RESEARCH ARTICLE OPEN ACCESS

Cost Estimation Based On Functional Point for Exchange Server Mailbox Hide

[1]Saoud Sarwar And [2]Rajendra Kumar Sharma

[1]*Research Scholar, Department of Computer Science & Engineering Bhagwant University, Ajmer, Rajasthan*
[2]*Professor, Department of Computer Science & Engineering Bhagwant University, Ajmer, Rajasthan*

ABSTRACT

In the corporate world, the major challenges to estimate the project cost to avoid any loose in the project and also submit the project on time. The costing of projects or design of the algorithmic to estimating the cost of a software project based on functional point is the best practice and approach. Accurate estimation of software production cost for effective management of the project including budgeting, controlling, and programming is so important. In recent years, software has become the most expensive part of the computer projects. Some of these costs are resulted from human resource effort. The speed in which the processes used in software development field have changed makes it very difficult the task of forecasting the overall costs for a software project. By many researchers, this task has been considered unachievable but the techniques that based on functional point methodology and heuristic knowledge of the software development architecture, we can estimate the cost of the project with + 5 % to -4 % variation.

Keyword: Cost estimation, effort, functional point, EI, EO, EQ, ILF, EIF.

Date of Submission: 25-08-2017 Date of acceptance: 09-09-2017

Practical use cases for the cost estimation for Microsoft Exchange server hide and unhide mail account from global address list (GAL)

When user is on boarding then checking the business logic for Mailbox assignment
If user`s come under the criteria for Mailbox assignment then check the user`s effective start date
if
User`s Effective Start date >= “sysdate +2”
then assigned the Mailbox to user at Exchange 2010 and hide the mailbox at exchange 2010
and insert the entry in the table on the schema of CDI DB with a flag as false.
If
Effective start date > “sysdate +2”
then only hide the user`s mailbox account. Run the process to check the effective start date < “sysdate -2” in the table of the schema of the CDI DB and get all the accounts and then execute the process to UNHIDE the mailbox before un-hiding the mailbox, check the user`s account already hide via some other process at Exchange 2010.
Once UNHIDE the mailbox at Exchange 2010 then set the true flag in the DB.

I. INTRODUCTION

This paper presents a cost estimation based on functional point. A practical use cases for the cost estimation for Microsoft Exchange server hide and unhide mail account from global address list (GAL).

Responsibility of the project manager is to have accurate estimates of effort, cost, and schedule involved in software development. There is major relation and impact based on accurate estimates of effort, cost, and schedule or timeline or delivery get failed, exceed budget and go over scheduled or bottleneck on work environment[19] [21]. Software effort and cost estimation is the set of techniques and procedures that organize an estimate for proposal bidding, project planning and probability estimates. Accurate effort and cost estimation means better planning and efficient use of project resources such as cost, duration and effort requirements for software projects.

Function point methodology with an enhanced approach, focus on type of the domain of the software projects. The assessment of a project based on functional and non functional requirements should be the first step for a project manager before going to find the cost or effort of the project [22].

The perspective of model described it linked with the current reality of the software development considering as basis the software product life cycle and the current challenges and innovations in the software development area [17]. Based on the author's experiences and the analysis of the existing models and product life cycle it was concluded that estimation models should be adapted with the new technologies and emerging systems and they depend largely by the chosen software development method.

II. CASE STUDY – HIDE & UNHIDE MAILBOX:

Flow Diagram of the requirement, mentioned in fig. 1 and fig. 2.

1) GE mail provisioning for employees.
2) Mail provision to Exchange 2010 (windows active directory).
3) Insert in to CDI DB according to the Effective Start date.
4) Add: msExchHidefromAddressLists attribute in the Microsoft Exchange 2010 server [3].
5) Table : cdi1_mailboxhide , Insert a row with employees effective start date and create date

