E-Business Software Product Line Methodology Based on SMEs Characteristics

Ni Made Satvika Iswari[1,3], Eko Kuswardono Budiardjo[1], Zainal Arifin Hasibuan[2]
[1] Faculty of Computer Science, Universitas Indonesia, Depok, Indonesia
[2] Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang, Indonesia
[3] Faculty of Engineering and Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia
[1] satvika@umn.ac.id, [2] eko@cs.ui.ac.id, [3] zhasibua@dsn.dinus.ac.id

Accepted 02 June 2022
Approved 10 July 2022

***Abstract*— The growth of Small and Medium Enterprises (SMEs) will certainly have a positive influence on economic development in a country. However, many SMEs struggle to survive, grow, and show limited productivity. It was recorded that there were around 63.9% of SMEs in Indonesia whose turnover had decreased by more than 30% during the COVID-19 pandemic. Among the factors that cause the low survival rate of SMEs, a factor that is considered critical is the lack of success in the use of e-business. The use of e-business by SMEs is not a one-size-fits-all solution, because SMEs have various characteristics. In this study, an E-Business Software Product Line Methodology based on SMEs characteristics was proposed using Software Product Line Engineering (SPLE) approach. In general, the proposed methodology was sufficient to describe the aspects needed in building a software development methodology. Aspects that had not been described were those related to software project management. The proposed methodology is useful for building an e-business application platform that can be customized based on SMEs characteristics.**

***Index Terms*— Customization; E-Business; Platform; SMEs; SPLE**.

I. Introduction

Despite being a key contributor to the economy in several countries, SMEs experience several challenges caused by several factors, such as social, economic, geographical, cultural, or other factors that are beyond their control. Among the factors that cause the low survival rate of SMEs, a factor that is considered critical is the lack of successful adoption and use of Information Technology [1], or what is known as e-business. The usage of e-business can help to make business procedures more efficient, which can help SMEs flourish. Several studies have stated that IT adoption in SMEs dependent on the size of the company [2]. However, this is contrary to [3], which states that the size of the organization does not guarantee the success of IT adoption. According to the survey, firms that have made IT a habit in their operations will find it easier to adopt IT than traditional organizations.. In addition, the field of services also affects the adoption of IT. Based on the very diverse characteristics of SMEs, an e-business solution is not a one-sized-fits-all solution. The proposed solution needs to consider two main stakeholders in the system to be built, namely SME actors and e-business application providers. In addition to these two stakeholders, a platform provider that connects SME actors and e-business application providers is also needed.

For SME actors, an e-business system is needed, which consists of various kinds of e-business applications according to the characteristics of the organization. Organizational characteristics can be viewed from several main factors, namely Environment, Organizational, and Technology[4]. Based on the organizational characteristics, SMEs can obtain recommendations for e-business applications that can support their organization's business processes. Through the main platform proposed in this study, SMEs can easily get customized e-business applications according to their organizational characteristics. For e-business application providers, providing different e-business applications for each SME is a challenge. If building different e-business applications always starts from scratch, then of course it will consume a lot of resources, both time, cost, and effort. Through the main platform proposed in this study, e-business application providers can customize applications that are built without having to start from scratch. This study is a further development of the Conceptual Model of the Use of E-Business by SMEs [5], using a complementary research model between Design Science Research (DSR) and Behavioral Science Research (BSR) [6].

II. Software Product Line Engineering

SPLE stands for Software Product Line Engineering, and it is a methodology for developing software applications (software intensive systems and software products) that uses platforms and mass customization. The motivation of SPLE is to reduce product line development costs, improve quality, reduce production time, reduce maintenance efforts, overcome product line evolution,

