors, the OSS governance boards in a firm can take the correct and timely decision on their source of supply. The CAP model helps in the classification of different artifacts based on the business impact and control complexity. So based on this the contribution strategies may be made.

Keywords—Open Innovation; Contribution Acceptance Process; OSS; Repository; Information Meta Model

Tob Regul Sci.™ 2021;7(6-1): 7546-7557
DOI: doi.org/10.18001/TRS.7.6.1.73

1. INTRODUCTION

The terminology of Open Innovation (OI) has been in the news for quite some time now and it has been able to draw attention of a large number of scholars from different parts of the World belonging to a wide variety of disciplines since the time of its evolution. However, it still remains untouched or unexplored if we discuss it in terms of software engineering [1]. But OSS environment is a different ballgame altogether. Off late, a large number of firms have been opting for OSS model as it changes the very pace of a firm's internal innovation methodology. Its major reason lies in the fact that a firm gets an easy and unhindered access to the external workforce [2]. This reduces the cost of internal maintenance and enhances the quality of the output, which is a win-win situation for any firm owner. Inflow of OSS ecosystem features is another benefit that the owners look out for. So it is clear that OI and OSS are made for each other and may be represented as the two side of the same coin. Any organization involving OI has been able to achieve far better financial output on the investment vis-à-vis the organization not involving OI. The sources of external knowledge available with the firms using OI and the OSS is the main reason for the better and a profitable output in their case as per the latest studies carried on the subject [3].

If a firm has to realize the importance and the likely future benefits it is likely to obtain by adopting the OI in the OSS environment, it has to have their product strategy in complete synchronization with the product planning. Also it has to work on its positioning in the ecosystem governance structure along with the participation in the OSS ecosystem. So a firm has to take a call to employ OSS completely or partially. It may prefer to base its products on OSS or it may even choose to employ OSS as a part or subpart of its sourcing strategy. To enable such synchronization, a firm has to enhance its product planning in such a manner that it becomes an easy decision for the firm to decide what portion to keep closed and what all to be left out for contributing towards OSS. This may be loosely termed as the strategic product planning in OI. The contribution strategy, where we decide what and when to contribute gains importance here and plays a vital role.

In this paper, the contribution acceptance process (CAP) model will be analyzed in detail which will try and bridge the gap between various contribution strategies that the firm may adopt for the purpose of enhancing their profits and increasing the quality of their output. Considering OSS as an asset which is not a part of the company, the CAP model is roughly based on the Kraljic's Portfolio Purchasing Model which was given by Peter Kraljic in 1983. Its purpose is to help a buyer increase the supply security and minimize the cost by making the most of a company's purchasing power. So, it is clear from the above that the CAP model is the shoot from the famous Kraljic's model and when in case of a software firm, one has to decide which sourcing option to opt for, the CAP model comes in very handy and deserves to be analyzed in thorough detail [4]. And that is what the authors of this paper intend to do in the subsequent sections of this paper. The study of the CAP model is a must for the firms involving OSS strategy and analyzing OI benefits, like increased and improved innovation and lesser time to market. In the later part of the paper, an information Meta model will also be discussed and analyzed for its features and shortcomings. That will assist the firms to operationalise the CAP model. The meta model is an information based support model and needs to be implemented in conjunction with the requirement management infrastructure of the firm so that it can influence the contribution strategies of the firm. All this will be studied and analysed in the backdrop of a case study involving three different firms to enable generalizability of the discussed models.

2. RELATED WORKS

One needs to be well aware of the work that has gone into studies on the subject to be able to find in a connection with respect to how the software engineering along with the open source software may be linked with the Open Innovation (OI) [5]. This is also required to understand the origin source of the CAP model. OI is generally explained using the funnel model which represents a firm's R&D procedure. The funnel is supposedly permeable as the firm can be in communication with outside environment. This is more pertinent in case of an OSS. The coupled innovations take place where in inside – out and outside – in transactions occur simultaneously (Fig 1).

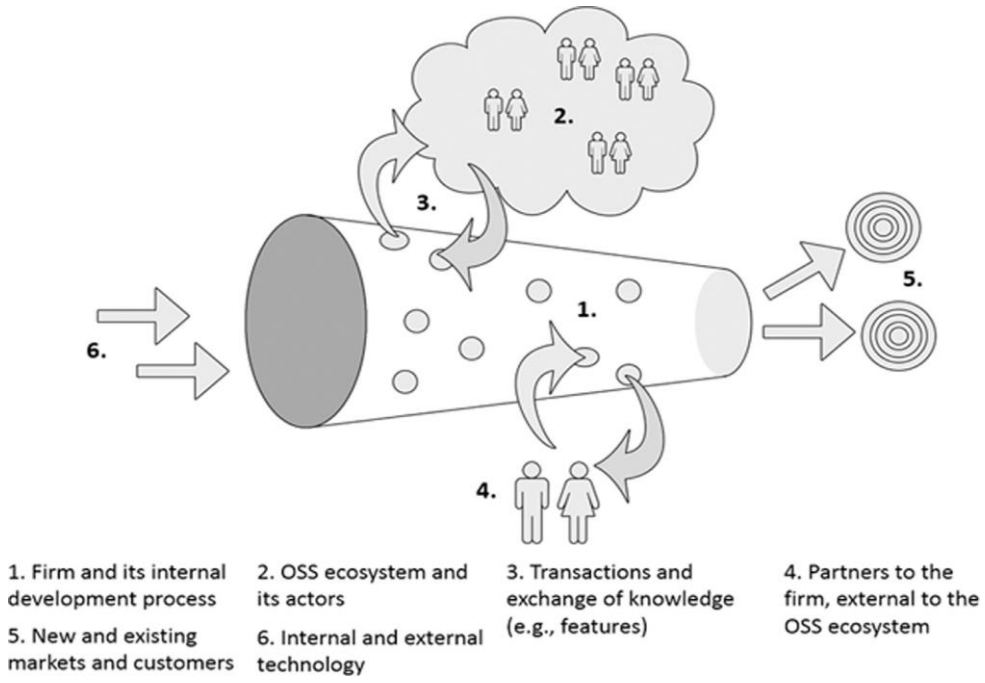

Fig. 1 Funnel Model

Contribution strategy to decide what and when to contribute as an OSS, was given by Wnuk in 2012. It is a must for a firm to know how it performs in various different OSS ecosystems. Only then can a firm clearly decide on what to contribute and when to

