

Evaluating Services Computing Systems Engineering Framework using An Acceptance Model

Novianto Budi Kurniawan^{#1}, Suhardi^{#2}, Yoanes Bandung^{#3}, Yuli Adam Prasetyo⁺, Purnomo Yustianto^{*}

[#]*School of Electrical Engineering and Informatics, Institut Teknologi Bandung, 10 Ganesha, Bandung, 40132, Indonesia*
E-mail: ¹noviantobudik@s.itb.ac.id; ²suhardi@stei.itb.ac.id; ³bandung@stei.itb.ac.id

⁺*School of Industrial Engineering, Telkom University, Indonesia*
E-mail: adam@telkomuniversity.ac.id

^{*}*School of Information Technology, Deakin University, Geelong, Australia*
E-mail: pyustian@deakin.edu.au

Abstract— The gap between business services and IT services becomes a major concern in services computing. As an approach for service-based IT solution, services computing systems are promised to be able to bridge the gap between these services. The implementation will require an engineering framework as a guide to building the systems. The framework needs to be evaluated to provide important feedback to the framework development. This paper outlines the evaluation of SCSE framework through an acceptance model. The study develops an acceptance model based on the experiences of a group of engineers after using the framework to build smart campus services systems. A survey involving 54 systems engineers with various engineering backgrounds was conducted to assess the experiences of the engineers in using the framework. The results of the acceptance model show that both perceived ease of use, represented by the level of agreement (v1) and perceived usefulness, represented by the level of importance (v2) deliver good results almost for the entire stages of the proposed framework. In addition, the user experiences of using the proposed framework are in the acceptable levels. The contribution of this paper is an enrichment of the engineering methodologies for the service-oriented system from the perspective of services computing.

Keywords— acceptance model; framework; services computing systems; services computing systems engineering; smart campus.

I. INTRODUCTION

Service innovations are required by each organization to improve its business services [1], [2]. Service innovations can be carried out by proposing new services or improving existing services. New services can be provided by enhancing existing services [3]. Better services can be provided to its users by improving their service experiences [4]. This can be achieved through the service technology and features improvements. Service technology and architecture, such as web services, cloud computing, mobile computing, and service-oriented architecture (SOA) have evolved in providing opportunities for realizing IT services systems that enable the service innovations [5], [6]. This circumstance will trigger the ability to present the systems that support business services innovations.

One of the challenges in services computing research is the service design process. This challenge is also a fundamental research problem in services computing [7]. Service design comprehension is essential for building and

developing a service computing-based system [7]–[9]. So far, a service design has not been based on a formal model of services computing systems [7], [10]–[12]. Service design in a large complex system often disregards the necessity for unifying systems engineering framework [13]–[16]. Highly complex services systems drive the need for an engineering framework. The required framework must support the interaction among services systems components with diverse functionalities. Thus, an engineering framework to build the systems is urgently required in this field. An engineering framework for services computing systems has been proposed in previous studies [17], [18]. The framework is expected as a structured guide in developing the systems. However, the framework needs to be implemented and evaluated in the real systems environment [18].

From a service domain, studies on service engineering have been accompanied by various forms of research streams [19]–[21]. Some of these streams are commonly based on the SOA concept. From the more comprehensive services systems domain, the studies on services systems

engineering have also been done extensively [15], [22]–[25]. The studies combine service engineering and systems engineering methodologies. These works offer a framework to be used as a process guide in engineering services and services systems from various perspectives.

Two perspectives underlie the concepts of services computing systems. First, from the view of systems characteristic, services computing systems are the main research subject that collaborates IT-enabled services systems, SOA, and services computing technology. The systems cover IT services, business services, and service values. So instead of focusing only on IT services, the systems must consider and meet the needs of business services [26]. The systems are built based on an alignment consideration between business and IT services [27].

Furthermore, the implementation of the systems must be able to provide service values to the organization [28]. The second perspective takes the view of systems engineering. Services computing systems engineering is considered a collaboration of multi-disciplines engineering methodology, i.e., services systems engineering, services engineering, and systems engineering. Services systems engineering combines SOA-based service engineering and systems engineering. From the SOA-based service engineering approach, the methods implement SOA principles to meet business service needs. From a systems engineering approach, principles and life cycle of systems engineering are adopted.

There are limitations to the previous studies. A service engineering methodology in general emphasizes the SOA approach to design IT services that support the business services. Some popular SOA based service engineering methodologies, i.e., SOMA, SOAF, MSOAM, and SOAD [21], [29] focus solely on IT services design based on the given business services needs [25][17]. The methodologies commonly shield the design of individual IT services without considering the characteristic of systems. This circumstance may risk the lack of the systems integration of the services. In practice, the analysis of business needs domains are typically required, such as in term of business strategy, business service model, business service requirement, business service innovation, service technology adoption, and service performance. The activities shall be fully considered to guarantee the alignment between IT services and business services [30].

