

Exploring reuse levels in ERP projects in search of an effort estimation approach

Onur Demirörs^{1,2}, Neslihan Küçükateş Ömüral³

¹ Computer Engineering Department, Izmir Institute of Technology, Izmir, Turkey

² School of Computer Science and Engineering, University of New South Wales, Sydney, Australia

³ Informatics Institute, Middle East Technical University, Ankara, Turkey

Abstract—Enterprise Resource Planning (ERP) projects have special characteristics that differentiate them from other kinds of software projects. Main difference that affects effort estimation results is high reuse rates of ERP projects. This paper presents our exploratory work to establish an approach to calculate reuse reflective size of ERP projects, which could be a primary input for effort estimation. We explored the usability of COSMIC function points as the base unit and its convertibility to reuse reflective size using reuse levels. We have performed a case study on an SAP Implementation project. We were able to calculate COSMIC size and reuse levels using the documents available for the project. We discuss the results as well as challenges and opportunities in the light of the case study.

Keywords- enterprise resource planning, effort estimation, function points, reuse

I. INTRODUCTION

ERP Systems are one of the most complicated and effort intensive information system solutions adopted by a variety of organizations. There are studies to identify factors effecting effort of ERP projects, studies where existing effort estimation models are tested by case studies and studies where new effort estimation methods are suggested specifically for ERP. However, none of the methods is widely accepted and validated as an appropriate method for ERP effort estimation [3]. ERP Effort Estimation is still a major problem for ERP project management as evidenced by duration and cost overruns.

ERP Systems have pre-built functionalities that mostly fulfill customer requirements. From a software development perspective it can be considered as reusing most of the pre-built functionality. We assume that as reuse dominates the development, reuse rates should be reflected in effort estimation for proper results. Based on this assumption we targeted to develop an approach for ERP effort estimation using reuse reflective size of the projects. To establish such an approach, we first need to calculate the size of the project and then we need to calculate the amount of reuse. As a first step, we performed a case study to understand if COSMIC function points can be used as a size measure and if we can calculate reuse levels by analyzing available documents as a

base of such an approach. To explore the potential, we performed a case study on a completed SAP Implementation project. We calculated COSMIC functional size and reuse reflective size for this project using the business blueprints. We have also applied the calculated size in an existing ERP effort estimation method and compare the results with the actual effort.

The rest of the paper is organized as follows: In Section 2, ERP function point based effort estimation and reuse studies published in the literature are summarized. In Section 3, we explain our targeted approach to calculate reuse reflective size and effort for ERP projects. The description of the case study project, the application of our approach and the results obtained are presented in Section 4. In Section 5, we presented conclusions, validity threads and suggested directions for future work.

II. BACKGROUND

ERP projects frequently suffer from high effort and cost overruns; yet there are only few research studies for resolving the problems. We performed a systematic literature review [3] to investigate effort estimation methods suggested for ERP projects by discussing their validity and limitations. One of the findings of our study was that in most of the effort estimation methods, function points were used as input of the model. “Business Blueprint” [14] covering the scope and the requirement specifications of the ERP projects was the main source for Function Point Analysis (FPA).

There are a few studies in the literature explicitly performing FPA for ERP projects [3]. Téllez [11] proposed a FPA based size estimation method estimating COSMIC function points by using business blueprint document and extended EPC diagrams. Second study exploring function point based size estimation was [7]. At that study, Kujipers proposed a method called “Automated FPA method” to calculate IFPUG function points automatically based on a repository.

Studies [8], [9] and [10] were the only studies explicitly suggesting ERP effort estimation methods based on size as function points. In [8], Erasmus introduced the effort estimation method called “COSMIC EPC”. In this method, business processes and related process steps are figured out by analyzing business blueprint document; effort is estimated

using COSMIC function points and conversion factors. Vogezang [9] developed effort estimation method called “refined approximate COSMIC-FFP” which is also capable of early estimates for ERP projects. Pierre and Daneva depicted that it is not possible to use same effort estimation method for every situation in ERP projects; instead they suggested using estimation strategies integrating FPA and Expert Judgments [10].

