Adapting Software Factory Approach into Cloud ERP Production Model

Journal of Computer Science IJCSIS, Muhamad Abu Bakar

Adapting Software Factory Approach into Cloud ERP Production Model

Dzulkafli Jalil
School of Computing,
Universiti Utara Malaysia, 06010 UUM Sintok,
Kedah, Malaysia
jay@smartlab.com

Muhamad Shahbani Abu Bakar
School of Computing,
Universiti Utara Malaysia, 06010 UUM Sintok,
Kedah, Malaysia
shahbani@uum.edu.my

Abstract - **Cloud ERP has introduced a more convenient way of acquiring the ERP software solution. With ready on demand, modular and multi-tenant models, customers are able to acquire ERP solution at lower cost and with simpler processes and steps. However, for ERP software providers, this scenario has created a more complex responsibility. The challenges of developing and managing cloud ERP can be broken down into the following; among others are, Multi tenancy solution, Software variant by customization dynamic request, Continuous software updates to keep up with technologies and business practices, modularity features, resource provisioning and security management. All these challenges introduce high workload in operational and technical knowledge which tag along with company workforce. The problem of managing high amount of knowledge content is more troublesome if we consider the inevitable fact of staff turnover. Cloud ERP as a service model does not seem to fit any present model of software development lifecycle as it is rather a combination of software development and outsourcing lifecycle model. The question is how to develop a model that can reduce the workload and the complexity of Cloud ERP production which is also resilient to staff turnover. Using Software Factory approach has been proven to be a successful approach to manage complex software management, we introduce a Cloud ERP Factory (CEF) model that can provide a systematic approach of managing ERP lifecycle with knowledge retaining feature. The Model is made up by five main components those are ERP product line, Platform, Workflow, Product Control and Knowledge Management. The model provides a guide to a systematic development and production of the Cloud ERP for commercial environment. In validating the model, it simulated into a real industry environment to test for its usability. The research incorporates Work system, Software Product Line and Resource Based Theories into the proposed model. The CEF model is intended as guide to develop a commercial Cloud ERP production environment.**

Keywords — ***Enterprise Resource Planning, ERP, Service Oriented Architecture, Cloud ERP, Software Factory, Software Product Line.***

I. INTRODUCTION

Enterprise Resource Planning (ERP) is undoubtedly a critical component of any business operations, thus making it almost a mandatory requirement to set up a business [1]. It refers to an enterprise business strategy and a set of industry-domain-specific applications that promotes customer and shareholder value by enabling and optimizing collaborative operational and financial processes [2]. The term ERP is initially defined by Gartner in 1990 and later revised the term to ERP II in 2000 which expanded the scope to include almost every business facets within an organization. Figure 1 illustrates a typical commercial ERP package.

Figure 1: Typical Commercial ERP package.

In the context of this research, the term ERP is referred to all systems within an organization such as Financial Management System (FIS), Human Resources Management (HRMS), Manufacturing Resource Planning (MRP) as well as systems with external interaction to other system such as CRM and SCM [3].

In attempt to reduce cost, ERP software providers have been trying to standardize their ERP packages into standard common modules with configuration capabilities [4], [5]. Over the years, with the knowledge gained from previous implementations, ERP vendors have been trying to come up with the configurable standard version of ERP software which has potential to be delivered to future clients [6].Using configuration features, it is hoped that customer's requirement can instead be solved with configuration options rather than with software code customization. Theoretically, future clients of the same business nature should be able to adapt to the standard modules. However, due to reasons such as technology progression and business model uniqueness, there seem to be fitting problems with the standard ERP software to simply being implemented at any new [7].

This after all, is the main struggle of ERP companies which is to keep updating new software versions in order to make it more adaptable to future clients. Unfortunately, recent research has indicated that, even ERP software solutions such as SAP and Oracle those are with more than half centuries of updated software revision, customization during the new implementation is inevitable [8]. So, despite the highly configurable ERP packages, code base customization remained a major requirement for most ERP implementation [9]. Besides, ERP customization also reflects that every business has its own unique business model which contributes to the organization's competitive advantages [7].