The methodology of services systems engineering emphasis is on the development of services systems to yield business services based on the systems perspective [31]. However, a gap still exists since the methodology does not cover the design and analysis of IT services. Some undertakings that emphasis on the analysis of existing IT systems and implementation strategy for technology adoption is a lack in the methodology. The concern for IT services operationalization process in enabling the service's systems is also neglected.

A systems engineering methodology should be a key consideration in building the services computing systems that align both IT and business services [17], [31]. The methodology elaborates system design and development process comprehensively and systematically. It should also consider a multi-discipline engineering approach in designing and realizing the systems [32]. From the

engineering approach, the services computing systems engineering practices a similar approach with the services systems engineering. It should also consolidate more extensive engineering knowledge to promote collaborative value creation of the systems [31], [32].

In this study, services computing systems engineering (SCSE) framework is defined as a methodology for developing the services computing systems. It covers the design, development, deployment, and evaluation of the systems. The methodology covers a complete engineering life cycle that enables the design and the implementation of the systems in a systematic manner. The methodology is visualized as stages and phases (sub-stages) in developing the systems. Fig. 1 shows the SCSE framework that enhances the previous studies [17], [31]. The framework is constructed from the meta model and lifecycle of services computing systems [31]. The SCSE framework covers two types of services, i.e., business services and IT services. In the area of services computing, IT services represent software services.

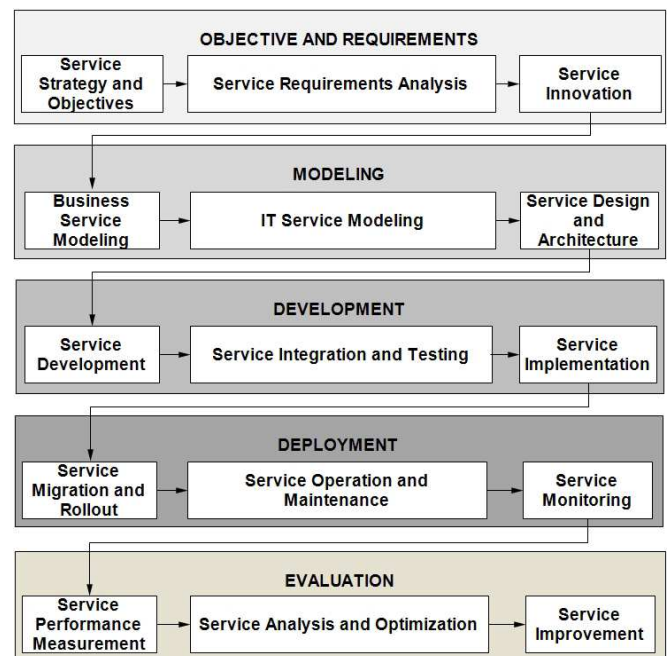


Fig. 1 Services computing systems engineering (SCSE) framework

The framework contains five stages, i.e., objectives and requirements, modeling, development, deployment, and evaluation. The first stage focuses on the identification of service strategy, objective, requirements, and innovation. The stage shall be based on the business needs with main consideration to improve the business service of the organization. There are three phases covered in the stage, i.e., service strategy and objectives, service requirements analysis, and service innovation. The second stage focuses on service modeling and system design. There are three phases covered in the stage, i.e. business service modeling, IT service modeling, and service design and architecture. This is the essential stage of the framework. The third stage comprises three phases: service development, service integration, and testing, and service implementation. The stage requires the engineers to implement the design and

architecture of the service from the second stage. The implementation of the stage relevant to the software development activities, but with a service-oriented approach.[33]. The fourth stage covers three phases: service migration and roll-out, service operation and maintenance, and service monitoring. The stage focuses on the deployment and operation of the systems. Finally, the fifth stage focuses on the evaluation process of the systems. There are three phases included in the stage, i.e., service performance measurement, service analysis, and optimization, and service improvement. Services dependability can be used to measure and evaluate the system's performance from the internal systems perspective [34], while services quality can be used to measure the performance from the external perspective [32], [35].

The purpose of this paper is to evaluate the framework using an acceptance model. The motivation of this research is to provide a guide for building the systems through a validated engineering framework. The evaluation delivers important feedback to the improvement of the framework. Furthermore, the evaluation of the framework can maximize the utilization of the framework. The remainder of the paper is ordered as follows. Section II describes the materials and methods for evaluating the framework. Section III discusses the results, and section IV provides the research conclusion.

II. MATERIALS AND METHODS

In assessing a newly designed artifact, evaluation is an important step to be made [36]–[38]. The artifact produced in this work is the engineering framework of services computing systems. The evaluation provides important feedback to the framework development [28]. A practical evaluation methodology is used to evaluate the proposed framework based on the experiences of the systems engineers after using the framework. This leads to the construction of the acceptance model. The evaluation methodology used in the study involves the implementation of both qualitative and quantitative techniques.

The study develops an acceptance model to evaluate the framework based on the experiences of the engineers in using the framework. Fig. 2 shows the methodology for evaluating the framework. First, the group of potential systems is identified to support the evaluation in a real environment. Second, an acceptance model is built to define user experiences. Third, survey data is collected from the engineers involved in the systems development. Fourth, the acceptance model is run using the data that have been collected. Finally, acceptance results are obtained and presented.