We performed a case study [5] to explore existing FPA based effort estimation methods namely [8], [9] and [10]. In that study, we observed that business blueprint documents could be a valuable source for FPA and FPA based effort estimation methods can be properly applied for ERP projects. The most critical parameters in these methods are modification and reuse levels; those parameters should be defined and calculated precisely. As the challenges related with the reliability of measurements are more under control, COSMIC is the most frequently used technique for developing estimation models for ERP based systems [18, 19]

Daneva [4] investigated function point based reuse measurement for ERP projects. She defined reuse for ERP projects as “adopted requirements in the client organization’s ERP project” and categorized reuse in three main levels. For three SAP projects, function points are calculated for all reuse levels to investigate how much reuse is achieved in those projects.

Santillo & Abran [12] developed an approach called “functional similarity” to define reuse from a functional perspective. They identified and quantified functional reuse for a set of case studies based on functional information collected as COSMIC function points. The method in this study was mainly for evaluating functional similarities; it did not define a relation between functional size and development effort. Then, studies [1] and [17] performed to develop an approach for the relation of functional size and development effort. Authors used the functional similarity concept for calculating functional similarity reflective size and effort in those studies. Beside these studies, Top, Tunalilar and Demirors performed a case study [2] to explore applicability of functional similarity concept and impact of functional similarity on effort values. One of the main conclusions of this study was that if productivity is calculated based on functional similarity reflective size, productivity values get closer to each other for different projects. They recommended performing further research to support this relation.

III. THE APPROACH

ERP is a growing industry with new products, solutions according to customers’ expectations. In this evolving environment, producing solutions that costs millions of dollars, it does not seem feasible to rely on analogy based effort estimation methods. Instead, we need to work on more precise effort estimation methods. One such approach is to use size as direct input as is the case for most engineering fields. In this case the size should be easily and consistently measurable. Based on our systematic literature review study [3], “Function Points” was the most common input

parameter used in ERP effort estimation studies. FPA can be measured once the requirements are known, even there are estimation methods to be used in earlier phases and based on business processes. Furthermore, FP has many uses in the software project management processes as monitoring scope change, early size estimation, normalizing performance and quality measures [13]. Thus, we assumed FPA could be the most likely size measure for a potential ERP effort estimation method.

Reuse is one of the most critical parameters that should be taken into account for ERP effort estimation. In an ERP Implementation project, most of the customer requirements can be fulfilled by ERP pre-built functionalities; in some cases, reuse rates of 80% are possible [4]. Thus, we might figure out the relation between reuse levels and effort estimates, and design an effort estimation method that accounts for relative cost of reuse.

Therefore we suggest that an approach can be built primarily on two variables: first we need the FP count and secondly we need to calculate the amount of reuse. Business Blueprint Documents could be the main source for FPA calculation, since it contains all details related to customer requirements. A Business Blueprint Document has mainly three parts as “Organizational Units”, “Master Data” and “Business Scenarios”. Especially by analyzing “Business Scenarios” part, one can list business scenarios and processes that will be implemented in the project, and requirements related to changes in business processes.

In ERP project management tools business scenarios are defined that includes related business processes and process steps. For an SAP Implementation Project, one can define all included business processes and process steps for a project by analyzing Business Blueprint Document and SAP Solution Manager Business Process Repository. We assume that we can calculate size of the project as COSMIC function points by using these business documents.

We have used reuse levels defined and empirically validated by Daneva [4]. She classified ERP reuse in four levels in her study:

- *Level 3*: In this reuse level, process and data components are completely reused. There is no need to perform any kind of change in this reuse level.
- *Level 2*: It refers minor enhancements that are applied to process and data components. “Minor enhancement” means changing a certain parameter of a business process or data component. This parameter change should not cause a change in the process logic.
- *Level 1*: This category refers to major enhancements applied to processes and data components. “Major enhancement” means conceptual level changes applied in process logic; the definition of the process or data component is modified at code level.
- *No Reuse*: In fact, this is not a real reuse level; this level refers completely new developments, reuse is not applied at all.

With this reuse level categorization, we can define reuse levels at process step levels based on the Business Blueprint

Documents. However, we need a method to calculate functional size of the project by taking into account these reuse levels, we need to reflect reuse rate in size calculation. Top, Demirors and Turetken [1] developed a method to calculate similarity reflective functional size which could be used for effort estimates. This similarity reflective functional size methodology could be applicable for ERP projects' effort estimation; main difference is we do not need to calculate internal (within project) reuse, instead we need to calculate external reuse, i.e. how much pre-built functionality of ERP will be used in the project.