contribute [6]. Depending on this, a strategy may be developed which can dictate the level of openness. This may vary from completely open to party open to completely closed. Apart from this there has been a lot of studies on the subject from the authors and researchers the World over. They have all proposed different models that a firm can adopt in order to maximize their profit and reduce the risk factor. In this paper, the focus will be primarily on the Kraljic's Portfolio Purchasing Model, the Contribution Acceptance Process Model (CAP Model) and the Information Meta Model.

3. KRALJIC'S PORTFOLIO PURCHASING MODEL

The models that describe a software artifact's value depreciation is termed as commoditization model. This model helps a firm to come to a decision of what and when may be contributed to OSS environment. Commoditization is also related to the life cycle of a product. This model also shows that commoditization is a continuous process and it is an inevitable one for all software artifacts. Another important and relevant model on the subject is The Kraljic portfolio purchasing model. In a software field, there are four major sourcing options available to the firm owners. They can opt for in house, outsourcing, COTS or the OSS. Generally there are two sourcing strategies considered [7]. The decision to choose from a specific source is a very critical one and it simplifies the argument how commodity part should be acquired, contributed and sourced in different ways. Kraljic portfolio purchasing model helps in developing the sourcing strategy of different materials and vast range of components that may be required for a product. The supply items are classified into profit impact and supply risk dimensions. These supply items are then accordingly positioned in a matrix. The four quadrants formed represent the distinct item category with its own specific purchasing strategy for the suppliers (Fig 2).

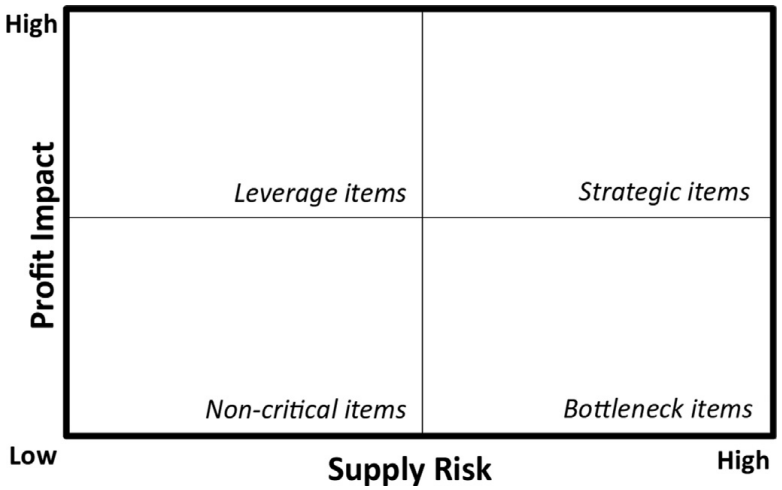

Fig. 2 The matrix used in Kraljic's portfolio purchasing model

In Fig 2, the strategic items are those which have the high profit impact and high supply risk. They are generally the single supplier items. The leverage items are the high profit impact and low supply risk items. Items are generally easily available and the buying power may be exploited within the supplier market. The bottleneck items are those that characterizes high supply risk and low profit impact. Here, one has to accept the dependence and make effort to reduce the negative effects. And lastly, the low supply risk and low profit impact items are the non critical items that have relatively very low added value per item.

The strategic product planning in OI is a critical requirement that has to be met in order to define the product and describe how the product will evolve with time. The aspects that should be considered here are quality, delivery model, target model, product positioning and sourcing options available and feasible. Product planning has to be carefully executed utilizing the road mapping, release planning and requirement management process. So the product planning in an OSS environment concerns the executive management.

In case of software engineering, artifacts might be structured and positioned in various separate ways depending on different contexts and different processes involved. The artifact has to support the inter communication between different departments in an organization. The software artifacts are to be stored centrally in a firm as a database that requires qualities like traceability and completeness. Opposite to this, an OSS ecosystem represents the other end of the spectrum. Here the approach is very informal. Here, the requirement is highlighted by many artifacts, which are in a manner complementing each other to present a complete picture. These are termed as informalisms.

4. CONDUCT OF THE RESEARCH

It is time now to discuss the research methodology. The study process and the research questions involved in conduct of the research are discussed in this section. The reference for the data has been taken from the paper written by J Linakar and H Munir in year 2016/17. Taking the case discussed of the Sony Mobile multinational firm which has an employee database of more than 5000 employees [8]. The mobiles and tablets are manufactured there. A specific branch was taken as a sample case with approx strength of 1600 employees. The software development cell of Sony mobile makes use of the agile technologies are in a process of developing more than 20000 different features and applications to be used the mobiles and tablets. Sony mobile makes full use of OSS and has involvement in several OSS projects. To manage it, they have a body termed as OSS governance board. With all these details in mind, we arrive at the research questions, the first of which is as to how can the contribution strategies be structured and designed which can support the product planning in the backdrop of OI? The second question that is required to be answered for compiling the data is asking as to what software and product planning artifact repositories and types are required and how to represent them in a meta model in terms of strategic product planning? It is assumed that these two questions should be able to help us create the base for exposing the benefits of OI.

