Cloud Based Dedicated Application for Multiple Organizations: Cloud Computing for All Organizations

Sheikh Muhammad Saqib

Institute of Computing and Information Technology

Gomal University, PAKISTAN

saqibsheikh4@hotmail.com

Manzoor Elahi

Computer Science

COMSATS Institute of Information Technology, Islamabad, PAKISTAN

tamimy@comsats.edu.pk

Muhammad Ahmad Jan

Institute of Computing and Information Technology

Gomal University, PAKISTAN

mr_ahmadjan@yahoo.com

Shakeel Ahmad

Institute of Computing and Information Technology

Gomal University, PAKISTAN

Shakeel_1965@yahoo.com

Muhammad Zahid Awan

Department of Business Administration

Gomal University, PAKISTAN

Zahidawan1967@gmail.com

Najeebullah Khan

Department of Public Administration

Gomal University, PAKISTAN

najeebullahdpagu@yahoo.com

Abstract

Desktop applications should be run over the cloud environment. This is the slogan of those organization which are familiar with the cloud computing. Organization's members are not only responsible for smooth running of desktop application but also have a burden of successful running of database server, back & recovery devices, time and cost of human efforts. CBCS (Cloud Based Custom Software) is the solution of such issues. Development companies are responsible for development of CBCS of an organization. Although CBCS is a dedicated service to an organization, but existing CBCS can also dedicate to other organization when development companies will follow the CBDAMO (Cloud Based Dedicated Customized Application for Multiple Organization) technique. This is 6-layer technique which provides a smooth way to development companies for reusing the existing CBCS to new CBCS.

Key Words: Cloud Computing, CBCS, DBCS, CBDAMO, Development Companies

1. Introduction

Besides Google App Engine, Microsoft Windows Azure Platform, IDE plug-in of cloud computing, many business organizations have their own software or customized applications [3]. Cloud does not fix to global applications. All stakeholders should be facilitated from cloud with respect to all angles. Private cloud computing may relate to expensive cost but cloud computing usage should be cheap and visible for all type of consumers. Under the umbrella of cloud benefits, CBCS (Cloud Based Custom Application) is the best attitude for removing desktop based application's demerits. It is global fact that CBCS is the replacement of DBCS and such software is dedicated software. Since different organization wants to utilize cloud environment, they should develop CBCS from development companies. Rapid development with minimum cost and maximum access should be the broad concept of development companies. They can achieve the above points by using the technique CBDAMO (Cloud Based Dedicated Customized Application for Multiple Organization). With such adoption dedicated software can be easily accommodate to the other application as shown in Fig-1.
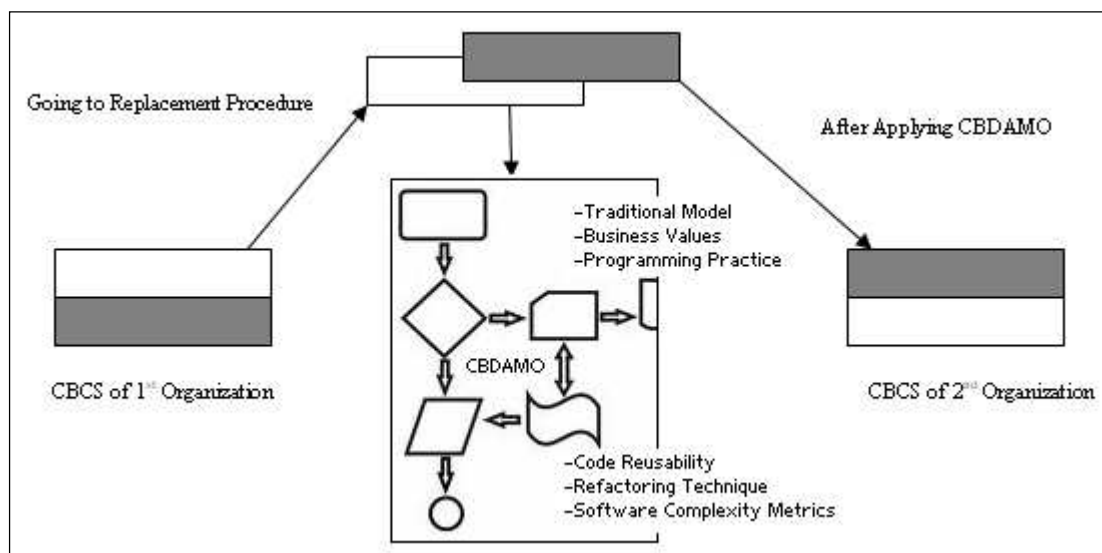