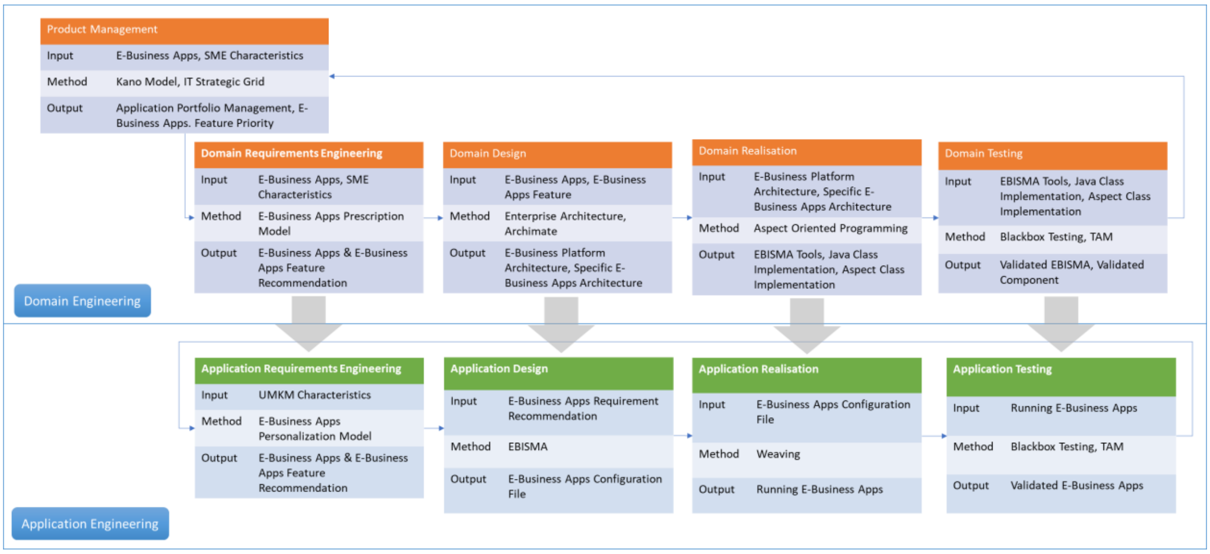
Product Management
Input
E-Business Apps, SME Characteristics
Method
Kano Model, IT Strategic Grid
Output
Application Portfolio Management, E-Business Apps. Feature Priority
Domain Requirements Engineering
Input
E-Business Apps, SME Characteristics
Method
E-Business Apps Prescription Model
Output
E-Business Apps & E-Business Apps Feature Recommendation
Domain Design
Input
E-Business Apps, E-Business Apps Feature
Method
Enterprise Architecture, Archimate
Output
E-Business Platform Architecture, Specific E-Business Apps Architecture
Domain Realisation
Input
E-Business Platform Architecture, Specific E-Business Apps Architecture
Method
Aspect Oriented Programming
Output
EBISMA Tools, Java Class Implementation, Aspect Class Implementation
Domain Testing
Input
EBISMA Tools, Java Class Implementation, Aspect Class Implementation
Method
Blackbox Testing, TAM
Output
Validated EBISMA, Validated Component
Domain Engineering
Application Requirements Engineering
Input
UMKM Characteristics
Method
E-Business Apps Personalization Model
Output
E-Business Apps & E-Business Apps Feature Recommendation
Application Design
Input
E-Business Apps Requirement Recommendation
Method
EBISMA
Output
E-Business Apps Configuration File
Application Realisation
Input
E-Business Apps Configuration File
Method
Weaving
Output
Running E-Business Apps
Application Testing
Input
Running E-Business Apps
Method
Blackbox Testing, TAM
Output
Validated E-Business Apps
Application Engineering

Fig. 1. The Proposed E-Business Software Product Line Methodology based on The Characteristics of SMEs

overcome complexity, increase cost estimates, and increase benefits for customers [7].

There are two development processes in SPLE, namely domain engineering and application engineering[7], which will be explained as follows:

1. Domain engineering process consists of five key subprocesses, namely product management, requirements engineering domain, design domain, realization domain, and testing domain. This process results in a platform that embraces the commonality of application and variability to support mass customization.

2. The application engineering process consists of subprocesses, namely application requirements engineering, application design, application realization, and application testing.