With this study, we intend to identify the problem in the initial phase and after this, the artifact design process starts. In this the artifacts addressing the research problems was created. And lastly the validation is done. The data was collected throughout the process and it was only finally that the conclusion was given after reporting of the results. In the problem identification process, the detailed understanding of the problem context and current practices is done. The informal discussion with the decision making body of the Sony Mobile was carried out to find out the importance being given to OSS by the firm. The detailed study of the documentation of the company was done regarding the OSS. The study and the informal interaction with the firm employees revealed that in order to arrive at a meaningful and productive outcome, there is a requirement of a combination of the technology based artifact and the organization based artifact [9]. The technology based artifacts lets the firms decide onto the contribution strategies for software whereas the organizational artifacts let the firm decide regarding the adoption of technology based artifacts. The answer to the first question gives us an artifact that lets the decision makers decide on the extent of the OSS that may be involved in the company process. The extent of the contribution of the OSS ecosystem is decided by the answer to this question. A matrix is created that allows the company to decide how to source and purchase particular material for their production.

In order to find the details as asked in second question, an information meta-model was formulated that facilitated communication and follow up on software artifacts along with their strategies for contribution. Initially during the investigation phase, it came out clearly and visibly that the information support is required to be integrated with the software artifact repositories which are utilized for management of requirements. It is only then that the information support would be able to effectively able to reach those involved in product planning and development. This led the investigators to six different and relevant repositories that affect our study [10]. They are the feature repository, the internal product portfolio, patch repository, architectural asset repository, commit repository and the contribution repository. The detailed description of specific attributes of these repositories has been presented in a tabulated manner in Table 1. This enabled the investigators to trace the contributions for the architectural assets, product requirements and platforms through the developer created patches for internal source code branches. With the data collected in this was analysed and the information meta model was formulated. The interconnection of the various repositories as mentioned above has been given in detail in the given figure (Fig 3).

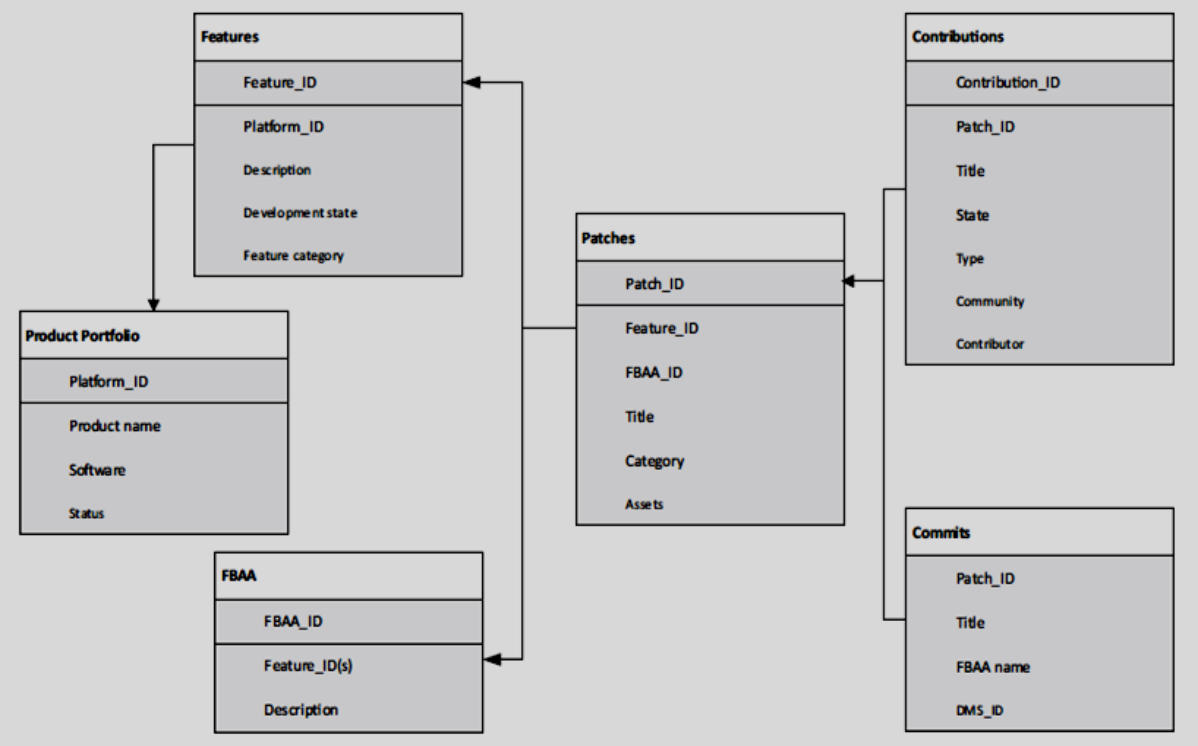

Fig.3 Meta model formulated with necessary software artifact repositories

The artifact validation is an important step in the above study and it helps to ascertain if the solution given by the candidates addresses the identified research problems or not and that too in an accurate manner. Artifact validation is done in two ways. They may be static validation or dynamic validation. In a static validation, the candidate solution is presented to the industry practitioners and a feedback is taken that refines the solution as per the design science process. In a dynamic validation, one expects to impose the candidate solution in the real work setting [11]. But that is not in practice as of now and may be considered in future studies.