The emergence of the Internet in previous decades has tremendously affected the ERP solution technology and its deployment methods [10]. The gaining momentum of Internet Cloud technology provides a natural progression of ERP applications into a Cloud model [11].Traditional mainframe and client server computing model has been made obsolete with the new web-based technology [12]. ERP adaptation into cloud computing model has shaped the cloud model into a few models aptly named private cloud, public cloud and hybrid cloud [13]. When the solution is accessible only within privately managed network such as on premise solution, it is called private cloud. Public cloud solution is the delivery of ERP package over public cloud such as Amazon, Google, Apple Cloud etc. Hybrid cloud model is the solution that allows the bridging the ERP over private and public cloud [14].

Cloud computing model with Service Oriented Architecture (SOA) approach has introduced new promising technology such as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). With SaaS model, ERP is termed as Cloud ERP and it is expected to inherit the advantages of cloud based solution such as 'on demand capability', scalability and maintainability and lower Total Cost of Ownership (TCO) for customers. The Cloud's Software as a Service (SaaS) technology has turned the ERP into a system that is expected to be on demand in nature [15]. And in order to leverage the Cloud economic of scale potential, Cloud ERP system should be in multi-tenancy mode [16]–[20]. With a multi-tenancy model, all users will be using a single version of software instance, thus making customization much less flexible compared to the single tenancy model [20], [21]. Fortunately, when the cost is less of a concern, single-tenant approach is still feasible as it can provide more customization flexibility. However, this solution creates a potential management challenge to deal with the complexity of managing different software instances or versions. Within the two extreme approaches, there is a multi-instance model in which a tenant get its own software or database instance within the Cloud ERP hosting [22].

Another approach is the multi-tenancy model with composite add-ons variant [23]. Figure 2 below illustrates how Cloud ERP single tenancy differs from that of Multi Tenancy model.

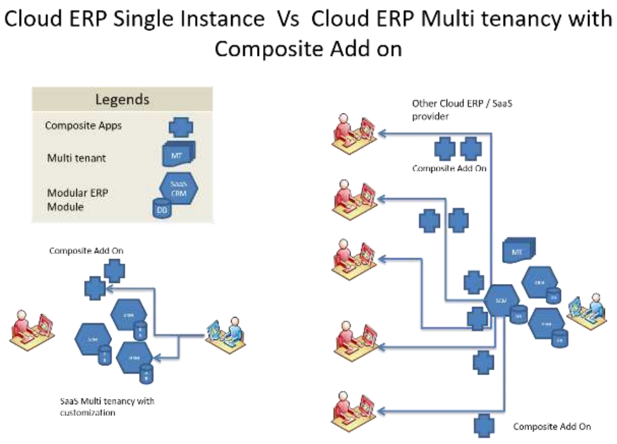

Figure 2: Cloud ERP Single Instance vs Cloud ERP Multi-tenancy with Composite Add-ons

In order to reduce the manageability complexity of customizations, we introduce a paradigm approach adapting from Software Factory. The Software Factory term has been coined since 1975 [24]. In their terminology, it is an integrated set of tools that supports the concepts of structured programming, program development, and program production libraries, and incorporates hierarchically structured program modules as the basic unit of production [24]. Today, Software Factory is a part of Software Product Line in Software Engineering field of study which is referred to an approach that configures extensive tools, processes, and content using a template based on a schema to automate the development and maintenance processes for variants of products by adapting, assembling, and configuring framework-based components [25]. In a simpler term, software factory is an approach of software development life cycle to mimic factory manufacturing processes.