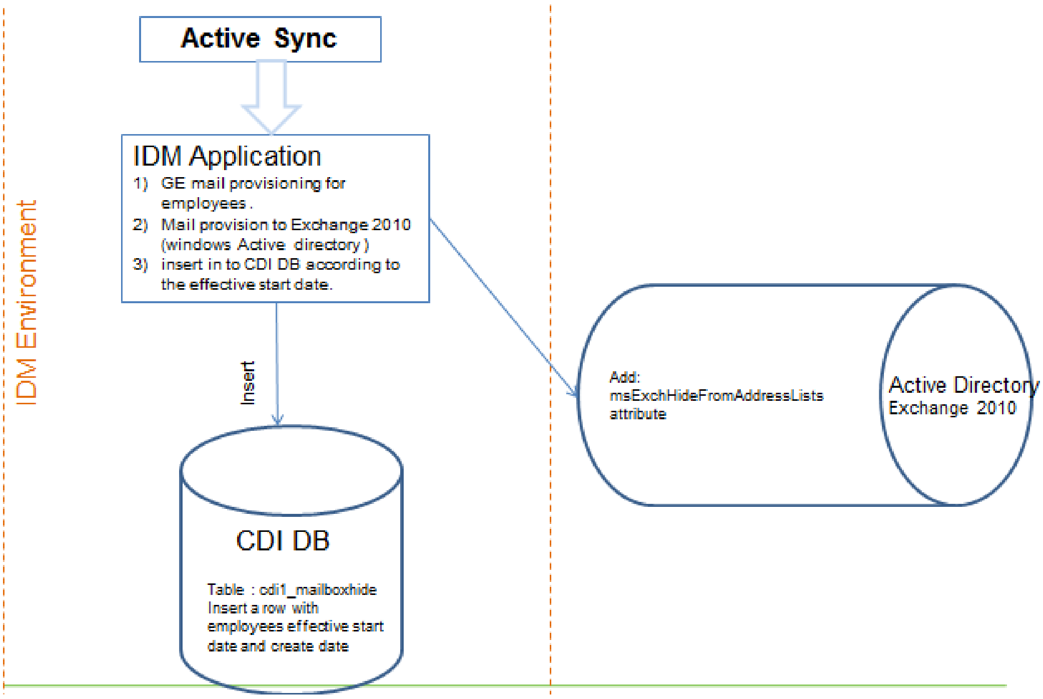

Active Sync
IDM Environment
IDM Application
1) GE mail provisioning for employees.
2) Mail provision to Exchange 2010 (windows Active directory)
3) insert in to CDI DB according to the effective start date.
Insert
CDI DB
Table : cdi1_mailboxhide
Insert a row with employees effective start date and create date
Add: msExchHideFromAddressLists attribute
Active Directory
Exchange 2010

Fig.1. Flow diagram of the requirement

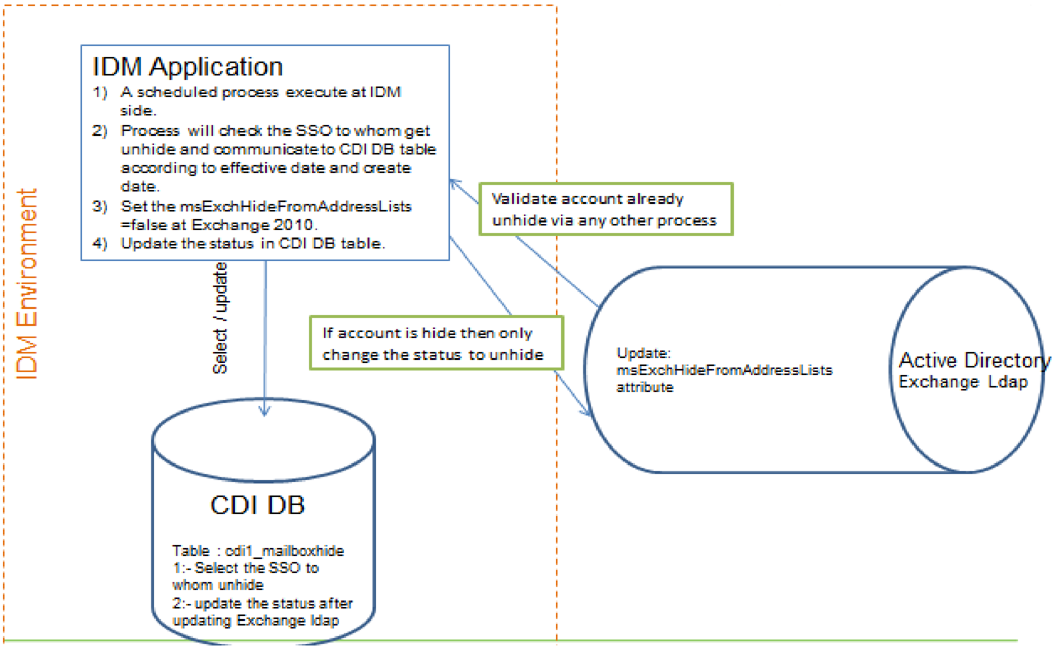

IDM Environment
IDM Application
1) A scheduled process execute at IDM side.
2) Process will check the SSO to whom get unhide and communicate to CDI DB table according to effective date and create date.
3) Set the msExchHideFromAddressLists =false at Exchange 2010.
4) Update the status in CDI DB table.
Validate account already unhide via any other process
Select / update
If account is hide then only change the status to unhide
CDI DB
Table : cdi1_mailboxhide
1:- Select the SSO to whom unhide
2:- update the status after updating Exchange ldap
Update: msExchHideFromAddressLists attribute
Active Directory
Exchange Ldap