In that study [1], two types of functional similarity reflective size calculation methods are introduced; continuous functional similarity reflective size (CS) and discrete functional similarity reflective size (DS). The discrete similarity reflective functional sizes are calculated by using constant functional similarity percentage values which were extracted from software enhancement approach of NESMA FSM method [20]. In NESMA, functional size is multiplied by an impact factor depending on amount of change. For DS calculation, amount of change of NESMA is considered as amount of functional similarity and impact factors as functional similarity percentage constants. We adapted this reflective size estimation method to ERP by using ERP reuse levels. We defined following functional similarity percentage constants for calculation.

TABLE I. SIMILARITY PERCENTAGE CONSTANTS

Reuse Level	Reuse Level Description	Highest Similarity Percentage Value	Functional Similarity Percentage Constants
Level1	major enhancements applied to reference processes and data components	max<=34 %	0,25
Level2	minor enhancements applied to reference processes and data components	0,67<max<1,0	0,75
Level3	process and data components that were reused without any changes	max=1,0	0,1
no-reuse	new development	max=0	1

We calculate reuse reflective size of an ERP project in three steps:

1. Measurement of functional size of the product with COSMIC (CFP)
2. Identification of reuse levels
3. Determination of reuse reflective size (RRS)

We perform these three steps for all business processes in business process structure of the project. Reuse Reflective Size calculation formulas, which are mainly adapted from DS, are as follows:

- *Reuse Level3*
RRS = CFP*reuse overhead

- *Reuse Level2*
RRS = (CFP*0.25) + (CFP*0.75*reuse overhead)
- *Reuse Level1*
RRS = (CFP*0.75) + (CFP*0.25*reuse overhead)
- *No-reuse*
RRS = CFP

For calculation of reuse reflective size, we also need to identify “reuse overhead” value for ERP. There are no studies for identifying reuse overheads in ERP projects. As a first attempt, we use reuse overhead as “0.10” as given in the DS methodology.

After calculation of reuse reflective size for all business processes, we sum up with total reuse reflective size of the project. This reuse reflective size could be used as a primary input for effort estimation of ERP projects.

IV. CASE STUDY

To explore applicability of our reuse reflective size measurement approach in real-life setting and to figure out potential improvement opportunities for the approach we have conducted a case study with an SAP Implementation project. We measured COSMIC functional size of the project, defined reuse levels, calculated reuse reflective size of the project and used this size value in an existing effort estimation method. This case study is designed to answer three questions:

- How/If does COSMIC function points be used as a size measure for ERP projects?
- How/If does reuse levels be calculated for ERP projects?
- Is reuse a valuable input for ERP effort estimation?

A. Description of the Case

This case project is an SAP Implementation Project. Kick-off meeting for the project was conducted in April 2015 and Go-Live of the project occurred on 01.01.2016; after 3 months' operational support and maintenance, project was completed in March 2016.

SAP modules that were adopted in this project were:

- MM (Material Management)
- SD (Sales & Distribution)
- PM (Plant Maintenance)
- FICO (Finance & Controlling)
- HR (Human Resource)
- PP-PI (Production Planning – Process Industries)

Company implementing SAP is one of the biggest mining companies in Turkey with 3 mine sites located in different cities of Turkey. Company has almost 1200 employees and 110 of them are SAP users. SAP Consultancy Company performing the project is the Turkey branch of one of the leading IT companies in Germany.

The project included 1 Project Manager, 7 Senior SAP Consultant and 5 Junior SAP Consultant from SAP

Consultancy Company; 1 Project Manager, 3 Process Analyst and 7 SAP Key-User from SAP Adopter Company.

SAP Consultancy Company performed SAP Implementation projects based on Accelerated SAP (ASAP) methodology. Relying on ASAP, they prepare Business Blueprint Documents in the requirement elicitation phase of the project. Business Blueprint Document for the project is gathered from the company for this case study.

SAP Consultancy Company used SAP CATS Time Sheet Module for recording working times and tasks. Company provided us actual effort values categorized based on project phases, modules and consultant experience levels for this case study. The list of utilized effort for this project is presented in Table 2.