II. MOTIVATION

Cloud ERP has promised to be a more affordable solution of acquiring an ERP [26].It delivers a simpler way of implementing the system without spending high cost on infrastructure [27], [28]. The on-demand feature of the Cloud services also hinted a more scalable model ever to customers [29]. At least for Enterprise SaaS application, customer can now order online a ready system to be deployed in a shorter time compared to that of the traditional ERP [27]. The simplification of acquiring from customer side does not come free. The ERP developers or ERP vendors are now facing a more complicated system to manage [30]. Technically, simplification or the automation on the customer side has shifted a burden of creating and managing an on-demand system in delivering and maintaining the Cloud ERP software [28], [29].

The nightmare of Cloud ERP management from vendor's view is further worsened by the fact that most customers could not simply settle with the supposedly proven configurable system [8], [27], [31]. They want to add more functionality to the system or in a way to customize it. In fact, they want to do it regularly not just one time. Even with a system which allows high configuration features, customers' business processes are dynamic and ever changing thus customization and updates are still required [32].This in turn poses a big challenge to ERP providers in developing and maintaining the software.

The challenge is then obviously greater, if we consider that most Cloud software is expected to be a multi-tenancy with single code version model for it to achieve economy of scale benefit [17], [19], [20]. Therefore, Cloud ERP with multi-tenancy customization model possess more serious challenge for the researchers and ERP practitioners [21].

The complexity problem in the back-end Cloud ERP management is also related to technology and industrial business process advancement that require frequent cloud software version updates [33], [34].

The on-demand nature of Cloud ERP enables customer to subscribe ERP solution by module. Customer can choose to add another module to be integrated with the existing modules. This ability is made possible by the software modularity features of Cloud ERP modules in which Cloud ERP is typically made of a combination of SaaS instance. The modules need to be loosely coupled and yet can be integrated when being deployed. This is another factor that causes complexity for the Cloud ERP back end management [35], [36].

As many software programmers and analyst are involved is developing ERP software, the concern about software quality consistency is real [37].Without proper measuring structure, there will be different software quality among the modules as programming and knowledge skill varied between programmers. Experience programmers will tend to produce a better software quality in term of functional and structural specification than that of junior programmers [38].

Since Cloud ERP product itself is about providing a business solution to enterprises, business process knowledge will be part of the company asset along with computing and programming knowledge. Technically it reflects a vast amount of knowledge pool that resides within the business operation that introduces knowledge management challenge. As this is a typical knowledge-based company, another problem is without a systematic management model, this knowledge is tagged along with the workers. Over dependency on knowledge workers could be a serious problem as inevitably the workers will leave the organization thus introducing knowledge gap in the Cloud ERP production. While one of the challenge is how to manage vast amount of knowledge, retaining the knowledge against the staff turnover is another challenging [39]. In summary, from ERP software provider perspective, Multi Tenancy feature, dynamic customization request, frequent software version updates and keeping modularity features of Cloud ERP have contributed to the high complexity and heavy workload of Cloud ERP production processes. In addition, craftsmanship effect due to varying staff skill creates functional and structural software quality problem faced by the Cloud ERP provider. Finally, due to vast amount of knowledge and over dependency on knowledge workers, knowledge management and retention poses another big problem to the Cloud ERP providers.

III. CLOUD ERP PRODUCTION CHALLENGES

Multi Tenancy, customization, software updates, software modularity, resource provisioning, security measures, knowledge management and inconsistence software quality. These are important features of Cloud ERP that make it vary from traditional way of software or ERP industry. Obviously, Cloud ERP management is not just a software development lifecycle. It is rather a software development solution that sits in an outsourcing business lifecycle. Although it is similar to SaaS solution, ERP as a service typically offer multiple SaaS solution that is well integrated when it is delivered. Figure 3 shows the difference between ERP as a Service compare to SaaS. Technically, most of the previous researches mentioned above did not address the ERP as a modular SaaS solution. In reality as well as in this research, we acknowledge that ERP is a modular SaaS solution thus making it much more complex to manage.