Fig.2. Flow diagram of the requirement

III. PROCESS MAP OF THE REQUIREMENT:

1. Employee On boarded on OHR System.
2. IDM will create the new user.
3. At mail provisioning IDM process will check the effective start date and create date.
4. If effective start date is greater than create date +2?
5. Provision the mail box and set the attribute msExchHideFromAddressLists=TRUE in Exchange 2010 [4].
6. Provision the mail and exit from process.
7. An insert will fire on the CDI DB Table: cdi1_mailboxhide_t with effective start date, create date and set update status = false.
8. Sync process start and communicate the CDI DB Table cdi1_mailboxhide.
9. Is there any SSO having effective start date equal to current date + 2?
10. Fetch the SSO LIST.
11. Is there next SSO?
12. If already updated, end the process.
13. Set attribute msExchHideFromAddressLists =FALSE in Exchange 2010 for each SSO.
14. Update the CDI DB Table: cdi1_mailboxhide_t with set update status = true.
15. End The Process.

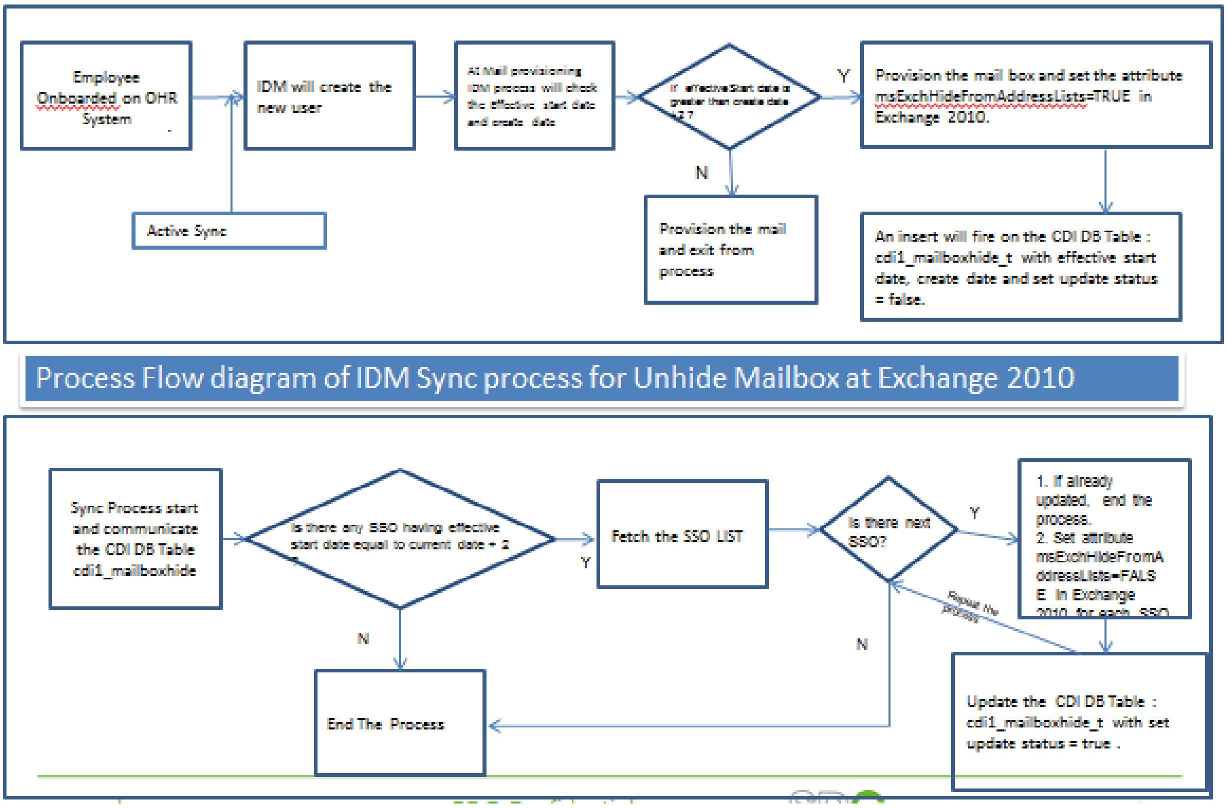

Employee Onboarded on OHR System
IDM will create the new user
At Mail provisioning IDM process will check the effective start date and create date
If effective Start date is greater than create date +2 ?
Y
Provision the mail box and set the attribute msExchHideFromAddressLists=TRUE in Exchange 2010.
N
Active Sync
Provision the mail and exit from process
An insert will fire on the CDI DB Table : cdi1_mailboxhide_t with effective start date, create date and set update status = false.
Process Flow diagram of IDM Sync process for Unhide Mailbox at Exchange 2010
Sync Process start and communicate the CDI DB Table cdi1_mailboxhide
Is there any SSO having effective start date equal to current date + 2 ?
Y
Fetch the SSO LIST
Is there next SSO?
Y
1. If already updated, end the process. 2. Set attribute msExchHideFromAddressLists=FALSE In Exchange 2010 for each SSO
Repeat the process
N
N
End The Process
Update the CDI DB Table : cdi1_mailboxhide_t with set update status = true .

Fig.3. Process flow diagram

IV. STEP WISE FUNCTION POINT CALCULATION

In a function point analysis, the following features are considered:

4.1 External inputs (EI): A method by which data crosses the edge of the system. Data may be used to revise one or more logical files. It may be eminent that data here means either industry or control information [1]. This means EI is known as a transaction function in which data goes "into" the application from outside the boundary to inside. Here data is coming external to the application [6].

- Data may arrive from data input monitor or from other source.
- An application gets in sequence via an EI.