III. DESIGN OF E-BUSINESS SOFTWARE PRODUCT LINE METHODOLOGY

In this study, an e-business software product line methodology based on the SMEs characteristics was proposed using Software Product Line Engineering (SPLE) approach. SPLE has been proven as a methodology for building a wide range of software products and software-intensive systems at lower costs, in less time, and with better quality than other methodologies. The methodology proposed in this study is shown in Figure 1. The proposed methodology adopts the existing process stages in SPLE [8]. In general, there are two processes that exist in this methodology, namely domain engineering and application engineering. The proposed methodology is used to build an e-business application platform that can be customized based on the SMEs characteristics.

A. Domain Engineering

The process in domain engineering is responsible for building a reusable platform, so in this process it is necessary to determine the commonality and variability of the product line being built. In this subprocess, the platform built consists of 2 types, namely the

1. **Information System Platform E-Business**. It is the main platform consisting of various integrated e-business applications. This platform connects e-business application users with e-business application providers.

2. **Application Platform E-Business Specific**, which are applications part of the E-Business Information System. The e-business applications designed can be customized according to user requirements.

Domain engineering has 5 subprocesses, namely product management, domain requirements engineering, domain design, domain realization, and domain testing. At this stage, an analysis related to e-business applications was carried out in accordance with the characteristics of SMEs based on their clusters. The process carried out was to analyze the requirements of e-business applications by SMEs using the Kano Model, as had been done in previous research [9].

1. **Product Management**. In this process, identification of common and variable requirements was conducted to develop portfolio management. The process of E-Business Application Portfolio used survey data that has been carried out at the commonality & variability application-level e-business. At this stage, the results of a survey of e-business by SMEs have been obtained which are evaluated using the Kano Model. The evaluation results of the Kano Model are based on 6 categories, namely attractive, one-dimensional, must-be,

indifferent, reverse, and questionable mapped into four quadrants of the combination of the Kano Model and the Strategic Grid.

2. **Domain Requirements Engineering**. The entire list of requirements that have been collected up to this stage will then be recommended to SMEs based on their organizational characteristics using the Advanced UCF algorithm. Meanwhile, the features of each e-business can be determined based on the preferences of the SMEs requirements. To get preferences for these features, SMEs will be given a pair of functional and dysfunctional questions based on the Kano Model. Features included in the Attractive, One-Dimensional, and Must-Be applications e-business. The features in the classification are features that are considered needed by SMEs and can increase the level of satisfaction of SMEs with the e-business used.

3. **Domain Design**. The E-Business Platform Architecture can be used for applications needed today or can be developed for the development of applications needed in the future. The main architecture is designed to be pluggable, where the implemented e-business can be used according to the characteristics of SMEs. Different service providers can create e-business apps based on their areas of competence. Platform providers, service providers, and SME players are the three stakeholders in this architecture. Platform providers provide as a link between small businesses and service suppliers. The platform provider's service is an e-business application framework that may be used to create e-business apps. Service providers are companies that develop e-business solutions based on their expertise and experience in specific industries. Service providers can create e-business in the proposed architectural design, which is divided into four categories: sales & marketing, manufacturing & production, finance & accounting, and human resources. SMEs, on the other hand, are the end customers of the e-business services offered. SMEs can utilize the profiling tool to get e-business recommendations that fit their characteristics.

4. **Domain Realization**. The architectural design that has been generated in the Domain Design is implemented using the Aspect Oriented Programming. Each SME will be given application recommendations that match the characteristics of its organization. For this reason, we need an application outside of the main e-business that functions to assess organizational characteristics and provide recommendations for e-business that are in accordance with these characteristics.

B. *Application Engineering*

Process on application engineering is responsible for obtaining product line applications from platforms built on the domain engineering. This is done by taking advantage of the variability of the product line and ensuring that the variability aggregation process is correct according to the specific requirements of the application. At this stage, SMEs can get recommendations for e-business applications according to their characteristics. The multi-label classification approach had been carried out in the previous study [10], used to link an SME profile with several appropriate e-business applications.

There are several sub-processes in Application Engineering which will be explained as follows:

1. **Application Requirements Engineering**. In this subprocess, a list of e-businesses apps for SMEs is generated using the Advanced UCF algorithm. In addition, SMEs can also choose features in each e-business using the Kano Model. By using the Kano Model, SMEs are recommended to choose features that are in the Attractive, Must-Be, and One-Dimensional because they will help the organization's business processes and increase user satisfaction.