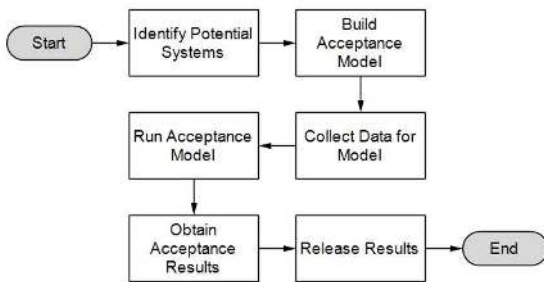


Fig. 2 A methodology for evaluating the proposed framework based on acceptance model (Adapted from [39])

A. Acceptance Model

Fig. 3 shows the acceptance model of the framework evaluation. The model adopts the technology acceptance model [40]. Four external variables are used in the acceptance model, i.e. Z_1 : level of the clarity of the framework (systematic stages, phases and steps), Z_2 : level of the clarity of the artefacts in every stage, Z_3 : level of the easiness for documenting the artefacts in every stage, and Z_4 : level of the easiness for a trace-back mechanism if error occurs. These variables will construct the perception of the users. There are two internal variables used in the acceptance model that construct the perception results of the systems engineers, i.e., perceived ease of use (v_1) and perceived usefulness (v_2). In this study, v_1 is represented in the form of a level of the agreement while v_2 is represented in the form of level of importance. Level of the agreement states the level of understanding, clarity, and ease of each stage. The variable represents the answers to individual perceptions in understanding each stage, phase and steps proposed in the framework. On the other hand, the level of importance states how important the role, contribution, and evaluation of each phase of the entire stages. This variable represents the individual's perception of the importance of each stage, the phases, and steps proposed in the framework.

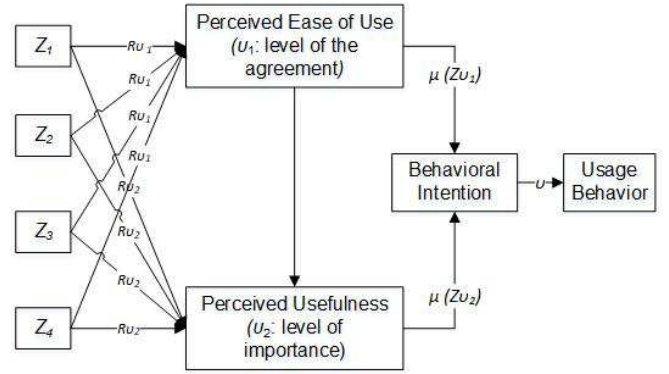


Fig. 3 Acceptance model (v) of the framework (Adapted from [40])

The acceptance model (v) for both v_1 and v_2 are defined as follows:

$$v_1 = \mu(Zv_1) = \mu(Z_i | Sv_1) \quad (1)$$

$$v_2 = \mu(Zv_2) = \mu(Z_i | Sv_2) \quad (2)$$

Where $\mu(Zv_1)$ and $\mu(Zv_2)$ are the average value of Z for v_1 and v_2 respectively, Z_i is the value of Z for both v_1 and v_2 where $i : [1..4]$ and $Z_1, Z_2, Z_3, Z_4 \in Z$. Sv_1 and Sv_2 are the stage value for each v_1 and v_2 . The model of Z_i for v_1 and v_2 are defined as follows:

$$Z_i(v_1) = \omega_z(Cv_1) \sum_{i=1}^{Np} \sum_{j=1}^M (Rv_{1ij}) \sigma_{ij} \quad (3)$$

$$Z_i(v_2) = \omega_z(Cv_2) \sum_{i=1}^{Np} \sum_{j=1}^M (Rv_{2ij}) \tau_{ij} \quad (4)$$

Where $\omega_s(Cv_1)$ and $\omega_s(Cv_2)$ are the weighted factors for the criteria used in v_1 and v_2 , respectively, with the range value of $20 \leq \omega \leq 100$ by 1 to 5 scales. Rv_1 and Rv_2 are the response value for both v_1 and v_2 for every Z_i . σ_{ij} : Binary variable with value 1 when the v_1 response of engineer i is valid on stage j , 0 otherwise. τ_{ij} : Binary variable with value 1 when the v_2 response of engineer i is valid on stage j , 0 otherwise. Np is the number of participants involved, and M is the number of stages.

B. Data Collection

This research conducts a case study to evaluate the proposed framework. For these purposes, a case study for smart campus services systems development is fully conducted. This study collected data and information through a case study of smart campus services systems development. SCSE framework was tested on the development of smart campus as a representation of service computing systems. The case study produced a smart campus design and application.

Furthermore, the development of smart campus services systems represents the engineering process using the SCSE framework. The practical experiences of the system engineers as participants in using the framework were evaluated using the acceptance model. There are 18 services systems developed in this case study that involve a group of systems engineers (See Table I). Each service system is assigned to different systems engineers. Thus, for each service system i selected, the engineer is assigned to work on the service system i .