- Information data can be either organize or business information.
- One or more internal logical files can be maintained using data.
- Internal logical file does not update if the data is control information.

4.2 External outputs (EO): A method by which data crosses the edge of the system to outside of the structure. It can be a user report or a system log report [20]. In the case of EO, data is received "out" of the system. Also, an EO can update an ILF. The data creates information or output files sent to other applications [18].

4.3 External user inquires (EQ): A count of the method in which both input and output outcome in data rescue from the system. These are essentially system inquiry processes [15]. It is also a business function having both input and output components. EQ is essential in data retrieval [14].

4.4 Internal logical files (ILF): In ILF manages information or user identifiable group of logically connected data reside totally within the application boundary. ILF holds data that are maintained through one or several basic processes. An ILF has the inherent meaning that it is internally maintained, it has some rational structure and it is stored in a file [12] [13]. This means a group of logically correlated data files that resides completely within the boundary of the application software and is maintained through outer input [9].

4.5 External interface files (EIF): EIF is data or control information which is used by the application for mention purposes only. Here the data, resides completely outside the application boundary, and is maintained in an ILF by another application. It has the natural meaning that it is externally maintained, a boundary has to be developed to get the data from the file [8]. This means a group of reasonably related data files that are used by the system for mention purposes only. These data files remain totally outside the application boundary and are maintained by external applications [7] [11].

4.6 Parameter weight table

Parameter	Low	Average	high
ILF	7	10	15
EIF	5	7	10
EI	3	4	6
EO	4	5	7
EQ	3	4	6

Complexity tables
FOR ILF AND EIF

	1 to19 DET	20 to50 DET	51 or more DET
1 RET	Low	Low	Average
2 to 5 RET	Low	Average	High
6 or more RET	Average	High	High

FOR EI

	1 to 4 DET	5 to 15 DET	16 or more DET
0 to 1 FTR	Low	Low	Average
2 FTRs	Low	Average	High
3 or more FTRs	Average	High	High

EO AND EQ

	1 to 5 DET	6 to 19 DET	20 or more DET
0 to 1 FTR	Low	Low	Average
2 to 3 FTRs	Low	Average	High
4 or more FTRs	Average	High	High

V. Function point Calculation

❖ When user is On boarding, then checking the business logic for mailbox assignment.