2. **Application Design**. To get recommendations for e-business that are in accordance with organizational characteristics, SMEs use the EBISMA application (Figure 2) as a tool. The recommendation results obtained from the EBISMA application are in the form of a list of e-business that are able to help SMEs to carry out their business processes. In addition to the list of e-business, EBISMA will also generate file containing the configurations required to produce e-business based on the specified features.

3. **Application Realization**. To get an application that can run based on customized requirements, it is necessary to configure the main class and aspect with weaving technique[11]. Additional features on the website implemented in the aspect. If these features are not needed, they can be removed by changing file the XML configuration. For users who want to use the Online Store Application can determine the combination of features that suit their requirements. By using this approach, different e-business apps can be built from the same platform. These applications are called the Software Product Family.

IV. Implementation and Analysis

In this study, the proposed methodology was implemented to build an application called EBISMA, which stands for E-Business for Indonesian SMEs. This application serves to assess the characteristics of the organization and provide recommendations for e-business applications in accordance with these characteristics. In addition, EBISMA also functions as a configurator that will produce e-business applications that are ready to be used by SME actors. The main view of the EBISMA Application is shown in Figure 2.

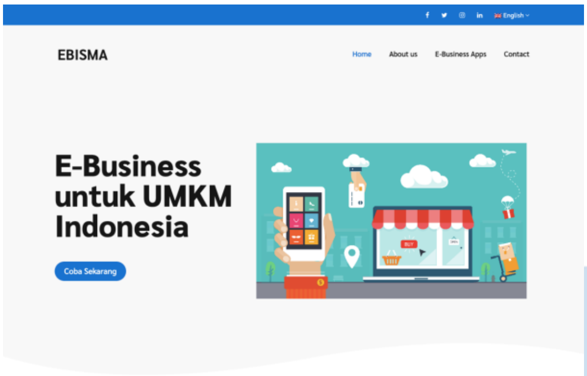

Fig. 2. EBISMA Application Main Page

In using EBISMA, SMEs are asked to answer several questions to assess the characteristics of their organization. The questions are divided into two stages, namely questions related to the identity of SMEs and questions related to the readiness of SMEs to implement e-business. After answering these questions, the system will provide recommendations for appropriate e-business applications. ESBIMA is an application that will generate a configuration file, which contains main classes and aspect classes that will be used to build specific e-business applications. The EBISMA configuration design is carried out using an Aspect Oriented Programming approach, as shown in Figure 3. A more detailed description of the AOP approach for implementing the EBISMA configuration has been presented in previous research [11].

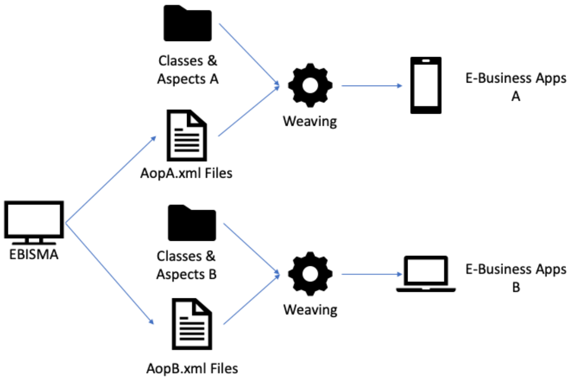

Fig. 3. E-Business Application Design with AOP [11]

V. Conclusions

At this stage, the proposed e-business software product line methodology was evaluated using a general framework for evaluating the software development methodology proposed by [12]. Implementing methodology evaluation framework can provide several advantages. Evaluation can help in understanding certain aspects of a methodology. Evaluation can also be used to compare methodologies, and thereby help users choose between several methodologies according to their desired goals. In addition, improvements can be accelerated using the evaluation results. In this section, the evaluation was carried out by assessing the criteria with a measurement matrix with a value of 0 to 6 based on [13].