TABLE II. EFFORTS UTILIZED

Project Phase	Effort (Person-Hours)
Business Blueprint & Infrastructure	616
SAP Basis	108.5
Controlling & Budgeting	226
Finance	78
Human Resources	59
Logistics	59.5
Sales & Distribution	32
Service & Energy App.	53
Customization & Development	1230.5
SAP Basis	85.5
Mobility	85
Controlling & Budgeting	263
Finance	130
Human Resources	373.5
Logistics	90.5
Sales & Distribution	62
Service & Energy App.	141
Integration & Go Live	470.75
SAP Basis	16.25
Mobility	5.5
Controlling & Budgeting	131.5
Finance	77
Human Resources	144
Logistics	74.5
Sales & Distribution	8
Service & Energy App.	14
Operation & Support	683.75
SAP Basis	37
Mobility	7.5
Controlling & Budgeting	272
Finance	46.75
Human Resources	196
Logistics	29.5
Sales & Distribution	8
Service & Energy App.	87
Total	3001

B. Application of the Approach

Business Blueprint Document is the main source for requirement analysis in this case study. By reading “Business Processes” part and using SAP Solution Manager Business Process Repository, we were able to list all business scenarios & processes & process steps applied in this project as presented in Appendix A.

After creating business process structure, we defined the COSMIC data movements (Entry, Exit, Read and Write) on the process step level. Basically the total functional size for a business process can be calculated by summing up all data movements of included business steps. A sample view for calculation is shown in Figure 1.

Business Process	Module	Include	Entry	Exit	Read	Write	Total CFP
1. Sales Scenario		YES					
1.1. Master Data Governance for Customer		YES					10
1.1.1. Create customer	SD	YES	1			1	2
1.1.2. Search for customer	SD	YES	1	1	1		3
1.1.3. Display customer	SD	YES	1	1	1		3
1.1.4. Process customer via change request	SD	NO	1			1	0
1.1.5. Approve changes	SD	NO	1			1	0
1.1.6. Change customer	SD	YES	1			1	2
1.2. Contract Processing in ERP							18
1.2.1. Create contract	SD	YES	1			1	2
1.2.2. Create contract items	SD	YES	1			1	2
1.2.4. Display contract	SD	YES	1	1	1		3
1.2.4. Maintain target and estimated values	SD	YES	1			1	2
1.2.5. Determine and maintain texts	SD	YES	1			1	2
1.2.6. Perform credit check	SD	YES			1	1	2
1.2.7. Determine and process message output	SD	YES	1			1	2
1.2.8. Monitor contract fulfillment	SD	YES	1	1	1		3

Figure 1. COSMIC function points calculation sample

After defining data movements, we calculated reuse levels for included process steps by analyzing related business process part of Business Blueprint Document. Then, we calculated reuse reflective size by using functional similarity percentage constants as in Table 1 and reuse overhead as “0.10”. A sample view from the calculation sheet is shown in the Figure 2.

Business Process	Module	Include	Reuse Level	Func. Similarity Constant	Total CFP	Reuse Reflective Size
1. Sales Scenario		YES				
1.1. Master Data Governance for Customer		YES				2,575
1.1.1. Create customer	SD	YES	Level2	0,75	2	0,65
1.1.2. Search for customer	SD	YES	Level3	0,1	3	0,3
1.1.3. Display customer	SD	YES	Level2	0,75	3	0,975
1.1.4. Process customer via change request	SD	NO				0
1.1.5. Approve changes	SD	NO				0
1.1.6. Change customer	SD	YES	Level2	0,75	2	0,65
1.2. Contract Processing in ERP						8,55
1.2.1. Create contract	SD	YES	Level2	0,75	2	0,65
1.2.2. Create contract items	SD	YES	Level2	0,75	2	0,65
1.2.4. Display contract	SD	YES	Level2	0,75	3	0,975
1.2.4. Maintain target and estimated values	SD	YES	Level3	0,1	2	0,2
1.2.5. Determine and maintain texts	SD	YES	Level1	0,25	2	1,55
1.2.6. Perform credit check	SD	YES	Level2	0,75	2	0,65
1.2.7. Determine and process message output	SD	YES	Level1	0,25	2	1,55
1.2.8. Monitor contract fulfillment	SD	YES	Level1	0,25	3	2,325

Figure 2. Reuse reflective size calculation sample

Based on this calculation, reuse reflective size of the project for MM, SD and PM modules was calculated as 112.55 COSMIC function points, whereas total size was calculated as 247 COSMIC function points without taking into account functional similarity.