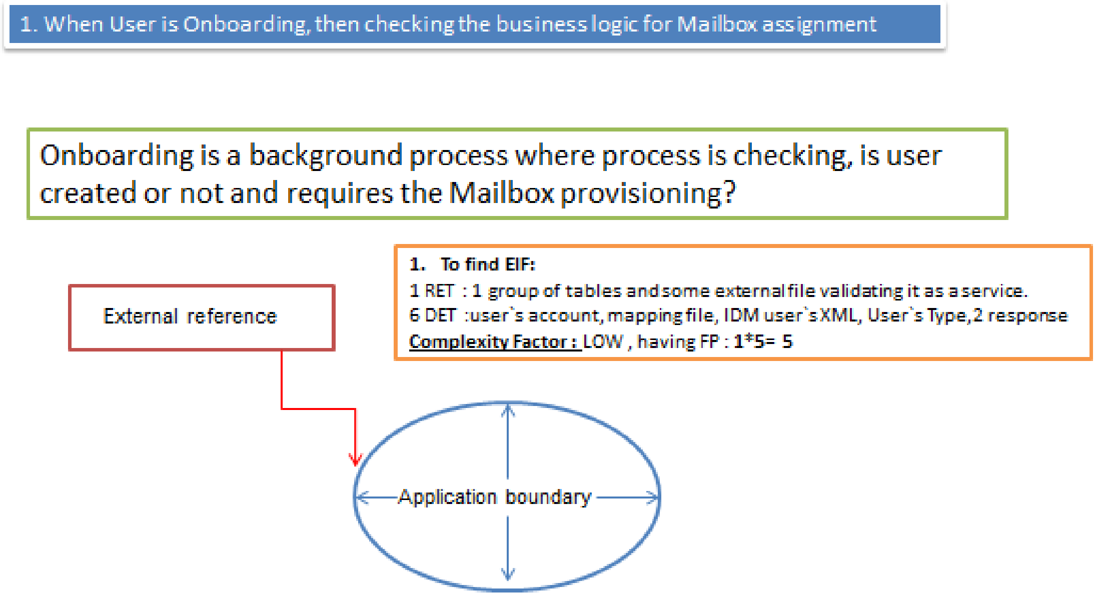
On boarding is a background process where process is checking, is user created or not and requires the mailbox provisioning?

- **To find EIF:**

1 RET : 1 group of tables and some external file validating it as a service.

6 DET :user`s account, mapping file, IDM user`s XML, User`s Type,2 response

Complexity Factor: LOW, (FP : 1*5= 5)



1. When User is Onboarding, then checking the business logic for Mailbox assignment

Onboarding is a background process where process is checking, is user created or not and requires the Mailbox provisioning?

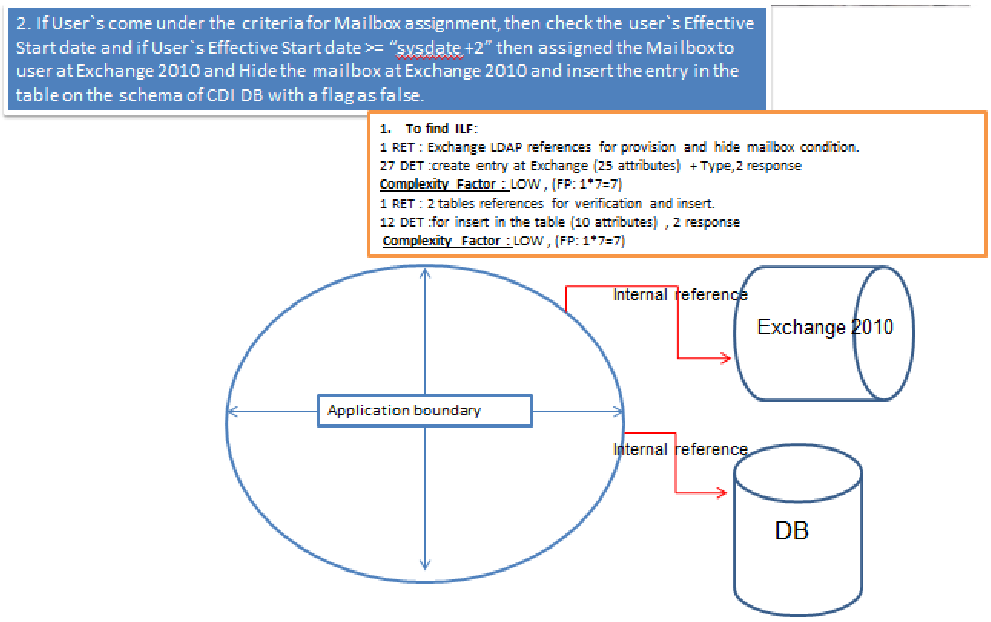
1. **To find EIF:**
1 RET : 1 group of tables and some external file validating it as a service.
6 DET :user`s account, mapping file, IDM user`s XML, User`s Type,2 response
Complexity Factor : LOW , having FP : **1*5= 5**

Fig.4. User on boarding

- If User`s come under the criteria for Mailbox assignment, then check the user`s Effective Start date and if user`s Effective Start date >= “sysdate +2” then assigned the Mailbox to user at Exchange 2010 and Hide the mailbox at Exchange 2010 and insert the entry in the table on the schema of CDI DB with a flag as false.