In this section, an assessment of the existing process in the methodology is carried out. The proposed methodology does not focus on the management aspects of software development projects, which consist of risk management, people management, quality management, configuration management, and project scheduling. Another thing that is not focused on the methodology is the aspect of efficiency, both in terms of time and resources.

A. Modeling Language Features

A modeling technique is a set of models that describe a system at various levels of abstraction and explain its various aspects. In this section, an assessment is made regarding the modeling language used in the proposed methodology. Based on the results of the assessment, which are shown in **Error! Reference source not found.**, in general the proposed methodology covers most aspects of the assessment in terms of the modeling language. Things that have not been covered in the methodology are aspects of formalization. The proposed methodology uses UML and ArchiMate as its modeling language and the two modeling languages do not provide formalization aspects to symbolize semantics. Another aspect is related to handling inconsistencies that have not been provided by the methodology. The last aspect is related to the User Interface, which is not provided by the methodology.

TABLE I. ASSESSMENT OF MODELING LANGUAGE FEATURES

No.	Aspects	Criteria	Values	Basic
1	*Ease of*	*Understanding*	6	Modeling language using *Archimate* and UML which are easy to understand in general
2		*Usage*	6	Modeling language using *Archimate* and UML which are easy to use in general
3	*Precisenses*		6	Modeling language is considered appropriate and can avoid ambiguity
4	*Power of language*	*Formalism*	0	*Archimate* and UML do not provide formalization aspects to represent semantics
5		*Supported views*	6	*Archimate* includes *structural view*, *behavioral view*, and *functional view*
6		*Support for transformation logic*	6	Model provides design from analysis phase to implementation phase for programming code
7	*Complexity management model*	*Modularity*	6	The model is designed in a modular fashion, which consists of a *java* (*common*) and *aspect* (*variable*).
8		*Handling model inconsistencies*	3	The model is designed with a clear step flow to avoid *overlapping*. However, techniques for dealing with inconsistencies are not discussed
9	*Expressivenes*	*Static and dynamic aspects*	6	UML represents the structural (*static*) and *behavioral* (dynamic) aspects of the system
10		*Physical architecture of systems*	6	Architectural design is built using *Archimate*

B. Process Features

The process in the methodology has two roles, namely regulating and directing development from analysis to implementation, and enabling traceability of development by determining results and milestones. In this section, an assessment of the existing process in the methodology is carried out. Based on the results of the assessment, which are shown in **Error! Reference source not found.**, the proposed methodology does not focus on the management aspects of software development projects, which consist of risk management, people management, quality management, configuration management, and project scheduling. Another thing that is not focused on the methodology is the aspect of efficiency, both in terms of time and resources.

TABLE II. ASSESSMENT OF PROCESS FEATURES

No	Aspect	Sub-Aspect	Criteria	Value	Baseline
1	*Lifecycle*	*Work Unit*	*Coverage of generic phases*	6	The methodology addresses the generic phases of software development (requirements, analysis, design, implementation, testing)
2			*Smooth transition between phases*	6	transition between phases is clear and smooth
3			*Seamless transition between phases*	6	Artifacts produced between phases are quite clear to make a *seamless transition*
4			*Kind of lifecycle*	6	The type of SDLC used is *incremental*
5			*Workflow*	6	Each phase of the methodology consists of different activities and has clear guidelines
6		*Product*	*Adequacy*	6	Each phases of the methodology produce sufficient artifacts to be used in the next phase
7			*Consistency*	6	Each phase of the methodology produces artifacts that are consistent and complement other artifacts with minimum *overlapping*
8			*Supported view*	6	The artifacts from each phase of the resulting methodology consist of *structural*, *behavioral*, and *functional*
9			*Abstract ction levels*	6	Artifacts represent abstraction levels for systems, *packages*, *components*, and *objects*
10			*Tangibility/Testability/ Visibility*	6	Artifacts produced are *tangible*, tested, and understood
11			*Appropriate Documentation*	6	documentation can be generated from the methodology
12		*Role*	*User involvement*	6	Users are involved in all stages of the methodology clearly
13			*Roles specification*	3	*Roles* of users have not been specified in detail
14	*Management*	*Risk management*		0	The methodology does not discuss *risk management*
15		*People management*		0	The methodology does not discuss *people management*
16		*Quality management*	*Quality control*	0	The methodology does not discuss *quality control*