Repository Name	Attributes	Description
Products	Platform ID	A unique ID for platform name
	Product name	Product name with the platform.
	Software	Related software description, e.g., Android, OSE, Epice, Kept etc.
	Status	Current standing of the platform, e.g., expired, announced etc.
Features	Feature ID	A unique Id for a feature, which refers to features.
	Platform ID	ID associated with the specific platform e.g. android, core etc.
	Description	Details of the feature.
	Development state	Refers to the current status a feature's implementation, e.g., started, executed.
	Feature category	Refers to the type of feature, e.g., new functionality, bug fix, extension etc.
	Contribution Strategy	Refers to whether the requirement is contributable or not.
FBAA	FBAA ID	A unique Id for each Feature Based Architecture Asset (FBAA).
	FP IDs	A combination of FP IDs associated with the FBAA.
	Description	Details of a FBAA.
Patches	Patch ID	A unique id for each patch.
	FP ID	A unique ID from the FP repository.
	FBAA ID	A unique ID from the FBAA repository.
	Title	A description of a patch.
	Category	Importance of a patch, e.g., market critical, development critical, stability, ecosystem critical etc.
	Assets	Refers to the type of a patch, e.g., bug fix, extension, operator requirement, platform related, generic etc.
Contributions	Contribution ID	A unique ID for each contribution.
	Patch ID	A unique ID from the patches repositories.
	Title	A description of a contribution.
	State	Refers the current state of the patch, e.g., ecosystem merged, already fixed, CEO rejected, legal reject, ecosystem review etc.
	Type	Refers to criticality of a contribution, e.g., trivial, non-trivial, bug fix etc.
	ecosystem	Refers to the ecosystem in which the contribution will be made, e.g., Google, Firefox etc.
	Contributors	Refers the contributor information.
Commits	Patch ID	A unique Id from the patch repository.
	Title	A detailed description of a commit.
	FBAA name	Commits associated with the FBAA.

Table 1. Description of selected attributes from the software artifact repositories.

5. THE CONTRIBUTION ACCEPTANCE PROCESS (CAP) MODEL

The CAP model has a base in the portfolio model which was proposed by Kraljic in 1983. The same has been discussed briefly in section 3 for this paper. This was developed way back to create the purchasing strategies for the firms from different suppliers. This model is based on software artifacts and the ways in which they should be sourced contributed as OSS. This model can be used both reactively and proactively. While using it reactively, the utility of the model is made to follow up on previously classified artifacts and for specific contribution request from development organization. In proactive utility, the model is systematically used on a portfolio to take a decision on the individual specific contribution strategies for each artifact and to get the overall picture of the artifacts in relation to each other. The matrix showing the CAP model has been presented in Fig. 4 [4].

The contribution acceptance process (CAP) model in a business impact vs control complexity scenario

The focal point of the matrix represents the CAP model in a simplified manner. In this an artifact is classified into any one of the four given quadrants which shows a particular artifact type with certain contribution strategy and characteristics. Based on this, the four types of the artifacts are as under:

Strategic artifact : *high business impact and high control complexity.*
Platform/ leverage artifact : *high business impact and low control complexity.*
Product/ bottleneck artifact : *low business impact and high control complexity.*
Standard artifact : *low business impact and low control complexity.*

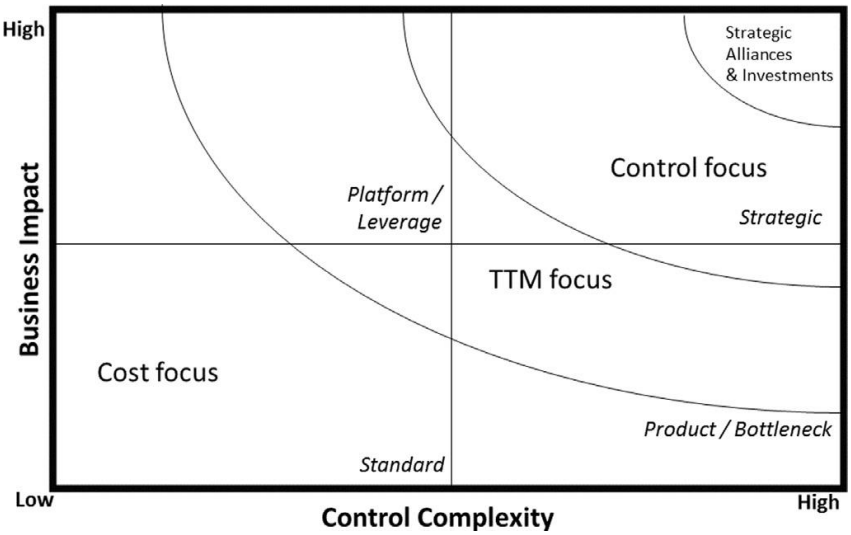

Fig. 4

6. ANALYSING THE REACTIVE APPROACH OF THE CAP MODEL

While trying to make use of the CAP model reactively, the study was carried out of the current practices on Sony Mobile's by J Linaker in 2016. In this approach, the keyword is the following up of the already existing artifacts that are classified in nature. This classification will depend on the life cycle of the technology of the artifact. With the reactive approach, separate contribution requests can be managed from the developing organization. To explain it, an example may be taken of a manager putting in a request for contributing to a certain artifact or putting in a request to work actively in a certain predesignated open source software ecosystem. So as and when someone desires to contribute, they end up passing through the board. But this leads to a bottleneck effect and bureaucracy at times. To avoid it, the process of the contribution has to be varied based on the level of complexity and the variety in a contribution. There are different levels of contributions in a CAP model. The three levels may be trivial, medium and major contributions. The trivial ones add to only a small level of change in the OSS ecosystems that are already in place. They enhance the codes that are not so significant and do not add to any kind of upgraded functionality in the system. They are more or less like the bug fixing and refactoring in a program. For this, only the approval of business is needed. Moving on to the medium ones, they involve substantially reformed functionality which may be completely new. They add on the new features and even the architectural changes. Coming over to major contributions now. They involve new codes and a major change in terms of IPR. Major internal changes are incorporated in the basic architecture of the setup and a lot of efforts go into developing it. For medium and major contributions, business manager alone cannot give the approval. For them, he has to prepare a statement of case so as to let the Governance board take

a decision keeping the legal and IPR factors in mind for the OSS contribution. The case is thoroughly investigated before taking a decision pertaining to medium or major contributions. After taking due cognizance of all the factors, the board may decide to accept or reject the proposal from the engineers

