TELKOMNIKA, Vol.15, No.1, March 2017, pp. 415~420
ISSN: 1693-6930, accredited **A** by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v15i1.3195

Reduced Software Complexity for E-Government Applications with ZEF Framework

Ahmad Nurul Fajar*[1], Imam Marzuki Shofi[2]
[1]Master in Information Systems Management, Binus Graduate Program,
Bina Nusantara University, Jakarta, Indonesia, Kebun Jeruk Raya Street No 27, (+6221) 53696969
[2]Informatics Engineering Deparment, Faculty of Science and Technology,
UIN Syarif Hidayatullah Jakarta, Jakarta, Indonesia, Ir.H.Djuanda Street No 95, (+6221) 7401925
Corresponding author, e-mail: afajar@binus.edu*[1], imam@uinjkt.ac.id[2]

Abstract

The situation of dynamic change is unpredictable and always growth increasingly. It also can happen anytime and anywhere. The one kind which is always changing is the government policy.This condition is suggested take the impact for software for information system. It will cause replacement, modification, and enhancement of software for information system. There is some commonality and variability of software features in Indonesian Government. Hence, to manage it, we present enhancement of Zuma's E-Government Framework (ZEF) for reduce software complexity.We enhance ZEF Framework using SPLE and GORE approach in order to improve traditional software development.It can reduce, if the changing continuously happen.The measurement of software complexity relate to functionality of system.It can describe with function point, because function point can describe logical software complexity also. The preliminary result of this study can reduce efficiency of software complexity such as information processing size, technical complexity adjustment factors and function points in e-government applications.

Keywords: *software product line, ZEF, e-government, GORE, software complexity*

1. Introduction

The government policy that always changing called dynamic environment in this study. It can impact for the legacy software for information system. The impact such as re-design, re-code, and re-implementation functionality of system. It causes inefficiency both of physical and logical. In general, software for information system is consists of some features. It represents the functionality of the system. A technique that can be used for managing commonalities and variability within a product line is feature modeling [1]. The need of agile system in dynamic environment such as illustrated in this study becoming crucial. Previous study has proposed of agile system in dynamic environment, but lack in detailing how to determine and manage commonality and variability in software features. Software for information system are developed with high commonality between different users, however, it is always customized for specific user needs.

SPLE can exploiting commonalities among related products in order to reduce software complexity.The commonalities are used to create a product platform that can be used as a common baseline for all products within a product family [2]. SPLE consists of domain engineering and application engineering. It can support to build a robust platform and build specific user applications [2]. SPLE requires many technical, financial, organizational, process and market considerations [3]. The benefit of SPLE compare to traditional reuse is maintenance [3, 4].

According to IEEE, software complexity is the degree to which a system or component has a design or implementation that is difficult to understand and verify [5]. The main factor and critical success factor of software development is the ability to understand relationships between requirements, design, coding, and testing [6].

There is the process for exploitation, gathering, collecting and identifying user needs in system development that called requirement engineering. According to [7], it means A Process for gathering and identifying user needs, goal of system, and documenting in a template or form.

Received October 15, 2016; Revised December 20, 2016; Accepted January 29, 2017

Regard of it, there are two approach such as traditional approach and goal oriented approach. The different between goal approach and traditional approach is goal approach used for development of software that has much complexity. It is because goal approach has a characteristic for having goal orientation in actor [7].

2. Research Method

This study has several previous researches, and will be a state of the art in this topic. We present the state of the art in research method, such as GORE, FODA, and SPLE. According to [7], the orientation of goal and actor is involved in Goal Oriented Requirement Engineering (GORE). Then, the orientation has increased dramatically in popularity. The various of modeling requirement approach in traditional has several characteristic.The characteristic are form of low-level in the data, operation, and other which are more many understood by other internal programmers & developers [8]. According to [1], FODA (Feature Oriented Domain Analysis) has a number of extensions to the original FODA (Feature Oriented Domain Analysis) notation integrate with Cardinality-based feature modeling. The each feature in a hierarchy of features has a feature cardinality is called A cardinality-based feature model.A feature cardinality has an interval, and the interval of the form [*m*..*n*], where $m \in \mathbb{Z} \wedge n \in \mathbb{Z} \cup \{*\} \wedge 0 \leq m \wedge (m \leq n \vee n = *)$ [1]. Features with the cardinality [1..1] are referred to as mandatory, whereas features with the cardinality [0..1] are called optional [1]. Besides that, features can be arranged into feature groups, where each feature group has group cardinality [1]. A group cardinality is an interval of the form (m– n), where m, n $\mathbb{Z} \wedge 0 \leq m \leq n \leq k$ [1].