17			*Quality assurance*	0	The methodology does not address *quality assurance*
18		*Configuration management*		0	The methodology does not address *configuration management*
19		*Project scheduling*		0	The methodology does not address *project scheduling*
20	*Development context*	*Ease of*	*Understanding*	6	The development process is easy to understand
21			*Usage*	6	The development process easy to use in the *context* development *e-business software*
22		*Efficiency*	*Time*	0	Methodology does not address time efficiency
23			*Resources*	0	Methodology does not address resource efficiency
24		*Precision*	*Traceability*	6	The resulting artifacts have traceability to requirements
25			*Formalism*	0	The methodology does not address aspects of *formalism*
26			*Expressiveness*	6	Each phase of the methodology is clearly defined to avoid ambiguity
27			*Rationality*	6	Each phase of the methodology is defined rationally
28			*Completeness*	3	Each phase of the methodology is fully defined for all aspects, such as artifacts and processes. However, some aspects such as *roles* and *resources* have not been defined
29		*Maintainability*	*Modularity*	6	The model is designed in a modular manner, consisting of a *java* (*common*) and *aspect* (*variable*)
30			*Reusability*	6	The model is designed in a modular manner and can be reused
31			*Testability*	6	The methodology provides testing phases in *the engineering* and *application engineering*
32		*Complexity Management*	*Evolvability*	6	phase *Product Management* allows for product development
33			*Extensibility*	6	phase *Product Management* also allows for additional product features
34			*Distribution*	6	Methodology supports modeling and implementation of components that support distributed functionality
35			*Integrity*	6	Methodology provides an integrated architecture

C. *Applicability Features*

Another important aspect that needs to be evaluated is related to the application of the methodology in software development projects. In this section, an assessment is carried out on aspects of the application of the methodology, which includes aspects of pragmatic, marketability, and application limitations. Based on the results of the assessment, which are shown in Table 3, most aspects of the application of the methodology have been discussed in detail. Things that are not discussed in this methodology are aspects of team skills and the level of satisfaction of the development team. This is not the focus of the proposed methodology.

TABLE III. ASSESSMENT OF APPLICABILITY FEATURES

No	Aspect	Sub-Aspect	Criteria	Value	Basic
1	*Pragmatics*	*Adaptability*	*Size and complexity*	6	The methodology covers a fairly large and complex process
2			*Criticality*	6	The methodology has *impact* on the success of *E-Business Software Product Line*
3			*Scalability*	6	Methodology can be used on different project sizes
4		*Extant resources*	*Available information*	6	Information related to the methodology is clearly available
5			*Tools*	6	EBISMA *tools* are available to support the methodology
6		*Required resources*	*Team skills*	0	The methodology does not address aspects of *team skills*
7			*Platform suitability*	6	The methodology can be used on different platforms
8	*Marketability*	*User satisfiability*		6	Users are satisfied with the product resulting from the methodology
9		*Development team satisfiability*		0	The methodology does not address *development team satisfiability*
10	*Application constraints*	*Legal constraints*		6	The methodology does not limit legal aspects
11		*Technical constraints*		6	The methodology limits on the use of AOP (*Aspect Oriented Programming*) for design implementation
12		*Management constraints*		6	The methodology does not limit management aspects, such as management culture, etc.
13		*Environment constraints*		6	Methodology does not constrain the development environment

D. *Type-specific Methodology Features*

In this section, an assessment of the specific aspects provided by the proposed methodology is carried out. Based on the results of the assessment, as shown in Table 4, the proposed methodology meets the aspect of a type-specific, where the methodology uses the AOP concept to design and implement the model.

TABLE IV. TYPE-SPECIFIC METHODOLOGY FEATURES

No	Aspect	Value	Basic
1	*Concept-specific*	6	The methodology uses the AOP concept to design and implement the model
2	*Philosophy*	6	The methodology uses the AOP concept because AOP considers modularization, encapsulation, and *crosscutting concerns* as the main focus

Based on the results of all methodological assessments, in Table 5 summary of the results of the methodology assessment is presented. In general, the proposed methodology is sufficient to describe the aspects needed in building a software development methodology. Aspects that have not been described are those related to software project management. This aspect is a limitation in this research.