7. ANALYSING THE PROACTIVE APPROACH OF THE CAP MODEL

Using the proactive approach is a complicated process and the process has to be completed step by step. The suggested approach to opt when making the use of the model in a proactive manner is as follows. In step 1, there is an urgent and inescapable need to decide on the level of abstraction and the scope to initiate the process. The scope may cover a product, the active functional area and maybe the platform that we are planning to use. The abstractions level may be decided that can cover factors like features, requirements and components. Keeping all this in the backdrop, the background information is collected that can be market intelligence, details regarding architecture, the analysis of the impact that is likely to affect the output, the OSS based ecosystem details and may be license compliance factors. All this can be the start point for the open discussion forum which takes us to step 2, wherein classification has to be done to the level of abstraction and the mapping of the artifacts is required to be done as per the matrix. For this, one can begin with the first or the original set of the artifacts that are part of the matrix. This may be followed by the synchronizing and emphasizing the mapping of the artifacts. After this, the balance of the artifacts in the matrix may be mapped. The classification may be done based on the set of questions. The joint answers to all the questions posed may be mapped on a Likert Scale wherein the values are given from 1 to 4. Based on the outcome from the Likert Scale, decision may be taken as to which direction of the two quadrants does the artifact lie. The questions may be framed in the following suggested manner like how will the firm's profit and revenue get affected by the process, how will the customer and the end user value be affected, how is the product differentiation likely to be affected in the bargain, the same factors about the leading technology and the ongoing trends, and may be how is it going to affect in case of any kind of shortage? All these answers are to be answered on a scale of 1 to 4.The answers to these questions will tell the managers if they do have the knowhow and capability to adapt to the technology, or if any barriers regarding technology exist in their system, the level of innovative mindset and lack of alternate options along with the limitations faced by the firms can be arrived at using the answers to the questions posed. The mean value of both x axis and y axis is calculated and the artifact is then plotted onto a matrix (Figure -4). This will give four quadrants. Then a general consensus is derived at keeping the contribution strategy and the contribution objective in mind. In this manner, all can get accustomed to the process followed and the classification criteria used. The entire procedure is informal keeping the focus on the quality. After all kinds of deliberations of the answers to the questions and plotting the same on the matrix, a final set of discussions and reflections is required to be performed which takes us to the step 3.

Step 3 is nothing but the further reiteration of mapping of the artifacts which caters for any factor left unattended. And finally when the group reaches a consensus, it is must that entire procedure is documented, which is the step number 4. In the step 4, one should focus on the documentation of whatever has been the outcome of the initial 3 steps and the decisions may then be communicated to the desired members in a firm with the required channels which are utilized by the information meta model. In the step 5, preferably do not do anything but monitor the progress and keep following up the decisions. The same required to be done against a specific time frame. This task may be performed by the project management for the sole reason that they have the accountability towards OSS executive of the firm.

6. PUTTING THE CAP MODEL IN OPERATIONAL USE

If one desires to put specific contribution strategies into practice in a firm, there has to be a proper process for the same. One requires information support that can tell which artifacts completely or in part requires to be contributed. If one desires to follow contribution strategy implementation and adapt accordingly, then it is required to see what, where and when has been contributed in the field [8][12].The efforts in this section will be to find the answer to our second question and arrive at the information meta model through which we shall be able to record and convey the operational utility of the CAP model. When the meta model was superimposed on the data collected from the Sony Mobiles, the investigation resulted in the choosing of six repositories as shown in Fig 3 which are as under:

•**Product Portfolio repository**: The product portfolio repository is required to support a firm's software platform strategy where one platform is reused in multiple products.

•**Features repository:** It stores the information about each and every feature which may require to be updated by different roles as the product development takes place.

•**Feature-Based Architecture Assets repository:** This repository stores the features that have the common functionality in different kinds of products

•**Patch repository:** The adaptations in a firm that help the firm to update the product as per customer requirements are stored as a patch in the Patch repository.

•**Contribution repository:** Those patches that have been contributed back in the OSS ecosystems are stored in the contribution repository.

•**Commit repository:** Information is stored in patch artifacts about the technical implementation and is treated as an abstraction layer and stored in commit repository.

7. THE USABILITY OF THE CAP MODEL

The CAP model assists the OSS governance board to closely monitor the requests for the contribution and plan the decisions strategically. It offers a blueprint to home on to those projects that are more beneficial to the company. It can help the decision makers to align the contribution decisions with internal strategies of the business and the product and that too while considering the dimensions of complexity in a business. The questions decided in the beginning of the paper have proved to be very effective in coming to the conclusion about the importance to the CAP model in a firm. The questions may be further modified based on the different firms. The OSS governance board applies the CAP model as and when they have to review the new project.