C. Effort Estimation with Reuse Reflective Size

By applying the approach, reuse reflective size value is almost a half of the standard size. Is this a valid result? To understand this, we used this reuse reflective size in an existing ERP effort estimation method suggested by Vogezang [9]. According to this method, total effort (in months) is correlated exponentially with the size of the project. Effort formula is defined as shown in Fig. 3.

$$\text{Time Delivery} = \frac{\text{Size}^{\text{Power}}}{\text{PL}}$$

Time_{Delivery} = Time to delivery of the process-chain in months
 Size = Functional size in Cfsu
 Power = 0,20 for a single production line; 0,37 for two production lines
 PL = Number of deployed production lines (1 or 2)

Figure 3. COSMIC-FPP Effort Formula

This method emphasizes that usage of production lines during implementation of the project will influence effort value. A production line is defined as a set of business functions handling business processes' specific events. In our case project, all project is handled in one production line, thus we used PL as "1" and power as "0.20".

Realized effort of these modules for our case study project was 2.15 person-months (considering 8 hours' working day). We calculated effort and corresponding MRE for "Size without considering reuse" and "Reuse Reflective Size":

Based on COSMIC FP (without considering reuse):

$$\text{Time Delivery} = \llbracket 247 \rrbracket ^{0.2/1} = 3 \text{ months}$$

$$\text{MRE} = 0.39$$

Based on Reuse Reflective Size:

$$\text{Time Delivery} = \llbracket 112.55 \rrbracket ^{0.2/1} = 2.57 \text{ months}$$

$$\text{MRE} = 0.19$$

D. Results and Discussions

The first objective of the case study was to evaluate applicability of COSMIC FPA for ERP projects. We observed that ERP Business Blueprint Document is a proper source for function point analysis. We could easily calculate COSMIC function points for the SAP Implementation project by just analyzing Business Blueprint Document and using SAP Solution Manager Business Process Repository. SAP Solution Manager provides content for all SAP applications in the form of business scenarios, business processes and process steps. Instead of creating the Business Blueprint from scratch, it can be created in SAP Solution Manager by using Business Process Repository. This will shorten total time required for CFP calculation.

The second objective was to explore reuse level calculation based on Business Blueprint Document. This document includes all necessary details to understand if minor or major enhancements required for related business processes. Reuse level definitions are clear for enhancements; we could easily decide reuse levels for the SAP Implementation Project.

Reuse levels used in our approach is valuable for especially for classification of requirements. We used these levels to decide functional similarity constants that will be used in an effort estimation formula. Instead of using reuse levels and constant rates, we could analyze if it would be

possible to measure reuse rate more precisely for ERP projects.

Reuse overhead is used as 0.10 for reuse reflective size calculation in this case study. For a proper result, reuse rate for ERP projects should be calculated for a set of projects. Reuse rate could differ based on ERP module or business scenarios. Further empirical studies are required to explore effects of the reuse overhead.

For the last objective of the case study, effort estimation result by using reuse reflective size is much better than the results obtained by using standard size as CFP. As depicted, the Mean Relative Error for effort estimation with Reuse Reflective Size is much better.

We observe that by calculating size using COSMIC, and defining reuse overhead value for ERP, calculating productivity for ERP domain and by measuring reuse rate precisely; it could be possible to make more reliable effort estimates.

V. CONCLUSIONS

In this paper, we introduced an exploratory approach for calculating reuse reflective size of ERP projects which could be used for effort estimation. We applied this approach in an SAP Implementation project to evaluate it in a real-life setting and identify improvement opportunities.

Based on the approach, reuse reflective size of an ERP project is calculated in three steps; COSMIC function points' calculation, reuse levels definition and reuse reflective size calculation. For calculating COSMIC function points, we used Business Blueprint Document of the project and SAP Solution Manager Business Process Repository. Reuse levels are calculated by analyzing "Business Processes" part of Business Blueprint. After calculating reuse reflective size, this size value is used for effort calculation.