The process of discovery and exploiting modeling of what is common and what differ between product variants is baseline of SPLE [2]. It is an approach that develops and maintains families of products with variability to support reuse in software for information system development [2, 6]. It allows realizing a real improvement in time to market, cost, productivity, quality and flexibility. In fact, SPL techniques are explicitly capitalizing on commonality [2]. Product line engineering has become an important and widely used approach for the efficient development of whole portfolios of software products [2]. Variability is the ability of a system to be efficiently extended, changed, customized or configured for use in a particular context [9]. In order to implement the variability concepts, there is a variability management (VM). It is one of the fundamental concepts in SPLE, which is purpose to support variant in products. This is not only taking into account the commonalities but also the variability extracted from the domain [9]. In order to achieve the good implementation, we should consider the variability. It must be considered at each development phase from the requirements collection to the final implementation [9].

3. Results and Analysis

This Section describe about data, object experimental, scenario, simulation,ZEF Framework enhancement, and evaluation

3.1. The Data and Object Experimental

Indonesia has central government and local government, which has characteristics in similarity of business process and rules. This condition, because the commonalities and variability in software features. According to the Indonesian E-Governments blueprint, the hierarchy on E-Government function is dividing into (a) block of functions; (b) block of sub-functions; and block of modules. The condition of Indonesian e-government applications depicted in Table 1 below:

Table 1. Indonesian E-Government Applications

Function Group	Number of Sub Function	Number of Module
Support and Services	4	15
Politic and Legal	2	5
Defense and Safety	2	6
Law and Policies	2	6
Economic	2	6
National Development	4	25
Publication	2	7

The Case in our study is module budgeting application. These application consists of 4 sub modules, include: budget planning, budget realization, budget monitoring, and budget evaluation. Budget planning focus to manage activity, identify what is activity. The activity has outcome, output and indicator. Budget realizations focus how to synchronize planning and realization. Budget monitoring focus how to monitor the activity and realization. Budget evaluation is how to evaluate all the activity. Table 2 below described the commonality and variability features.

Table 2. Case Study: Budgeting Application

Feature	Commonality	Variability
Manage Program	X	
Submit	X	
Approval	X	
Reject	X	
View & Report	X	
Calculate	X	
Manage TOR	X	
Manage Transaction	X	
Manage Indicator	X	
Manage Input	X	
Manage Output	X	
Manage Outcome	X	
Manage Impact	X	
Formula		X
Lakip	X	
Create and legalized SPP		X
Create and legalized SPM		X
Create legalized SPD		X

3.2. The Scenario and Simulation

In this study, we develop several scenarios. This scenario is implemented in simulation environment. The environment for simulation is described in Figurre 1.

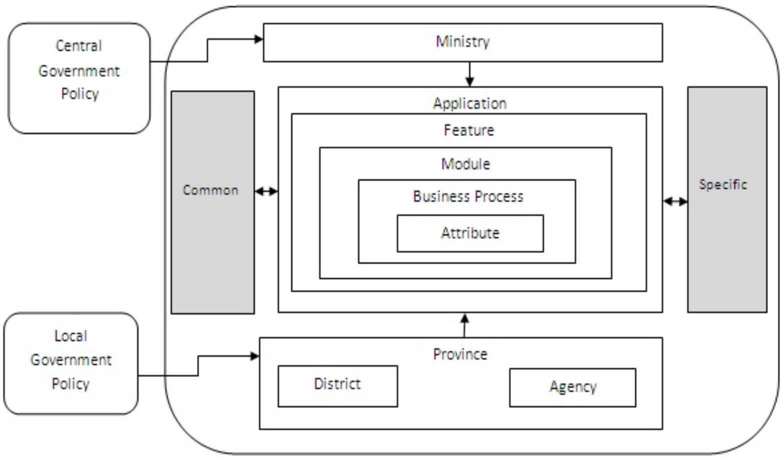

Figure 1. Environment for Simulation

The Case in our study is budgeting application.The scenario based on the environment simulation above are:

Feature Embedded, this scenario describe feature that consist in local government and central government. This feature refers to budgeting application. There are 15 features that cover business process for budgeting planning, budgeting realization, budgeting monitoring and budgeting evaluation.

Table 3. Feature Embedded Scenario

Feature	Local Government	Central Government
Manage Program	V	V
Submit	V	V
Approval	V	V
Reject	V	V
View & Report	V	V
Calculate	V	V
Manage TOR	X	V
Manage Transaction	X	V
Manage Indicator	V	X
Manage Input	V	X
Manage Output	V	V
Manage Outcome	V	V
Manage Impact	V	V
Formula	X	V
Lakip	X	V
Create and legalized SPP	X	V
Create and legalized SPM	X	V
Create legalized SPD	X	V

3.3. The ZEF Framework Enhancement

ZEF Framework has created by Professor Zainal Arifin Hasibuan, Dr. Eko K Budiardjo, and Dr. Ahmad Nurul Fajar in 2012. In this work, we make enhancement of ZEF Framework. We proposed enhancement ZEF Framework with adding SPLE and GORE Approach. It used for accomadate the construction of SPL Platform and Goal of the organization. It also reference and inspired by [10] mechanism.

The ZEF Framework Enhancement described in Figure 2. It is explain about the mechanism of to construct software product line platform. In order to construct it, we used goal model to transform from domain engineering to application engineering in SPLE. It means we make analysis in domain engineering before create goal model. Then, in order to capture feature model, we used goal model for eliminate the semantic and meaningfull from the domain. In the last stage, we develop application platform from feature model in application engineering phase.

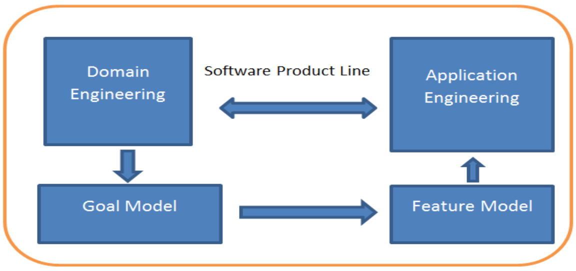

Figure 2. ZEF Framework Enhancement

According to Figure 2 above, we proposed the mechanism for checking consistency in goal model that described in Figure 3 below. Figure 3 described the flow of stage for checking consistency in goal model. This mechanism is derived from GCC Method which is achieved from goal model in domain engineering phase.

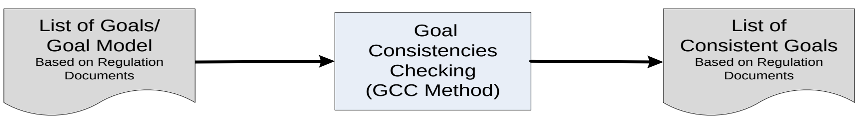

Figure 3. Mechanism for Checking Consistency in Goal Model

According to scenario and simulation that had been described, the preliminary result, we improve function point analysis for calculate software complexity. The improvement are:

1. We added component software
2. We classify degree of complexity

There are the step to estimate cost development:

1. Calculate CRUD Function Point

The calculation function point for CRUD functionality is described in Figure 5 below. This figure represent the value of function point which is focus on CRUD functionality in the software.