- **To find ILF:**

1 RET: Exchange LDAP references for provision and hide mailbox condition.
27 DET: Create entry at Exchange (25 attributes) + Type, 2 response.
Complexity Factor: LOW, (FP: 1*7=7)
1 RET: 2 tables references for verification and insert.
12 DET: For insert in the table (10 attributes), 2 response.
Complexity Factor: LOW, (FP: 1*7=7)



2. If User`s come under the criteria for Mailbox assignment, then check the user`s Effective Start date and if User`s Effective Start date >= “sysdate +2” then assigned the Mailbox to user at Exchange 2010 and Hide the mailbox at Exchange 2010 and insert the entry in the table on the schema of CDI DB with a flag as false.

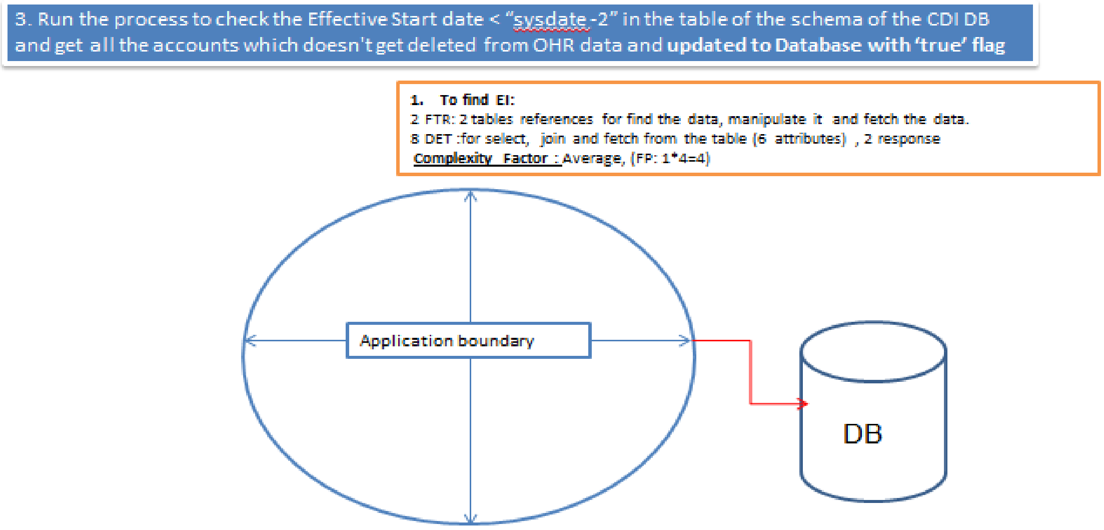
1. **To find ILF:**
1 RET : Exchange LDAP references for provision and hide mailbox condition.
27 DET :create entry at Exchange (25 attributes) + Type,2 response
Complexity Factor : LOW , (FP: 1*7=7)
1 RET : 2 tables references for verification and insert.
12 DET :for insert in the table (10 attributes) , 2 response
Complexity Factor : LOW , (FP: 1*7=7)

Fig.5. Mailbox assignment

- ❖ Run the process to check the Effective Start date < "sysdate -2" in the table of the schema of the CDI DB and get all the accounts which doesn't get deleted from OHR data and updated to Database with 'true' flag.

- ▪ **To find EI:**

2 FTR: 2 table's references for find the data manipulate it and fetch the data.
8 DET: For select, join and fetch from the table (6 attributes) , 2 response
Complexity Factor: Average, (FP: 1*4=4)



3. Run the process to check the Effective Start date < "sysdate -2" in the table of the schema of the CDI DB and get all the accounts which doesn't get deleted from OHR data and **updated to Database with 'true' flag**

1. **To find EI:**
2 FTR: 2 tables references for find the data, manipulate it and fetch the data.
8 DET :for select, join and fetch from the table (6 attributes) , 2 response
Complexity Factor : Average, (FP: 1*4=4)