TABLE V. SUMMARY OF E-BUSINESS SOFTWARE PRODUCT LINE METHODOLOGY ASSESSMENT

Aspect			Assessment
Generic features	Modeling language		Partial coverage
	Process	Lifecycle	Partial coverage
		Management aspects	No coverage
		Development context	Partial coverage
	Applicability		Partial coverage
Specific features	Concept-specific		Acceptable coverage

VI. CONCLUSIONS

In this research, an e-business software product line methodology was produced. The research contribution refers to several issues for further development of the software product line engineering [7], which consists of domain specialization, software product line evolution, multiple product lines, and tool support. The applicative research contribution from this research is that SMEs can get recommendations for e-business applications that are in accordance with the characteristics of their organizations. In addition, the results of the analysis of commonality and variability can be a reference for e-business application developers to make the transition from single application development to product family development. E-business application developers can determine the priority of application development based on the commonality and variability analysis. E-business application developers also can take advantage of the e-business platform architecture to develop services that match their expertise.

The following will explain in more detail the research contributions which refer to several issues that are issues of further development of the Software Product Line Engineering[7].

A. *Domain Specialization*

The biggest challenge for research in Software Product Line Engineering (SPLE) is to specialize framework for an application domain. By specifying the SPLE framework for an application domain, a series of types can be generated to model domain-specific variation points, variants, variability dependencies, and constraint dependencies. To take full advantage of domain specialization, it is necessary to define a domain-specific modeling language to define software development artifacts [7].

In this study, the proposed SPLE methodology is devoted to building a Software Product Line for e-business domain for SMEs. To be able to support the use of e-business by SMEs, several e-business apps is needed that function to assist SME business processes for certain functional areas. Sales & marketing functional areas need e-business in the form of Online Store, E-marketplace, etc. Meanwhile, in the human resources functional areas, e-business applications are needed in the form of payroll, management personnel, etc. The use of the e-business will certainly help SME business processes. Therefore, in this study, a recommendation model for e-business applications was produced based on the characteristics of SMEs. Based on the results of these recommendations, SMEs can use e-business applications that are in accordance with their business characteristics.

B. *Evolution of the Software Product Line*

In the development of the Software Product Line, software developers are not only faced with the evolution of the Software Product Line over time, but also with the existence of different variants at the same time (variability in space). Managing the evolution of Software Product Line artifacts over time and ensuring the consistent integration of changes in all affected Software Product Lines is a key research challenge. Developing and validating comprehensive techniques that support the management of Software Product Line artifact evolution over time and the management of variability in artifacts is also a research issue open to further study [7].

In the Product Management subprocess of the proposed E-Business Software Product Line Development Methodology, a product roadmap has been generated, which is represented in the form of an application portfolio. In the application portfolio, it can be determined which e-business applications are useful for the present and which e-business applications will be useful in the future. The resulting application portfolio combines the McFarland matrix with the Kano Model. By combining these two theories, it was hoped that it will provide appropriate results from grouping e-business applications, both in terms of user preferences and in terms of their contribution to business processes. In addition, the results of the combined analysis of McFarlan and Kano will support the evolutionary management of the E-Business Software Product Line.

C. *Multiple Product Lines*

In some domains, there is a need to manage variability across product lines. However, solutions for defining and managing variability across product lines and across all software development artifacts remain unclear. Managing variability across product lines is even more challenging if the product lines are owned by different companies [7].

In the Product Management stage of the proposed E-Business Software Product Line Development Method, an e-business application portfolio has been obtained which functions as a product roadmap, which is used for application planning and management based on current and future contributions to the business. In addition to the e-business application portfolio, at the Product Management stage, priority features will be developed for e-business applications. The priority of feature building starts with basic features, then performance features, entertainment features, and ends with potential features.