8. PROCESS OF INITIALISING DECISION MAKING

The initialising or enabling decision making involves a composition for supporting the architecture. The architectural component of the software is involved the moment we take a decision for composite kind of software. This might be like choosing a particular COTS component of the software. Association between a class and the other expresses it clearly. It involves decision based on the evidence and the architecture of a composite software. At the same time, author feels it is an unavoidable requirement that there must be a shared vocabulary to enable and common and correct understanding in the context of decision making at all levels by one and all and also to cater for the utilizing the evidence from varied taxonomy. In a scenario, when there is no shared vocabulary, it is very likely that there might be a lot of misunderstanding and the varied evidence, as mentioned earlier, may not stay usable, or will become very hard to reuse between different decisions which are independent of the evidence source. All this is represented in the UML diagram given in figure 5. This figure shows the areas and their relations with the help of a Unified Modelling Language diagram. A special mention is required to the comments that are associated with Taxonomy, Decision maker and Evidence based Knowledge Repository.

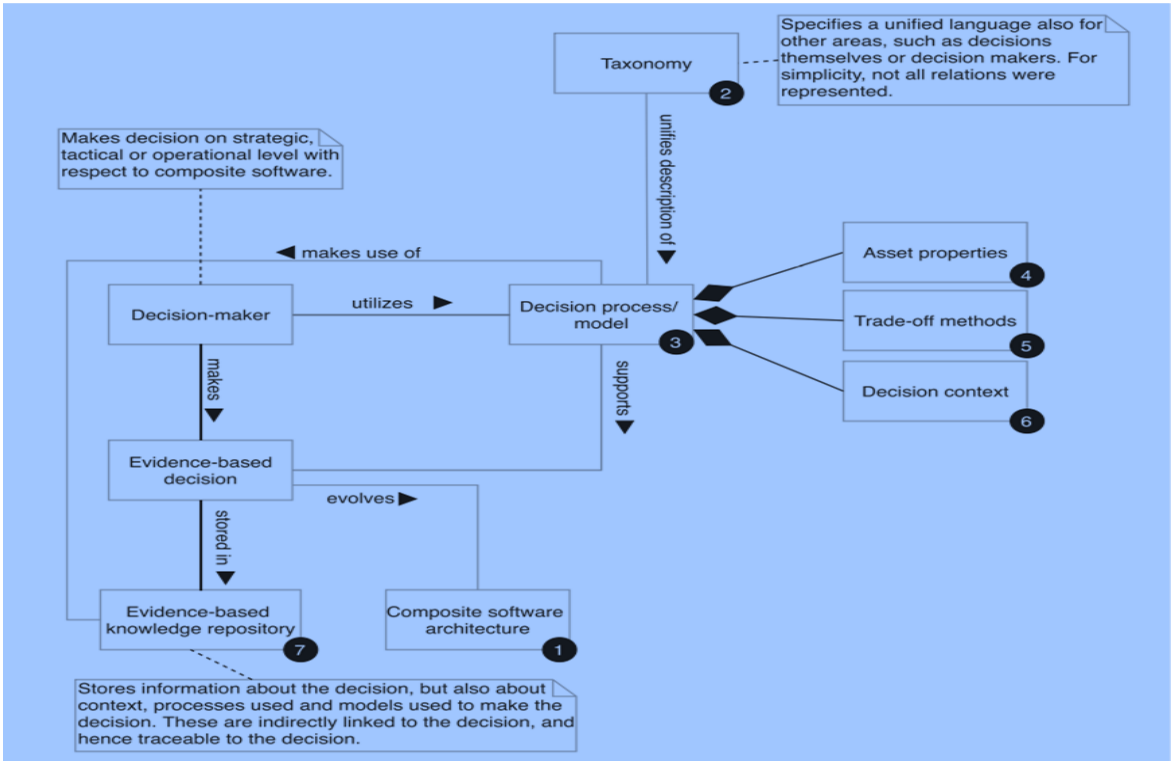

Fig.5 Research Areas: An Overview

There are a few areas that have been identified where some amount of research needs to go into. A specific process or a model pertaining to decision is required to support the decision making which can ensure that no process and no stakeholder is left out. It is also a must to include as the different models vary in their performance and output in terms of flexibility and quickness of taking decision. With this at the backdrop and looking at the options of the assets and industrial practices, they become the key drivers in the process of making a decision, with the help of which the decision options were assessed. If we take it that all the required properties may not be possibly achieved simultaneously, it becomes necessary that we opt for making the trade-offs. This leads us to include the area termed as trade off methods. It emphasizes the requirement to have support for explaining the basis of making decision. This form the basis of going in for evidence-based approach, as the evidence is likely to be dependent on the context. This particular phenomenon has been highlighted by a large number of researchers and is being explored in detail by the research scholars from all over the world.
In support of an approach that is evidence based, there is an inescapable requirement of a knowledge repository. This will act as a place for storing all the available information at a place that may be made available to the decision makers as and when required. For example, the idea of a knowledge repository based on evidence is a necessity in the field of health care. The information stored in a repository could be of any type from the data to context descriptions to problems faced to decisions taken to the success or the failure of the decision. So, it is not a hard and fast rule as to what may be stored. The idea is to store any information that may come handy at any point of time. It is left to the user to decide as to what information he wants and how much evidence-based input is he looking after. It is for the user to decide the relevance of the required information and the available information. For this, the knowledge repository is required to be filled or populated with the information that is relevant to the decision makers of different fields. This is, in itself an extremely challenging task as one can never be sure as to what information is more critical for the decision makers in what scenario. As shown in the figure above, we may identify seven areas of interest regarding the process of initializing decision making. The details of the classes as per the figure 5 above may be enumerated as below:

(a) Composite S/W architecture. It involves a development of a suitable decision-making procedure that may serve a wide variety of architectural techniques and a different domain of application of the information.
(b) Unified language and taxonomy. As discussed earlier, it is required to have a common language and a common structure to exhibit and keep up the decision making as an important facet.
(c) Decision processes and models. The development of the decision processes and related models as checklists is an important factor that can help the user or the decision maker in the process of selecting amongst the various origins of different assets.
(d) Asset Properties. Enabling the possibilities of making the identification of important properties that are likely to be the outcome of the different solutions and the different assets.
(e) Decision Trade-Off. Creating the decision trade off situations so that the software asset selection and the comparison by taking into consideration the factors like asset properties and environmental factors.
(f) Decision Context. The context may be utilised to enhance the decision-making qualities of an individual as it enhances the suitability of solutions which are in a large way dependent on the context.
(g) Evidence-based knowledge repository. Creating a knowledge repository to support decision making by balancing utility-based usefulness and identification of documentation costs.