Component Sofware	Complexity									Total CFP
	Simple			Average			Complex			
	Count	Weight	Point	Count	Weight	Point	Count	Weight	Point	
	A	B	C+	D	E	F	G	H	I	C+F+I
Input	4	3	12		4		3	6	18	30
Output	2	4	8	7	5	35	3	7	21	64
Query Online	2	3	6	2	4	8	2	6	12	26
File Logic	4	7	28		10	0	3	15	45	73
InterfaceExternal		5	0		7	0	4	10	40	40
Total CFP										233

Figure 5. Calculate CRUD Function Point

2. Calculate RCAF (Relative Complexity Adjustment Factor)

From figure above, Total CFP = 233. The next step is calculate RCAF (Relative Complexity Adjustment Factor). RCAF used to evaluate complexity characterstics. Table 4 below described the result of RCAF Calculation:

Table 4. RCAF Calculation

Subject	Score					
	0	1	2	3	4	5
Back up/Recovery						v
Data Communication		v				
Distributed Processing						v
Transaction Rate				v		
Updating File Master	v					
Installation	v					
Input,Output,Query Online, File			v			
Data Processing			v			
Reuse Code					v	
Flexibility/Response Change					v	
End user Efficiency		v				
Total = RCAF						**27**

3. Calculate Function Point (FP)

The last step is Calculate FP, we can calculate FP using : FP = CFP x (0.65 + 0.01 x RCAF). After that, if CFP = 233 and RCAF = 27. Then FP = 233 x (0.65+0.001 x 27) = 4095,441.

3.4. Evaluation

The evaluation in our study will compare CRUD function point, RCAF and function point between using enhancement ZEF Framework and without enhancement ZEF Framework. The result of the evaluation relate to case in our study. Value for calculate CRUD function point, RCAF and function achieved from software architect expert. There are 5 software architect expert contribute to judge the value. Table 5 below described the result of the evaluation of enchancement of ZEF system Architecture.

Table 5. Evaluation

Variable	Using Enhancment ZEF Framework	Without Enhancment ZEF Framework
CRUD function point	233	525
RCAF (Relatively Complexity Adjustment Factor)	27	46
Function Point	4095,441	15721,65

4. Conclusion

The enhancement ZEF Framework is construct with SPLE and GORE approach. The results from it is The Framework that can provide guidance for software developers to construct software product line platform. It can reduce efficiency software complexity such as information processing size, technical complexity adjustment factors and function points in the case of budgeting application for e-government applications.

Acknowledgements

This paper was fully supported by Faculty of Computer Science; University of Indonesia.This paper also was fully supported by Prof. Zainal A Hasibuan, Dr. Eko K Budiardjo and Prof Heru Suhartanto from Faculty of Computer Science University of Indonesia.

References

[1] Krzysztof Czarnecki, Chang Hwan Pet*er Kim. Cardinality-Based Feature Modeling and Constraints:A Progress Report*. OOPSLA'05, Workshop on Software Factories. San Diego, California, USA. 2005.

[2] Günter Böckle, Klaus Pohl, Frank van der Linden. Germany: Springer. 2005: 19-38.

[3] Sami Ouali, Naoufel Kraiem, Henda Ben Ghezala. Framework for Evolving Software Product Line. *International Journal of Software Engineering & Applications*. 2011; 2(2): 34-51.

[4] Robak S. *Feature Modeling Notations for System Families.* Proceeding of the International Workshop on Software Variability Management. Portland, Oregon. 2003: 58-62.

[5] IEEE Standards Association. 1076.3-2009. *IEEE Standard VHDL Synthesis Packages*. New York: IEEE Press; 2009.

[6] Stan Bühne, Klaus Pohl. Domain Requirements Engineering. Germany: Springer. 2005: 93-216.

[7] Shofi IM, Budiardjo EK. *Addressing OWL ontology for goal consistency checking*. The 14th International Conference on Information Integration and Web-based Applications & Services. New York, USA: ACM Press. 336.

[8] Fajar AN, Shofi IM. Addressing Consistensy Checking of Goal Model For Software Product Line Government Tourism System. *Journal of Environmental Management and Tourism*. 2016; 1(13): 5-10.

[9] R Bonif´acio, P Borba. *Modeling scenario variability as crosscutting mechanisms*. In Proc. of AOSD '09. 2009: 125-136.

[10] Mardiana T, Adji BT, Hidayah I. Stemming Influence on Similarity Detection of Abstract Written in Indonesia. *Telkomnika.* 2016; 14(1): 219-227.