D. *Tool Support*

High-quality software engineering in the industry requires adequate tool support. However, for most aspects of Software Product Line Engineering, adequate tool support does not exist. For example, support for tools used to manage variability across all development artifacts, or to manage linkages between product line applications and domain artifacts is very weak. Building seamless tool support to manage variability and development artifacts in process domains and application engineering is a challenge for future research [7].

In this study, a prototype was built named EBISMA, which stands for E-Business for Indonesian SMEs. This application serves to assess the characteristics of the organization and provide recommendations for e-business applications in accordance with these characteristics. In addition, EBISMA also functions as a configurator that will produce e-business applications that are ready to be used by SME actors. In EBISMA application, SMEs are asked to answer a number of questions to assess the characteristics of their organization. The questions are divided into two stages, namely questions related to the identity of SMEs and questions related to the readiness of SMEs to implement e-business. After answering these questions, the system will provide recommendations for appropriate e-business applications using the Advanced UCF algorithm.

REFERENCES

[1] M. Bernaert, G. Poels, and M. Snoeck, Information Systems for Small and Medium-sized Enterprises. 2014. doi: 10.1007/978-3-642-38244-4.

[2] V. A. Mabert, A. Soni, and M. A. Venkataramanan, “The impact of organization size on enterprise resource planning (ERP) implementations in the US manufacturing sector,” Omega (Westport), vol. 31, no. 3, pp. 235–246, 2003, doi: 10.1016/S0305-0483(03)00022-7.

[3] P. O. H. Putra, H. B. Santoso, and Z. A. Hasibuan, “Antecedents and patterns of e-Business adoption among small and medium enterprises (SMEs),” 2017 Second International Conference on Informatics and Computing (ICIC), pp. 1–6, 2017, doi: 10.1109/IAC.2017.8280638.

[4] P. O. H. Putra, Z. A. Hasibuan, and P. O. A. Sunarya, “Depicting E-business application adoption for small and medium enterprises (SMEs): The e-business triangle model,” 3rd International Conference on Computing, Engineering, and Design, ICCED 2017, vol. 2018-March, pp. 1–6, 2018, doi: 10.1109/CED.2017.8308135.

[5] P. O. H. Putra and H. B. Santoso, “Contextual factors and performance impact of e-business use in Indonesian small and medium enterprises (SMEs),” Heliyon, vol. 6, no. 3, p. e03568, 2020, doi: 10.1016/j.heliyon.2020.e03568.

[6] A. Hevner and S. Chateerjee, Design Research in Information Systems : Theory and Practive, vol. 28. 2012. doi: 10.1007/978-1-4419-6108-2.

[7] K. Pohl, G. Böckle, and F. van der Linden, Software Product Line Engineering. Foundations, Principles, and Techniques. New York: Springer-Verlag Berlin Heidelberg, 2005. doi: 10.1007/978-3-642-36583-6_1.

[8] K. Pohl, G. Böckle, and F. van der Linden, Software Product Line Engineering: Foundations, Principles, and Techniques. 2005.

[9] N. M. S. Iswari, E. K. Budiardjo, and Z. A. Hasibuan, “E-business applications recommendation for SMES using advanced user-based collaboration filtering,” ICIC Express Letters, vol. 15, no. 5, pp. 517–526, 2021, doi: 10.24507/icicel.15.05.517.

[10] N. Made, S. Iswari, E. K. Budiardjo, H. B. Santoso, and Z. A. Hasibuan, “E-Business Application Recommendation for SMEs based on Organization Profile using Random Forest Classification,” 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp. 522–527, 2019.

[11] N. Made Satvika Iswari, E. K. Budiardjo, and Z. A. Hasibuan, “Aspect oriented programming approach for variability feature implementation in software product line engineering,” 2020 5th International Conference on Informatics and Computing, ICIC 2020, pp. 4–8, 2020, doi: 10.1109/ICIC50835.2020.9288558.

[12] S. Hesari, H. Mashayekhi, and R. Ramsin, “Towards a general framework for evaluating software development methodologies,” Proceedings - International Computer Software and Applications Conference, pp. 208–217, 2010, doi: 10.1109/COMPSAC.2010.69.