Fig-1: Conversion of CBCS of 1st Organization to Other by using CBDAMO

2. CBCS (Cloud Based Custom Software)

Software contrasted with the use of software packages developed for an organization is known as DBCS (Desktop Based Custom Software) while CBCS (Cloud Based Custom Software) is a committed service of an organization which is run on cloud situation. TA (Tinny Application) restricts the access of CBCS only in relevant organization and provides a mean to high-ups of organization to set their encryption policy. With such adaptation, data on third party spaces can not be readable for irrelevant person [10].

3. CBDAMO (Cloud Based Dedicated Application for Multiple Organization)

CBDAMO is a technique for developing custom software under the power of cloud computing. Fig-2 is showing the work of CBDAMO on each phase of traditional model. Here dropping lines are showing the sequence of CBDAMO on Analysis, Design, Test and Code phase of first increment of incremental model. Such activities will be performed on all increments.
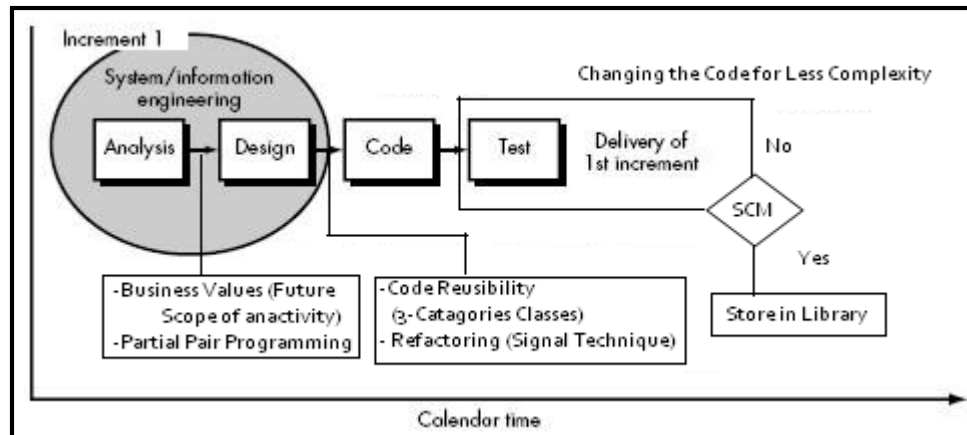
Fig-2: CBDAMO Activities

From above figure it is clear that CBDAMO consists of following activities:

-Selecting Traditional Model

-Business Values for an Activity

-Programming Practice

-Code Reusability

-Refactoring Technique

-Software Complexity Metrics

3.1. Selecting Traditional Model

Without suggesting a road, how and where a vehicle can run? So selection of model is very necessary for development of a system. Since CBCS is customized software, so it will require a traditional model. Since waterfall model does not allow for much reflection or revision, it will not be useful for CBCS [4]. In prototype model, creation of prototype is not a complete software program, so there will be possibility of designer flaws that may leads to losses of important resources [5].Incremental is the advancement of waterfall model. By using such model, working software quickly, early and flexible. Less cost requires for changing the requirements and scope, easier to test & debugs and easier to manage risk [6]. Hence incremental model will be highly suitable for developing CBCS.

3.2. Business Values for an Activity

All requirements are gathered in the analysis phase of incremental model. So before going to design phase, first of all it is very necessary to find out the future scope of each activity. Two questions suggested in "Concentration on Business Values for SOA-Services: A Strategy for Service's Business Values and Scope" should be carried out on each activity after the analysis phase.

-An activity for an organization has same business values as other organization?

-Same activity can have different business values for many organizations?

A template is given in [1], which contains different options for finding the scope of a service. This template can be applied on each activity/module for determining the place of each activity. Theme of this template is used to determine:

-Developed activity can be used in only in an organization, for which CBCS is to be developed.

-Developed activity can also be used in other organization of same type.

-Developed activity can also be used in other organization of different type.

 By getting above information CBCS can be easily developed under the umbrella of all dimensions of same and other organization requirements.