We calculated reuse reflective size for our case study project. We used this size in an effort estimation method suggested for ERP in the literature. MRE is calculated as 0.19 whereas it was calculated as 0.39 when reuse is not taken into account for size calculation.

Based on our case study, it can be concluded that COSMIC size measurement and reuse reflective size could be a valuable input for effort estimation. We observed that the size calculation can be applied for ERP projects. Since ERP project tools as SAP Solution Manager are capable of generating Business Blueprint documents, these tools could also be used for calculation of size and effort estimates of an ERP project automatically. Approaches such as [15] and [16] can be used as a basis and can be extended for ERP projects.

Although the exploratory case study shows promising success, main threat to validity is having only one retrospective case study to evaluate the approach. To resolve this threat, the authors plan to conduct new case studies to apply the approach for a variety of projects. In those case studies, concerns mainly related to usage of reuse levels, corresponding constants and reuse overhead will be deeply analyzed.

APPENDIX A

BUSINESS PROCESSES

- Sales Scenario
 - Master Data Governance for Customer
 - Initial load for customer
 - Search for customer
 - Display customer
 - Process customer via change request
 - Approve changes
 - Change customer
 - Contract Processing in ERP
 - Create contract
 - Create contract items
 - Display contract
 - Maintain target and estimated values
 - Determine and maintain texts
 - Perform credit check
 - Determine and process message output
 - Monitor contract fulfillment
 - Assemble-to-Order Processing in ERP
 - Create assemble-to-order
 - Select inquiry or quotation
 - Determine business partner
 - Create order items
 - Perform material configuration
 - Create returnable packaging items (optional)
 - Check availability
 - Schedule order
 - Maintain prices, conditions, and costs
 - Determine and maintain texts
 - Check foreign trade data
 - Perform credit check
 - Determine and process message output
 - Monitor sales order processing
 - Billing in ERP
 - Create billing document
 - Determine business partner
 - Determine prices and conditions
 - Determine and maintain texts
 - Determine foreign trade data
 - Post rebate accruals
 - Determine and process message output
 - Perform retroactive billing
 - Generate intercompany billing
 - Order Fulfilment in ERP
 - Create delivery
 - Generate Picking List/Request
 - Send Print Delivery Documents
 - Post goods issue
- Procurement Scenario
 - Supplier Master Data
 - Create Supplier
 - Extend Supplier master data
 - Display Supplier master Record
 - Contract Processing in ERP
 - Create contract
 - Maintain authorized business partner
 - Create contract items
 - Maintain target and estimated values
 - Determine and maintain texts
 - Perform credit check
 - Determine and process message output
 - Monitor contract fulfillment
 - Processing purchase requisitions
 - Create or process purchase requisitions
 - Release purchase requisitions
- Assign source of supply to purchase requisitions
- Generate or manage versions of purchase requisitions
- Monitor or view list display of purchase requisitions
- Processing purchase orders
 - Create or process purchase orders
 - Release purchasing documents
 - Find new source of supply
 - Compare quotations
 - Generate or manage versions of purchase orders
 - Monitor the output of messages
 - Monitor or view list display of purchase order
- Store Replenishment
 - Run requirements planning
 - Check order proposals
 - Change order quantity
 - Add additional articles
 - Save order list
- Processing Contracts and Sourcing Rules in ERP
 - Process vendor master data
 - Process message conditions
 - Process contracts in ERP
 - Process purchasing info records
 - Process source list
 - Process quota arrangement
 - Process conditions for procurement
- Inbound Processing and Receipt Confirmation with Warehouse Management
 - Receive advanced shipping notification
 - Create inbound delivery
 - Post goods receipt
 - Create WM transfer order
 - Confirm WM transfer order
 - Send proof of delivery (POD)
- Maintenance Scenario
 - Phase-In Equipment
 - Create equipment
 - Create / add documents
 - Create measurement point(s) / counters
 - Create task list
 - Create maintenance plan
 - Create BOM
 - Create partners
 - Create classification information
 - Create warranty
 - Create permit
 - Create serial number information
 - Install in functional location or equipment
 - Phase-Out Equipment
 - Set equipment inactive
 - Review and close outstanding orders
 - Set maintenance plans inactive
 - Archive master data
 - Maintenance Planning, Scheduling and Dispatching
 - Create maintenance order
 - Define resources required for each operation
 - Check material and tools availability
 - Check budget
 - Define scheduling parameters
 - Define maintenance opportunities
 - Assign orders to an opportunity
 - Perform capacity leveling
 - Dispatch the order to crew or individual within crew
 - Print the job cards