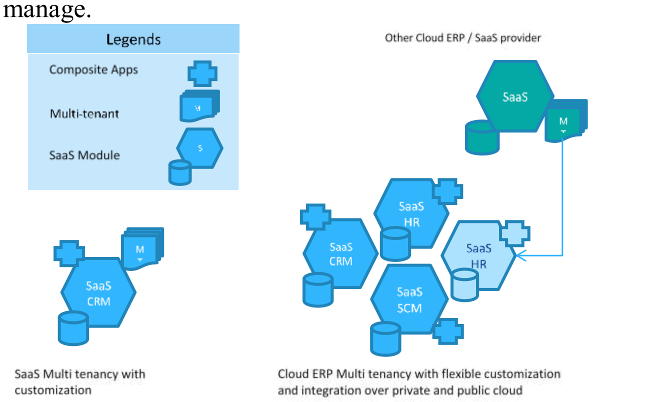

Figure 3: Cloud ERP with customization as compared to SaaS solution.

While all the researches mentioned above did address to their objectives, they did not address the overall view from back end management issues. What we need now is a model that provides a systematic management reference which can indirectly provide solution to the list of problems above.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

Previous researches with Software Factory approach has proven successfully in solving similar problem in various contexts. Table 1 list down 18 studies about Software factory in achieving industrialization software production targets. Group one researchers defined and demonstrated Software Factory in general view, Group two in educational environment, Group three is solving problem from within the company while Group four is solving the problem in software production company.

Group	Authors	Grouping	Limitation/Relevancy
1	(Harvey Bratman & Court, 1975; Canuel & Robert, 2012; Fernstrom, Narfelt, & Ohlsson, 1992; Helton, 2010; Nomura, Tonini, Hikage, & Tonini, 2007)	General	• Provides definition of Software Factory • Addressing Software Development methodologies • Did not cover Cloud ERP environment
2	(Harvey Bratman & Court, 1975; Canuel & Robert, 2012; Fernstrom et al., 1992; Helton, 2010; Nomura et al., 2007)	Educational Purpose	• Application of SF into classroom and educational settings • Did not cover Cloud ERP environment
3	(Comitz & Pinto, 2006; Piho, Tepandi, Roost, Parman, & Puusep, 2011; Siy et al., 2001; Thoreson, Chief, Company, Corporation, & Louis, 1989)	MIS Solution	• Adaptation of Software Factory in corporate environment • Management Information System view • Internal development model • Did not cover Cloud ERP environment
4	(Greenfield & Short, 2003a; Li, Li, & Li, 2001; Lim, Ang, & Parvi, 2000; Rockwell & Gera, 1993)	Software Vendor	• Adaptation of Software Factory in software production for software vendors • Mass customization approach • Cover outsourcing model • Did not cover Cloud ERP environment

Table 1: Software Factory Model & Adaption Grouping

Although all researchers listed in Table 1 may have variation in their Software Factory concepts and models, their basic principles are similar that are mainly to provide a systematic approach of developing software by reducing human craftsmanship effect and promote reusability. Researcher from Group four provides the most relevant Software Factory insight from software developers' view. However, their solutions are intended for traditional software production model not Cloud SaaS model or ERP model thus making Software Factory for Cloud ERP model is lacking in previous researches.

IV. PROPOSED SOLUTION

In response to the research questions, we identify Software Factory approach as a proven model to tackle the problem statements mentioned above.

Software Factory is defined as a structured collection of related assets that aids in producing computer software applications or software components according to specific, externally defined end-user requirements through an assembly process [40]. A software factory applies manufacturing techniques and principles to software development to mimic the benefits of traditional manufacturing [24]. Software Factory (SF) is widely understood as a model which mimics the efficiency of manufacturing process flow in handling software development process [24]. [40] then reiterates the model by defining a shift of focus from just supporting code production and documentation to the information flow between vendors and clients so the correct information is available at the right time. The SF model has since evolved into a detailed development environment created to manage application development lifecycle [25].