There are 54 engineers with various engineering backgrounds involved in the framework evaluation, i.e., Information Technology (IT), Information Systems (IS), Information Technology Services (ITS), and Software Engineering (SE) (See Fig. 4). A training and knowledge sharing is delivered to the systems engineers, both from the theoretical and technical concepts that are relevant to the application of the proposed framework. Each engineer is also equipped with a technical briefing regarding the service systems characteristics and the technique to develop the systems using the proposed framework.

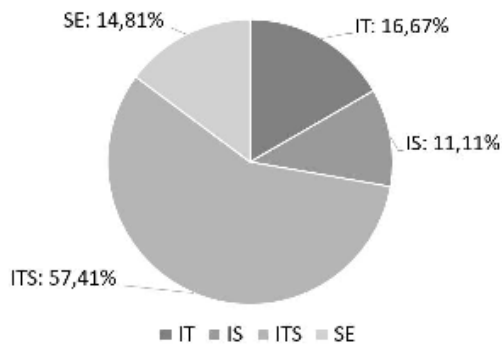


Fig. 4 Percentage of engineers grouped by engineering backgrounds

The acceptance model in this study is implemented through a survey. Respondents of the survey are the systems engineers that involved in the case study. A survey was conducted to assess the experiences of the systems engineers after using the framework. The objective of the survey is to observe the feasibility of the proposed framework,

constructed from four variables [21], i.e. $Z_1, Z_2, Z_3,$ and Z_4 (See Section IV.B for the details). The questionnaire is based on two acceptance models, i.e., level of agreement (v_1) and level of importance (v_2). The questionnaire uses a Likert Scale with 5 value for each of the acceptance model based on level of agreement (1: strongly disagree, 2: disagree, 3: neutral, 4: agree, and 5: strongly agree) and level of importance (1: not important, 2: slightly important, 3: moderately important, 4: important, and 5: very important).

III. RESULTS AND DISCUSSIONS

A. Systems Identification

Table I shows the list of service systems that are used to evaluate the framework. The systems are used as a medium to implement the framework. Group of engineers is assigned to build the systems using the proposed SCSE framework.

The set function of the services system (SS) is formulated as follows:

$$SS_i = \bigcup_{i=1}^n \bigcup_{j=1}^m SI_{ij} S_{ij}, O(SS_{ij}), O \in S \wedge S \in SS \quad (5)$$

Where SS : service system, SI : services interface, $SS_i \leftarrow SI_{1j}, SI_{2j}, SI_{3j}, \dots, SI_{ij}$, S : service, $SS_i \leftarrow S_{1j}, S_{2j}, \dots, S_{ij}$, O : operation of service, $S_j \leftarrow O_{j1}, O_{j2}, \dots, O_{ij}$, m : number of service in service system i , n : number of service system. An operation O pertains to service in its relevant service system, and a service interface SI belongs to its respective service system SS .

TABLE I
LIST OF SMART CAMPUS SERVICES SYSTEMS

No	Service Systems (SS)	Number of		
		SI	S*	O
1	Smart Learning Management System	16	12	84
2	Personalized Learning System	8	7	22
3	Assessment System	7	6	18
4	Smart Classroom	12	12	50
5	Library Management System	13	8	24
6	Smart Attendance	5	4	17
7	People Identification	9	5	24
8	Geographic Information System	7	7	25
9	Bathroom Management System	12	10	28
10	Smart Parking System	10	8	32
11	Teaching Management System	16	12	61
12	Financial System	14	14	45
13	Office System	11	8	29
14	Market Management System	11	10	30
15	News Management System	8	8	28
16	Smart Building System	12	10	47
17	Waste and Water Management	9	6	16
18	Health Monitoring System	14	14	55
Total		194	161	635

* Number of services after being composed

There are 18 services systems in the domain of smart campus [41]. Each service system contains a various number of services interfaces, composite services, and service operations. The numbers are obtained from the design results produced by the engineers when building the system. The differences in the number of SI , S^* , and O show that the

systems are varied as well as the size of the systems. Smart learning management system and teaching management system are two service systems that have the most number of service interfaces. Although the size of the service systems varies, in principle, the systems engineers will get the same environment in using the SCSE framework. The practical experiences of the engineers in building the systems using the proposed framework are the main data for the evaluation.

A. Acceptance Results

Table II and Table III show the results of data processing for both levels of the acceptance model v_1 and v_2 for every Z value based on stages (Stg). Each Z value for both v_1 and v_2 shows the average perceptions of variable Z for each stage. The perceptions of the engineers are assessed based on the criteria of Cv_1 and Cv_2 which have been corrected by the weighted factor ω_z . It can be seen from the Table II that the highest average value for the level of the agreement v_1 is on the variable Z_1 which reaches 79.8, followed by variable Z_3 which reaches 79.1, variable Z_2 which reaches 77.2, and variable Z_4 with the lowest average value reaching 76.0. The results indicate that the perceptions of the engineers of the variable Z_1 and Z_3 are likely to be better than the other two variables. Meanwhile, if the results are broken down per stage, it can be seen that the perception's average value of the level of the agreement towards stage 1 (82.5) and stage 2 (80.20) occupies the two highest positions, followed by stage 5 (79.9). This indicates that the three stages hold a vital function in implementing the framework. Stage 1 and stage 2 have become the foundation of the proposed framework. Showing from the value of R^2 , the variable Z_1 gave the highest contribution value in the acceptance model for the level of agreement, which reached 68.90 percent.