9. DECISION CONTEXT

In software engineering, the most vital role is played by the context. It includes the research as well as practice in the software engineering. In research, it is strongly argued that context is extremely vital as it decides the suitability of different solutions. In terms of practice or the industrial practice, different methods rely wholly or partially on the context. The contextual information is retained by the experience factory in a manner by which it classifies and characterises the various projects for which different experiences are retained to enable identification of similar projects. Processes of requirements take into consideration the specifications of the system context like environmental usage scenario along with the specifications of the requirements of the system. Generally, the contextual inputs serve the primary purpose of making decisions. This is supported by the experience factory that helps to decide the good practices that may be used in a particular project. Along with this, the context information in the process of requirements helps in deciding which all requirement projections are applicable and valid. The architectural decisions are also analysed and explicated with the contextual setup in mind. We are likely to face three challenges with respect to decisions that we take and may form the basis for future research. They are discussed in detail in the succeeding paragraphs.

(a) Determination of relevant context information. For the purpose of decisions that are architectural in nature, one needs to ask about which particular contextual information to gather. They have been made amply clear in the figure 6 below. It caters of the contextual information keeping the relevance as the key property. The figure shows five different dimensions, i.e product, stakeholder, business and market, development and methodology and the organisation. Certain context elements that may be described are also shown in the figure, even though their relevance will be decided on case-to-case basis. This context model is a visual representation of the model. The questions may be answered based on the empirical evidence based on a number of decisions. In particular, the research has to identify what information plays crucial role in decision making. It is because recording all the information is not practically feasible. This may be studied with the help of the decision and the outcome achieved. If we consider an example, in any particular scenario, from the perspective of research method, multiple directions are suggested like gaining of in-depth insights of decisions with the help of different case studies and also like gaining general insights with a large scale gathering of decision cases with the help of surveys. The insights that have been gained earlier serve as a guide to develop to form case surveys. The gathered information then needs to be organised. Decision makers will have to use the context as a filter to identify the decisions relevant to them and their problem.