- Maintenance Execution
 - Review assigned jobs
 - Execute job
 - Confirm job

REFERENCES

- [1] O. O. Top, O. Demirors, and O. Turetken, "Making functional similarity count for more reliable effort prediction models," 2009 24th International Symposium on Computer and Information Sciences, 2009.
- [2] O. O. Top, S. Tunalilar, and O. Demirors, "Evaluation of the Effect of Functional Similarities on Development Effort," 2008 34th Euromicro Conference Software Engineering and Advanced Applications, 2008.
- [3] N. Kucukates Omural and O. Demirors, "Effort Estimation for ERP Projects — A Systematic Review," 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2017.
- [4] M. Daneva, "Understanding Functional Reuse of ERP Requirements in the Telecommunication Sector: An Empirical Study," 2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement, 2014.
- [5] N. Kucukates Omural and O. Demirors, "Effort estimation methods for ERP projects based on function points," Proceedings of the 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement on - IWSM Mensura 17, 2017.
- [6] "Common Software Measurement International Consortium", The COSMIC Functional Size Measurement Method Version 3.0.1. Measurement Manual, May 2009, [online] Available: <http://www.cosmicon.com/portal/public/COSMIC%20Method%20v3.0.1%20Measurement%20Manual.pdf>.
- [7] C. Kuijpers, "Automated FPA (eFPA) in SAP Environment - Visions and Experiences of Automated Function Point Analysis," 2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement, 2014.
- [8] P. I. Erasmus, "The COSMIC EPC method-An ERP functional size measurement method delivering time and cost estimates," 2012.
- [9] F. Voegelzang, "Using COSMIC-FFP for sizing, estimating and planning in an ERP environment," 16th International Workshop on Software Measurement, 2006.
- [10] P. Erasmus and M. Daneva, "ERP Services Effort Estimation Strategies Based on Early Requirements," 2015.
- [11] F. M. Téllez, "Solving the size estimation problem in ERP project context: the eEPC-COSMIC approach," 2009.
- [12] L. Santillo, A. Abran, "Software Reuse Evaluation Based on Functional Similarity in COSMIC-FFP size components." Software Measurement European Forum–SMEF. 2006.
- [13] B. Ozkan, O. Turetken, and O. Demirors, "Software Functional Size: For Cost Estimation and More," Communications in Computer and Information Science Software Process Improvement, vol. 16, pp. 59–69, 2008.
- [14] SAP. SAP Help Portal. [Online]. https://help.sap.com/saphelp_sm71_sp11/helpdata/en/45/f6da633a292312e10000000a11466f/content.htm?no_cache=true
- [15] M. Kaya and O. Demirors, "E-Cosmic: A Business Process Model Based Functional Size Estimation Approach," 2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications, 2011.
- [16] B. Aysolmaz and O. Demirors, "Automated Functional Size Estimation Using Business Process Models with UPROM Method," 2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement, 2014.
- [17] B. Usgurlu, O. O. Top, E. Ungan, and O. Demirors, "A Clustering Based Functional Similarity Measurement Approach," 2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications, 2010.
- [18] E. Ungan, O. Demirors, O. Ozcan Top & B. Özkan, "An experimental study on the reliability of COSMIC measurement results." In International Workshop on Software Measurement (pp. 321-336). Springer, Berlin, Heidelberg, 2009.
- [19] O. Ozcan Top, O. Demirors & B. Ozkan, "Reliability of COSMIC functional size measurement results: A multiple case study on industry cases." In Software Engineering and Advanced Applications, 2009, SEAA'09. 35th Euromicro Conference on (pp. 327-334),IEEE,2009.
- [20] ISO/IEC (2018). IS 24570:2018 Software Engineering – NESMA functional size measurement method