TABLE II
RESULTS OF ACCEPTANCE MODEL LEVEL OF THE AGREEMENT (v_1)

Z_i	$\mu(Z_i/Sv_1)^*$					v_1	R^{**}	R^2
	Stg1	Stg2	Stg3	Stg4	Stg5			
Z_1	85.6	80.8	76.7	72.6	83.4	79.8	0.830	0.689
Z_2	80.8	79.7	75.6	72.3	77.5	77.2	0.761	0.580
Z_3	84.1	82.3	76.7	71.5	80.8	79.1	0.821	0.675
Z_4	79.3	78.2	73.8	70.8	77.8	76.0	0.704	0.496
μS_i	82.5	80.2	75.7	71.8	79.9			

* Cronbach's Alpha: 0,861 with 54 valid cases

** Correlation R is significant at the 0.05 level (2-tailed)

TABLE III
RESULTS OF ACCEPTANCE MODEL LEVEL OF IMPORTANCE (v_2)

Z_i	$\mu(Z_i/Sv_2)^*$					v_2	R^{**}	R^2
	Stg1	Stg2	Stg3	Stg4	Stg5			
Z_1	88.9	85.6	82.6	79.3	84.1	84.1	0.934	0.871
Z_2	86.3	85.2	80.0	78.2	82.6	82.4	0.962	0.926
Z_3	84.1	86.0	83.8	77.5	81.9	82.6	0.949	0.900
Z_4	86.3	86.7	85.2	81.5	83.0	84.5	0.965	0.931
μS_i	86.4	85.9	82.9	79.1	82.9			

* Cronbach's Alpha: 0,957 with 54 valid cases

** Correlation R is significant at the 0.05 level (2-tailed)

The result of the acceptance model for level of importance v_2 shows the results that are slightly different from the level of agreement (See Table III). The results come with a

relatively higher average value. The highest average value for the level of importance v_2 is in the variable Z_4 which reaches 84.6, then followed by the variable Z_1 which reaches 84.1, variable Z_3 which reaches 82.6, and the variable Z_2 with the lowest average value reaching 82.5. The results indicate that the engineers' perceptions of the variable Z_1 and Z_4 are likely to be better than two other variables, although the difference is relatively small.

Meanwhile, if the results are breakdown per stage, it can be seen that the average value of the importance level shows the results that are relatively higher than the level of agreement. The average value of the engineers' perceptions of stage 1 (86.4) and stage 2 (85.9) occupies the two highest positions, followed by stage 3 and stage 5 which have the same value (82.9). This indicates that these stages play a very important role in implementing the framework. Showing from the value of R^2 , the variable Z_4 gives the highest value of contribution in the acceptance model for the level of importance, which reaches 93.10 percent. This is slightly different from the results in the previous model. From a level of the important point of view, the easiness for a trace-back mechanism, if an error occurs is a top priority for the engineers.

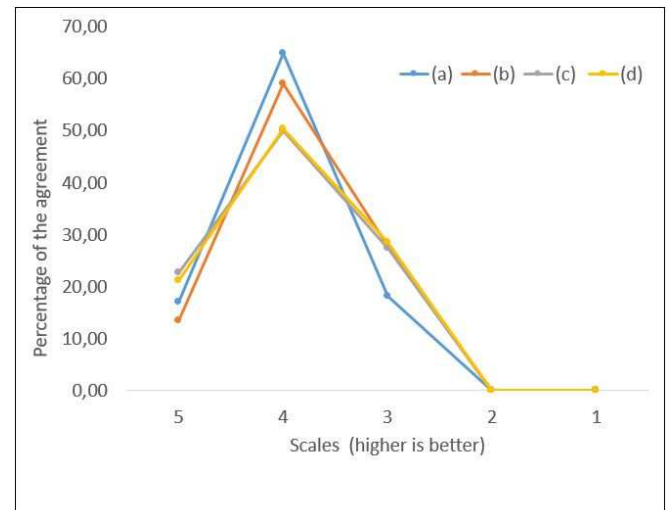


Fig. 5 Perceived ease of use of using the framework: level of the agreement. (a): level of the clarity of systematic stages, phases and steps [Z_1], (b): level of the clarity of the artefacts in every stage [Z_2], (c): level of the easiness for documenting the artefacts in every stage [Z_3], and (d): level of the easiness for a trace-back mechanism if error occurs [Z_4].