3.3. Programming Practice

A development company develops a CBCS for an organization and such CBCS can be developed for other organization after some years. And possibility exist either programmer which developed such CBCS, have no longer with that company. Then who convert existing CBCS to new CBCS. So solo programming is not required for developing CBCS. Pair programming is may be best for such development but it has its own draw back. B.Ahmad has suggested a best technique for development which contains all benefits of solo programming and pair programming is known as partial pair programming. In such technique two programmers and one navigator will be familiar with the whole code [7]. When three people involve on development of same code, then in future person will be available to Development Company for conversion of existing CBCS to new CBCS for other organization.

3.4. Code Reusability

Coupled code can not be easily reused due to meshing of statements. Content coupling and data coupling can be handled by designing the classes in three categories i.e. Class for database connection, class for executable & and non executable queries and a class for developed activity [8]. Now such type of code can easily accept changes for CBCS of other organization.

3.5. Refactoring Technique

Refactoring is used for maintainability & extensibility. Different technique is used for refactoring. But here since we are using different classes for development of an activity as discuss above then refactoring technique suggested by B.Ahmad will be very suitable for CBCS conversion. Here it is suggested that each class for an activity should be consist of three methods (input, processing & out put). And each method must have a signal for achieving different operation. Such technique of refactoring is very useful for multi dimensional reusability [9]. When input output and processing procedures have separate methods then graphical user interface, input interface and output interface of CBCS of an organization can be converted to CBCS of other organization.

3.6. Software Complexity Metrics

Degree to which a component a plan that is difficult to understand and verify is known as software complexity. There is a direct relationship between complexity of code and understandability. Factors which make a code difficult to understand are the weight of complexity. Common code complexity measures are of two types such as McCabe metric and Halstead's. Code complexity determine by McCabe metric depends upon no of control path created by the code. Halstead approach depends on the mathematical interaction among the number of variables, code complexity and the type of programming language statements [2]. If complexity is higher then developer will reset the code to make simple before placing in library. CBCS of an organization with less complex code can be easily changeable for CBCS of other organization.

4. Conclusion

CBCS will be the future trend for all organization due to achieving all merits of cloud computing. All organization will replace their DBCS by developing their CBCS. CBDAMO layers for development of CBCS is very useful for development companies i.e. when they will develop a CBCS, then such CBCS can also be reusable for other organization. A.Jan [10] suggested a TA application for CBCS, which protect CBCS of an organization from all possible security threats. So development companies can develop a single CBCS by using incremental model. Partial pair programming practice is very beneficial for development, because it contains all benefits of pair programming and solo programming. Due to such technique development companies can have many developers of same activity. Creation of three level classes through three section methods with passing of a signal is a best solution for making developed CBCS for one organization to other.

References

Bashir Ahmad, Concentration on Business Values for SOA-Services: A Strategy for Service's Business Values and Scope, (IJCSIS) International Journal of Computer Science and Information Security, Vol. 9, No. 5, May 2011.

Kamaljit Kaur, Static and Dynamic Complexity Analysis of

Software Metrics, World Academy of Science, Engineering and Technology 56 2009

Interview of cloud computing expert Janakiram MSV, Refactoring Is The First Step In A Cloud Migration , Feb 2011.

Advantages and Disadvantages of WaterFall Model, IT ACUMENS, October 15, 2008 http://www.waterfall-model.com

SDLC - Prototype Model, GreekInterview, 10th Jan 2008, http://www.learn.geekinterview.com/it/sdlc/prototype-model.html

Incremental lifecycle model ,SoftDevTeam, http://www.softdevteam.com/Incremental-lifecycle.asp

Bashir Ahmad, Partial Pair Programming: Link between Solo and Pair Programming, Innovative Systems Design and Engineering ISSN 2222-1727 (Paper) ISSN 2222-2871 (Online) Vol.2, No.3, August 2011

Shakeel Ahmad, Reusable Code for CSOA-Services: Handling Data

Coupling and Content Coupling, (IJCSIS) International Journal of Computer Science and Information Security, Vol. 9, No. 5, May 2011

Bashir Ahmad, Refactoring for Multi-Dimensional Reusability, Innovative Systems Design and Engineering ISSN 2222-1727 (Paper) ISSN 2222-2871 (Online) Vol 2, No 4, September 2011

Muhammad Ahmad Jan, Extension of Tinny Application for Cloud Based Custom Software with Encryption Queue: A Strategy to Protect Data on Cloud, Innovative Systems Design and Engineering ISSN 2222-1727 (Paper) ISSN 2222-2871 (Online) Vol 2, No 3, August 2011