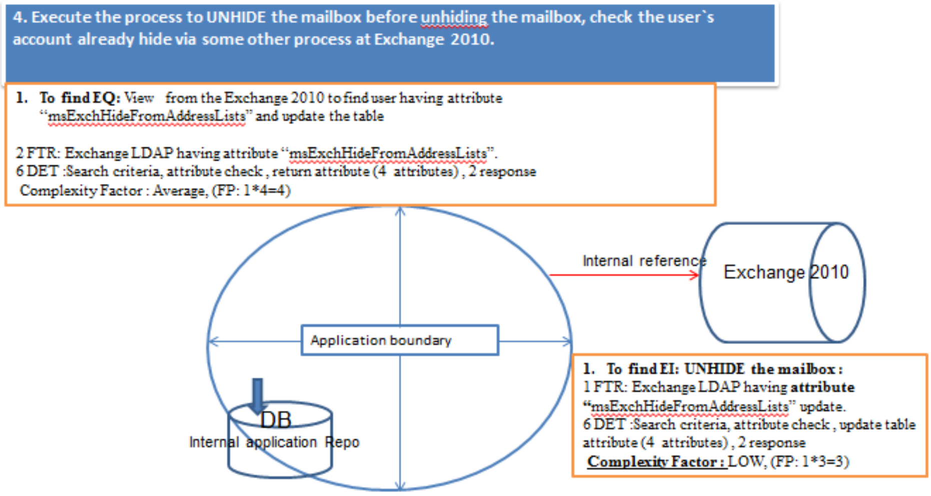
Fig.6. Running the process

- ❖ Execute the process to UNHIDE the mailbox before unhiding the mailbox, check the user`s account already hide via some other process at Exchange 2010.
- ▪ **To find EQ:** View from the Exchange 2010 to find user having attribute "msExchHideFromAddressLists" and update the table.

2 FTR: Exchange LDAP having attribute "msExchHideFromAddressLists".
6 DET: Search criteria, attribute check, return attribute (4 attributes), 2 response
Complexity Factor: Average, (FP: 1*4=4)

- ▪ **To find EI: UNHIDE the mailbox :**

1 FTR: Exchange LDAP having attribute "msExchHideFromAddressLists" update.
6 DET: Search criteria, attribute check, update table attribute (4 attributes), 2 response
Complexity Factor: LOW, (FP: 1*3=3)



4. Execute the process to UNHIDE the mailbox before unhiding the mailbox, check the user`s account already hide via some other process at Exchange 2010.

1. **To find EQ:** View from the Exchange 2010 to find user having attribute "msExchHideFromAddressLists" and update the table

2 FTR: Exchange LDAP having attribute "msExchHideFromAddressLists".
6 DET :Search criteria, attribute check , return attribute (4 attributes) , 2 response
Complexity Factor : Average, (FP: 1*4=4)

1. **To find EI: UNHIDE the mailbox :**
1 FTR: Exchange LDAP having **attribute** "msExchHideFromAddressLists" update.
6 DET :Search criteria, attribute check , update table attribute (4 attributes) , 2 response
Complexity Factor : LOW, (FP: 1*3=3)

❖ **Fig.7.** Execute the process

- Send the email to support team after completion of the Sync process with the status of the user`s success and failure (if failure then error msg).
- **To find EI:** Send the notification to DL:

1 FTR: set the results of the attributes updates.
8 DET: Set the variables at the email and notify.
Complexity Factor: LOW, (FP: 1*3=3)

VI. CONCLUSION

Today, almost no model can estimate the cost of software with a high degree of accuracy. This state of the practice is created because:
(1) There are a large number of interrelated factors that influence the software development process of a given development team and a large number of project attributes, such as number of user screens, volatility of system requirements and the use of reusable software components.
(2) The development environment is evolving continuously.
(3) The lack of measurement that truly reflects the complexity of a software system.
To produce a better estimate, we must improve our understanding of these project attributes andtheir causal relationships, model the impact of evolving environment and develop effective ways of measuring software complexity.

Based on these approaches we have categories the do requirements and collected the different software developed projects from different streams, we evaluated the function point and match with the actual effort based on project Kickoff meeting started and the project closure submitted.

Function point methodology with an enhanced approach, focus on type of the domain of the software projects. The assessment of a Project based on functional and non functional requirements should be the first step for a project manager before going to find the cost or effort of the project.

Total Function point calculation for the above mentioned use case.