Fig. 5 shows the assessment results of the acceptance model for the level of the agreement v_1 . It can be seen generally from the Figure that the responses of the level of the agreement to the proposed framework are good for all Z variables. This is indicated by the number of percentages that agree to the four Z variables above 70 percent (agree and strongly agree). The total percentage of neutral is below 30 percent. Meanwhile, the percentage for disagree and strongly disagree criteria are zero. The results indicate that the proposed framework can be well received by the engineers. The engineers agree that the framework can be easily understood and implemented, the artefacts can be easily evaluated, the output can be clearly documented, and an error trace-back mechanism can be done simply.

The following are the detailed results of the level of agreement for all Z variables. For the variable Z_1 , there are 72 percent of the engineers who agree that the stages, phases, and steps in this framework are clear, systematic and easy to understand, while 18 percent of the engineers are neutral, and none of them state disagree. For the variable Z_2 , there are also 72 percent of the engineers who agree that the steps of each stage and phase are clear, can be well implemented, and easy to be evaluated, while 28 percent of the engineers are neutral, and none of them state disagree. For the variable Z_3 , it represents 73 percent of the engineers who agree that the results of each stage and phase are easily documented, while 27 percent of the engineers are neutral, and none of them state disagrees. For the variable Z_4 , there are 71 percent of the engineers who agree that when an error occurs in a particular stage and phase, it is easy to do a trace-back mechanism, while 29 percent are neutral, and there are no engineers who respond did not agree.

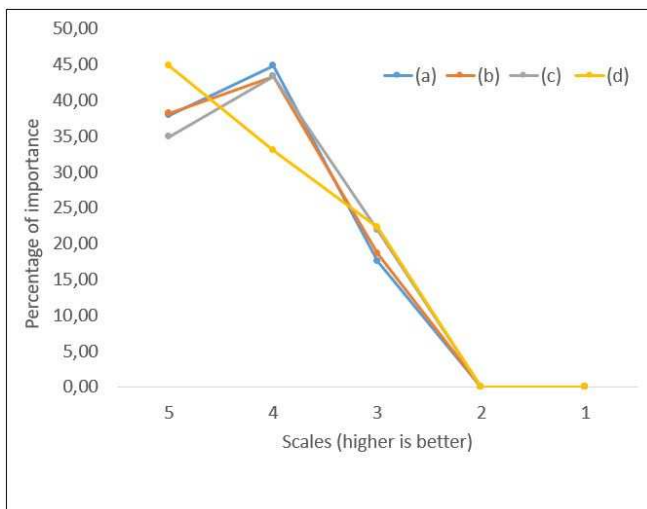


Fig. 6 Perceived usefulness of using the framework: level of importance. (a): level of the clarity of systematic stages, phases and steps [Z_1], (b): level of the clarity of the artefacts in every stage [Z_2], (c): level of the easiness for documenting the artefacts in every stage [Z_3], and (d): level of the easiness for a trace-back mechanism if error occurs [Z_4].

Fig. 6 shows the assessment result of the acceptance model for the level of importance v_2 . It can be seen generally from the Figure, that the responses to the proposed framework for the level of importance are good for all Z variables. This is indicated by the number of percentages that agree to the four Z variables above 75 percent (very important and important). This percentage value tends to be better compared to the results of the level of agreement v_1 which is only above 70 percent. Even for Z_1 and Z_2 , the percentage value is above 80 percent, while for Z_3 and Z_4 , the percentage value is close to 80 percent. The total percentage of neutral is below 25 percent. In addition, the percentage of slightly important and not important criteria are zero. The results indicate that the proposed framework has a good level of importance in almost all Z records. The engineers consider that it is very important to understand and implement each stage, phase, and step easily, it is very important to easily evaluate the artefacts of each stage, it is important to easily document the results, and it is important to do an easy trace-back mechanism.

The following are the detailed results of the level of importance for all Z variables. As for the variable Z_1 , it represents 83 percent of the engineers who feel the importance of a systematic framework, while 17 percent of respondents feel moderately important, and none of them states not important. For the variable Z_2 , there are 81 percent of the engineers who feel the importance of step clarity and the results of each stage and phase so that it can be easily evaluated, while 19 percent of the engineers feel moderately important, and no engineers stated not important. For the variable Z_3 , there are 78 percent of the engineers who feel the importance of ease in documenting the results, while 22 percent are neutral, and none of them stated not important. For variables Z_4 , there are also 78 percent of the engineers who feel the importance of ease in documenting the results, while 22 percent of the engineers are moderately important and no engineers who respond not important.

Based on the results of the acceptance models above, both perceived ease of use, represented by the level of agreement v_1 and perceived usefulness, represented by the level of importance v_2 , show good results almost for all stages of the proposed framework. Overall, each stage of the framework shows the good performance of ease and clarity in building the systems. The acceptance results of the Z values are promising especially on stages 1, 2 and 5. Each stage, phase, and step can be easily understood and well implemented. Outputs and artefacts of the stages can be properly evaluated. The outputs of the stages can be well documented. If something goes wrong, a good evaluation and trace-back mechanism can be systematically carried out.