A research suggests that by harnessing meticulous workflow described in SF, managing the software development process can be achieved with factory-like precision [41].The success of adopting the methodologies discussed in SF is totally dependent on the ability of the adjoining methods and tools that makes up the whole model [41].One of the most common methodologies linked with SF is work system theory which describes a connection where human capital and machines perform work utilizing necessary resources available to create specific products/services [42]. Resource based theory, another tool used by SF simply states that the resources available is a precursor to the ability of an organization to achieve competitive advantage and a superior long-term performance [43]. However, SF is merely a model and to be able to prove its adoption success is a challenge on its own [41]. In this context, we believe that the Software Factory approach would be the logical solution to be adopted to generate a Cloud ERP model from software developers' perspective which is able to improve the overall management complexity of Cloud ERP system.

Apart from focusing on Software Factory approach model, we have also identified theories in which to be included into the models namely work system theory and resource base theory. Work System theory a natural unit of analysis for thinking about systems in organizations. In organizational settings, work is the application of human, informational, physical, and other resources to produce products/services [42]. In this research, Work System Theory will be used to represent architectural and workflow model. Resource-based theory on the other hand is a basis for the competitive advantage of a firm lies primarily in the application of a bundle of valuable tangible or intangible resources at the firm's disposal [43].

Figure 4 below shows the Theoretical Framework was finalized during the development phase in order to develop the model.

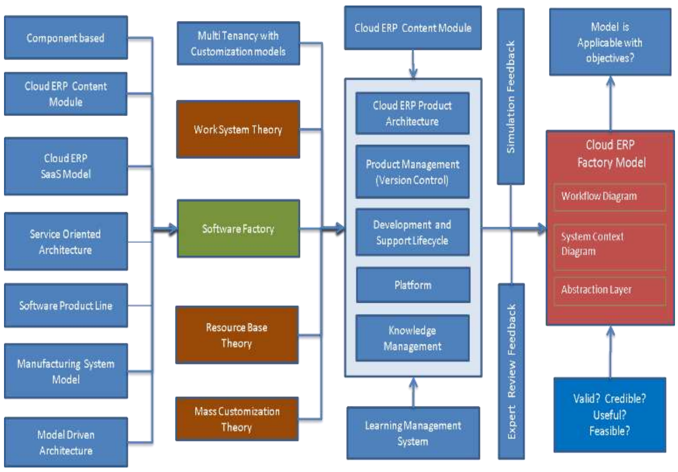

Figure 4: Theoretical Framework for Cloud ERP Factory Model

Figure 5 shows the early proposed model that make up the components and subcomponents of Cloud ERP Factory. Base on this model, we have constructed a more comprehensive model and its sub-models and finally develop the software system from the models for simulation and validation purposes.

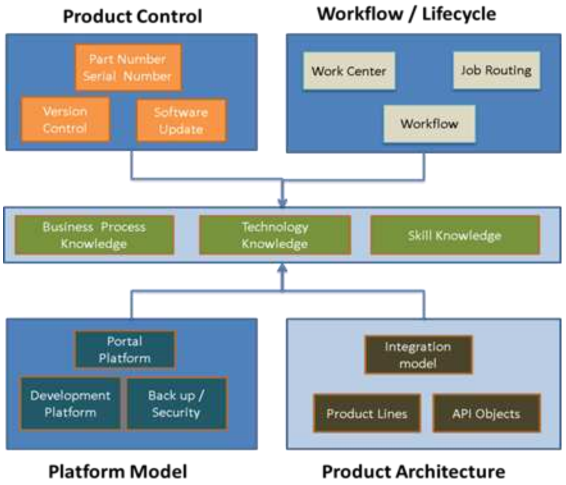

Figure 5: Proposed Cloud ERP Factory Model

Cloud ERP Factory Model Components:

The CEF model consists of five sub-components which are **Software Product Line, Platform, Workflow, Product Control** and **Knowledge Management**. The sub-components are broken down model from the main CEF Model. They are segregated in such a way to create focus in their own domain. Each sub-component has its own objectives, core elements, guidelines and checklists which can be replicated by other Cloud ERP providers to implement the CEF Model. Table 2 below lists the objectives and core elements for each CEF Model Components.