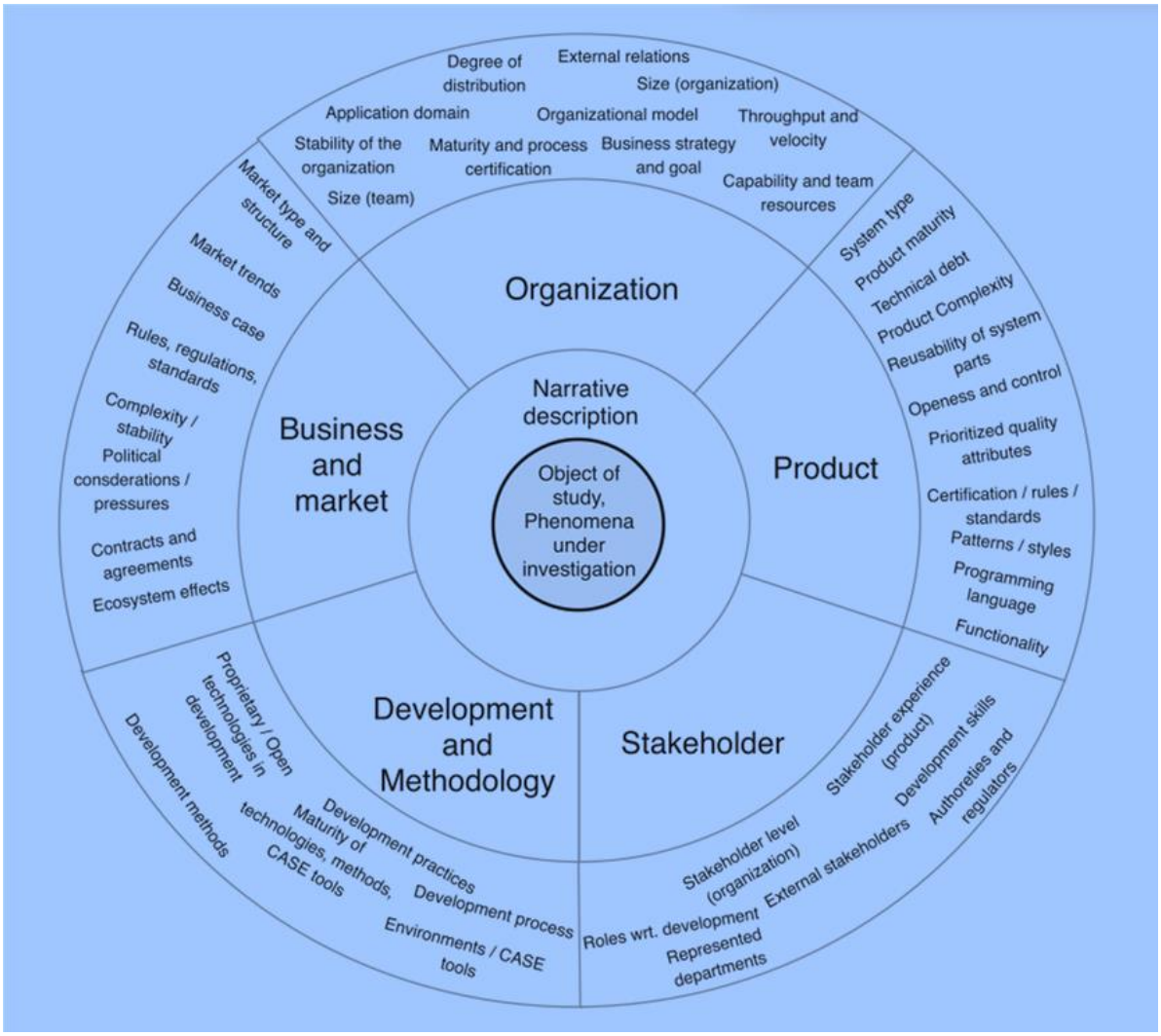

Fig. 6 Context information with 5 different dimensions

(b) Determining mechanisms to use context as a filter to identify relevant decisions. Decision makers are required to make use of the context to identify those decisions that are in particular specifically useful for their problem. So, there is a need to identify the specific scenarios by which any researcher would most likely be searching the particular decisions to visualise the inputs so that it can specifically support the selection and identification of relevant decisions. It is thus proposed that a study be carried out to identify the presentation methodology of the information is such a manner that the most relevant outcome may be drawn.

(c) Determining how to use contextual information in the context of decision making. After homing on to the ways in which the information has to be presented and identifying relevant requirements, there is a need to focus on a firm decision-making process that has to be put in practice. So, there is no harm identifying the particular information to consider when finding out similar scenarios and the method to weigh the various entries in the process of evaluating the identical factors in different cases.

10. CONCLUSION

The software business has undergone a complete change in the recent past wherein the firms have now started making use of the OSS very extensively. The firm owners have started rethinking and re planning their strategies to ensure the long term and sustained competitive advantage. After the advent of OSS, the firms now have adopted the value creation with shortened time to market and a completely new changed commoditization process. If one has to make full use of the changed ecosystem, there is a need to have improved strategic planning support system in place. The most important decision that a firm owner has to take is what to develop internally and what to outsource / open up. The available commoditization model caters for a lot of things but lacks in providing operational support that lets the decision governance body to decide what to contribute and when to contribute to the OSS ecosystem. The CAP model that has been discussed thoroughly in this paper tries to bridge the gap between product strategy and product planning at a firm level. This model does cater for the commoditization in the software industry. Contribution strategies may be set in relation to the business value and control complexity aspects. The firm that is keen to involve OSS in their product development and takes on the OI environment can reap greater benefits while working in a more organized manner.

10. FUTURE WORK

As they say that the pink is the new black, the OSS is the new age way for software development. The future of the software firms lie in adopting the OSS in a deliberate manner. This paper also leaves the readers and researchers with a lot of work to do on the subject. This paper provides a base for all the further studies on the subject and the authors of this paper would happily and willingly volunteer to provide any kind of assistance that the future researchers might require from them. There are a whole lot of avenues for work on the CAP model. This paper just introduces the topic to the readers and would like to invite the researchers from the World over to give different connotations to the CAP model after taking the data from different types of firms.

REFERENCES

[1] West, J., Bogers, M., 2013. Leveraging external sources of innovation: a review of research on open innovation. J. Prod. Innov. Manage. (2013).

[2] Ven, K.,Mannaert, H. , 2008. Challenges and strategies in the use of open source software by independent software vendors. Inf. Softw. Technol. 50 (9), 991–1002.

[3] Baars, A., Jansen, S., 2012. A framework for software ecosystem governance. In: International Conference of Software Business. Springer, pp. 168–180.

[4] Tyagi, S., Kumar, D, and Kumar, S. Understanding the nittygrities of software reliability and its testing procedures: a different approach, Journal of Information and Optimization Sciences, 2017, Vol. 38, No.6, pp.971–988.

[5] Tyagi, S., Kumar, D, and Kumar, S. Open source software: analysis of available reliability models keeping security in the forefront, International Journal of Information Technology, 2019, ISSN 2511-2104.

[6] Enkel, E., Gassmann, O., Chesbrough, H., 2009. Open r&d and open innovation: ex- ploring the phenomenon. R&d Manage. 39 (4), 311–316.

[7] Badampudi, D.,Wohlin, C. , Petersen, K., 2016. Software component decision-making: in-house, OSS, cots or outsourcing - a systematic literature review. J. Syst. Softw. 121, 105–124.

[8] J. Linakar., H. Munir., k Wnuk., C.E., 2017. Motivating the contributions: An open innovation perspective on what to share asOSS. J. Syst. Software 135 (2018) 17–36.

[9] Karlsson, L., Dahlstedt, G. , Regnell, B. , och Dag, J.N. , Persson, A. , 2007. Requirements engineering challenges in market-driven software development an interview study with practitioners. Inf. Softw. Technol. 49 (6), 588–604. Qualitative Software Engineering Research.

[10] Dahlander, L., Magnusson, M., 2008. How do firms make use of open source com- munities? Long Range Plann. 41 (6), 629–649.

[11] Aurum, A.,Wohlin, C. , 2007. A Value-Based Approach in Requirements Engineer- ing: Explaining Some of the Fundamental Concepts. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 109–115.

[12] Runeson, P., Höst, M., 2009. Guidelines for conducting and reporting case study re- search in software engineering. Empir. Softw. Eng. 14 (2), 131–164.