IV. CONCLUSIONS

This paper describes the evaluation of the SCSE framework. The framework is implemented through a smart campus services systems development and involving systems engineers with various engineering backgrounds. The evaluation of the framework is based on the acceptance model that delivers the experiences of the engineers of using the framework for building the systems. The framework evaluation is successfully conducted with promising results. The results of both acceptance models, in general, show a good acceptability level. The perceived ease of use, represented by the level of the agreement and the perceived usefulness, represented by the level of importance, show good results almost for the entire stages of the framework. According to the acceptance results, it can be seen that the user experiences of using the proposed framework are in the acceptable levels. Our future research aims to elaborate the SOA principles with the proposed framework by SoaML to build the services computing systems. Thus, several key principles of service-oriented architecture design will be fully covered in future efforts.

REFERENCES

- [1] H. P. Devarapalli, "Value of IT innovations for a sustainable business," *2014 17th IEEE Int. Conf. Intell. Transp. Syst. ITSC 2014*, pp. 184–187, 2014.
- [2] Z. Wu, S. Member, J. Yin, and S. Deng, "Modern Service Industry and Crossover Services: Development and Trends in China," *IEEE Trans. Serv. Comput.*, vol. 9, no. 5, pp. 664–671, 2016.
- [3] Q. Li, J. Shi, and H. Zhu, "A formal framework for service mashups with dynamic service selection," *Innov. Syst Softw Eng*, vol. 10, pp.