1. When user is On boarding, then checking the business logic for Mailbox assignment : 1 EIF = 5
2. If User`s come under the criteria for Mailbox assignment, then check the user`s Effective Start date and if user`s Effective Start date >= “sysdate +2” then assigned the Mailbox to user at Exchange 2010 and Hide the mailbox at Exchange 2010 and insert the entry in the table on the schema of CDI DB with a flag as false: 2 ILF = 14
3. Run the process to check the Effective Start date < “sysdate -2” in the table of the schema of the CDI DB and get all the accounts which doesn't get deleted from OHR data: 1 EI = 4
4. Execute the process to UNHIDE the mailbox before unhiding the mailbox, check the user`s account already hide via some other process at Exchange 2010: 1 EI +1 EQ = 3 +4 =7
5. Send the email to IDM support team after completion of the Sync process with the status of the user`s success and failure (if failure then error msg): 1 EI = 3

Total: 5+14+4+7+3 = 33

REFERENCES

[1]. Goether, Wolfhart B., Elizabeth K. Bailey, Mary B. Busby, Software Effort and Schedule Measurement: A framework for counting Staff-hours and reporting Schedule Information, CMU/SEI-92-TR-021, 1992.
[2]. https://technet.microsoft.com/en-us/library/ff381467.aspx.
[3]. https://social.technet.microsoft.com/wiki/contents/articles/30926.exchange-2013-hide-mailbox-from-address-list.aspx.
[4]. .https://social.technet.microsoft.com/Forums/exchange/en-US/7fef86b6-04af-4c1a-98ec-24dd99e6854a/hiding-users-from-global-address-list-in-exchange-2010?forum=exchangesvradminlegacy.
[5]. https://support.microsoft.com/en-in/help/924297/how-to-delete-corrupted-and-hidden-rules-from-a-single-mailbox-in-out.
[6]. Humphrey, Watts, A Discipline for Software Engineering, Addison-Wesley, 1995.
[7]. Dreger, Brian, Function Point Analysis, Prentice-Hall, 1989.
[8]. Garmus, David & David Herron, Measuring the Software Process, Yourdon Press, 1996.
[9]. Jones, Capers, Applied Software Measurement: Assuring Productivity and Quality, McGraw-Hill, 1991.
[10]. Jones, Capers, What are Function Points, 1997, http://www.spr.com/library/0funcmet.htm.
[11]. International Function Point Users Group (IFPUG) web site: http://www.ifpug.org
A function point FAQ: http://ourworld.compuserve.com/homepages/softcomp/fpfaq.htm.
[12]. C.F. Kemerer, “An empirical validation of software cost estimation models”, Communications of the ACM, vol. 30, no. 5, May 1987, pp. 416-429.
[13]. S. S. Vicinanza, T. Mukhopadhyay, and M. J. Prietula, “Software-effort estimation: an exploratory study of expert performance”, Information Systems Research, vol. 2, no. 4, Dec.1991, pp. 243-262.
[14]. D. V. Ferens, and R. B. Gumer, “An evaluation of three function point models of

estimation of software effort", IEEE National Aerospace and Electronics Conference, vol. 2, 1992, pp. 625- 642.

[15]. G. R. Finnie, G. E. Wittig, AI tools for software development effort estimation, Software Engineering and Education and Practice Conference, IEEE Computer Society Press, pp. 346-353, 1996.

[16]. DeMarco, Tom, Controlling Software Projects, Prentice-Hall, 1982.

[17]. F. J. Heemstra, "Software cost estimation", Information and Software Technology, vol. 34, no3.

[18]. D. R. Jeffery, and G. C. Low, "Calibrating estimation tools for software development", Software Engineering Journal, July 1990, pp. 215-221.

[19]. J. Hihn and H. Habib-Agahi, "Cost estimation of software intensive projects: a survey of current practices", International Conference on Software Engineering, 1991, pp. 276-287.

[20]. D. R. Jeffery, G. C. Low, and M. Barnes, "A comparison of function point counting techniques", IEEE Trans on Soft. Eng., vol. 19, no. 5, 1993, pp. 529-532.

[21]. M. Shepperd and C. Schofield, "Estimating software project effort using analogy", IEEE Trans. Soft. Eng. SE-23:12, 1997, pp. 736-743.

[22]. L. C. Briand, K. El Eman, F. Bomarius, "COBRA: A hybrid method for software cost Estimation, benchmarking, and risk assessment", International conference on software Engineering, 1998, pp. 390-399.

International Journal of Engineering Research and Applications (IJERA) is **UGC approved** Journal with Sl. No. 4525, Journal no. 47088. Indexed in Cross Ref, Index Copernicus (ICV 80.82), NASA, Ads, Researcher Id Thomson Reuters, DOAJ.

Saoud Sarwar. "Cost Estimation Based On Functional Point for Exchange Server Mailbox Hide." International Journal of Engineering Research and Applications (IJERA) , vol. 7, no. 9, 2017, pp. 13–20.