- 219–234, 2014.
- [4] D. X. Ding, P. J. Hu, R. Verma, and D. G. Wardell, “The Impact of Service System Design and Flow Experience on Customer Satisfaction in Online Financial Services,” *J. Serv. Res.*, 2010.
- [5] L. Zhang, J. Zhang, and H. Cai, *Services Computing*. New York: Springer US, 2007.
- [6] Z. Wu, S. Deng, and J. Wu, *Service Computing Concepts, Methods and Technology*, vol. 1. Waltham, MA: Elsevier Inc., 2015.
- [7] A. Bouguettaya *et al.*, “A Service Computing Manifesto: The Next 10 Years,” *Commun. ACM*, vol. 60, no. 4, pp. 64–72, 2017.
- [8] J. Hu, L. Huang, R. Wu, B. Cao, and X. Chang, “Model-Driven Design And Validation Of Service Oriented Architecture Based On Devs Simulation Framework,” *Int. J. Serv. Comput.*, vol. 2, no. 3, pp. 14–29, 2014.
- [9] S. Wang, L. Li, and J. D. Jones, “Systemic thinking on services science, management and engineering: Applications and challenges in services systems research,” *IEEE Syst. J.*, vol. 8, no. 3, pp. 803–820, 2014.
- [10] S. Eisele, “RIAPS: Resilient Information Architecture Platform for Decentralized Smart Systems,” in *2017 IEEE 20th International Symposium on Real-Time Distributed Computing*, 2017, no. section III, pp. 125–132.
- [11] N. Haile and J. Altmann, “Evaluating investments in portability and interoperability between software service platforms,” *Futur. Gener. Comput. Syst.*, pp. 1–18, 2017.
- [12] A. K. Mandal and A. Sarkar, “Service Oriented System design: Domain Specific Model based approach,” in *Computer and Information Sciences (ICCOINS), 2016 3rd International Conference on*, 2016, no. mputer applications; durgapur; india anirban sark, pp. 489–494.
- [13] J. Muvuna, T. Boutaleb, S. B. Mickovski, and K. J. Baker, “Systems Engineering Approach to Design and Modelling of Smart Cities,” in *IEEE International Conference for Students on Applied Engineering (ICSAE)*, 2017.
- [14] F. Boumahdi, R. Chalal, A. Guendouz, and K. Gasmia, “SOA (Formula presented.): a new way to design the decision in SOA—based on the new standard Decision Model and Notation (DMN),” *Serv. Oriented Comput. Appl.*, vol. 10, no. 1, pp. 35–53, 2016.
- [15] T. Böhm, J. M. Leimeister, and K. Möslin, “Service Systems Engineering A Field for Future Information Systems Research,” *Bus. & Information Syst. Eng.*, vol. 2, pp. 73–79, 2014.
- [16] Q. Gu and P. Lago, “Exploring service-oriented system engineering challenges: A systematic literature review,” *Serv. Oriented Comput. Appl.*, vol. 3, no. 3, pp. 171–188, 2009.
- [17] Suhardi, N. B. Kurniawan, and J. Sembiring, “Service computing system engineering life cycle,” in *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2017, vol. 2017-Decem.
- [18] Suhardi, N. B. Kurniawan, M. I. W. Pramana, and J. Sembiring, “Developing a framework for services computing systems engineering,” in *Proceedings of the 2017 14th International Joint Conference on Computer Science and Software Engineering, JCSSE 2017*, 2017.
- [19] T. Kohlborn, A. Korthaus, T. Chan, and M. Rosemann, “Identification and Analysis of Business and Software Services: A Consolidated Approach,” *IEEE Trans. Serv. Comput.*, vol. 2, no. 1, pp. 50–64, 2009.
- [20] J. Cardoso, K. Voigt, and M. Winkler, “Service Engineering for the Internet of Services,” in *ICEIS 2008, LNBIP*, 2009, no. 19, pp. 15–27.
- [21] Suhardi, R. Doss, and P. Yustianto, “Service Engineering Based on Service Oriented Architecture Methodology,” *TELKOMNIKA (Telecommunication Comput. Electron. Control.)*, vol. 13, no. 4, pp. 1466–1477, 2015.
- [22] J. M. Tien and D. Berg, “A Case for Service System Engineering,” *J. Syst. Sci. Syst. Eng.*, vol. 12, no. 1, pp. 13–38, 2003.
- [23] A. J. Lopes and R. Pineda, “Service Systems Engineering Applications,” *Procedia Comput. Sci.*, vol. 16, pp. 678–687, 2013.
- [24] J. Sum, “Service Systems Engineering: Framework & Systems Modeling,” *Inst. Technol. Manag. Natl. Chung Hsing Univ. Taichung*, vol. 40227, 2014.
- [25] Suhardi, N. B. Kurniawan, J. Sembiring, and P. Yustianto, “Combined Service System Engineering Framework based on Mapping of Service Engineering Components,” in *5th International Conference on Research and Innovation in Information Systems (ICRIIS2017)*, 2017, pp. 0–5.
- [26] Suhardi and N. B. Kurniawan, *Komputasi Layanan dan Sistem Komputasi Layanan*, 1st ed. Bandung: ITB Press, 2018.
- [27] A. T. Zadeh, S. Sahran, and M. Mukhtar, “Automated Service Identification Methods: A Review,” *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 6, no. 6, pp. 1053–1059, 2016.
- [28] A. A. Hamid and Z. Mansor, “Client’s Readiness Assessment Success Factors for Outsourcing Software Projects,” *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 6, no. 6, pp. 820–827, 2016.
- [29] P. Yustianto, Suhardi, and R. Doss, “Consolidating service engineering perspectives,” *2015 Int. Conf. Inf. Technol. Syst. Innov. ICITSI 2015 - Proc.*, pp. 1–7, 2015.
- [30] T. H. Trinh and V. Kachitvichyanukul, “A unified framework for the design of service systems Truong Hong Trinh * and Voratas Kachitvichyanukul,” *Int. J. Serv. Oper. Manag.*, vol. 15, no. 3, pp. 374–388, 2013.
- [31] Suhardi, N. B. Kurniawan, M. I. W. Pramana, and J. Sembiring, “Developing A Framework for Services Computing Systems Engineering,” in *14th International Joint Conference on Computer Science and Software Engineering (JCSSE2017)*, 2017, pp. 1–6.
- [32] N. B. Kurniawan, Suhardi, A. R. Adellina, and P. Yustianto, “Modeling Services Computing Systems toward Services Value Creation,” in *International Conference on Information Technology Systems and Innovation (ICITSI)*, 2017, pp. 348–354.
- [33] A. Kossiakoff, W. N. Sweet, S. J. Seymour, and S. M. Biemer, *Systems Engineering Principles and Practice*, Second Edi. A John Wiley & Sons, Inc., 2011.
- [34] J. Huang, C. Lin, X. Kong, B. Wei, and X. Shen, “Modeling and analysis of dependability attributes for services computing systems,” *IEEE Trans. Serv. Comput.*, vol. 7, no. 4, pp. 599–613, 2014.
- [35] D. Ameller, M. Galster, P. Avgeriou, and X. Franch, “A survey on quality attributes in service-based systems,” *Softw. Qual. J.*, vol. 24, no. 2, pp. 271–299, 2016.
- [36] G. Karokola, S. Kowalski, and L. Yngström, “Evaluating A Framework for Securing e-Government Services – A Case of Tanzania,” in *2013 46th Hawaii International Conference on System Sciences*, 2013, pp. 1792–1801.
- [37] J. Xu and S. Yao, “Performance Evaluation of Service-Oriented Applications Using NHPP Model and Petri Nets,” *Int. J. Serv. Comput.*, vol. 4, no. 2, pp. 16–31, 2016.
- [38] A. Dresch, D. P. Lacerda, and J. A. V Antunes Jr., *Design Science Research: A Method for Science and Technology Advancement*. Switzerland: Springer, 2015.
- [39] F. Aqlan, L. Al-fandi, S. Ramakrishnan, and C. Saha, “A Framework for Selecting and Evaluating Process Improvement Projects Using Simulation and Optimization Techniques,” in *Proceedings of the 2017 Winter Simulation Conference*, 2017, pp. 3840–3851.
- [40] V. Venkatesh and F. D. Davis, “A model of the antecedents of perceived ease of use: Development and test,” *Decis. Sci.*, vol. 27, no. 3, pp. 451–481, 1996.
- [41] W. Muhamad, N. B. Kurniawan, Suhardi, and S. Yazid, “Smart Campus Features, Technologies, and Applications: A Systematic Literature Review,” in *International Conference on Information Technology Systems and Innovation (ICITSI)*, 2017, pp. 384–391.