CEF Model Components	Objective	Core Elements
Product Line	Modularity, Integration, Product Family Group	System Module, Module Components, Integration Model
Platform	Reusability, Layer, Standard Interface	User Portal, Software Platform Architecture, Security/Backup
Workflow	SOP, Segregation, Time Frame	Ticketing, Workflow, WO Routing
Product Control	Quality Control, Version Control, Anti-Infringement	Part Number, Serial Number, Version Control, Software Update
Knowledge Management	Repository, Retrieval & Dissemination, Knowledge Gathering	Knowledge identification, Retrieval, Knowledge Logging, Knowledge Mapping

Table 2: CEF Model Components Summary

Software Product Line Model

Product Architecture model development for Cloud ERP Factory is modelled after Software Product Line body of knowledge. Since ERP covers the entire business activities in the software form, one have to clearly define the product range and scope that will be built by the set-up factory. Just like when producing a physical product in a traditional manufacturing factory, we have to clearly identify the product family that we will manufacture. By defining Cloud ERP product line, we will know the manufacturing capability requirement of the factory. Figure 6 below shows how the Software Product Line model can be used to dictate the product group/family.

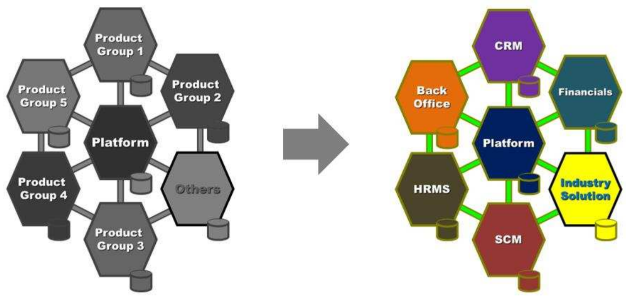

Figure 6: Software Product Line Model

Platform Model

Platform Architecture is a mandatory reusable component of Cloud ERP Product Architecture. The platform is defined as a common foundation of the Cloud ERP Product which provides a base as reflected in Figure 7. The model introduces two types of platform; user portal platform and software architecture platform. Each type of platform carries a significant importance to the CEF model as it ensures the type of product to be released.

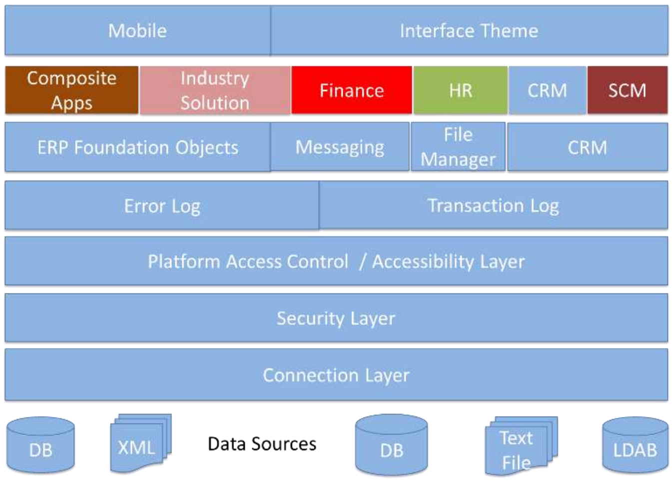

Figure 7: Software Architecture Platform Model

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

Workflow Model

Workflow model for the Cloud ERP Factory basically simulates the standard manufacturing process flow. Referring to Figure 8 below, it can be deduced that the Work centre in CEF represents the machinery in standard factory environment which translates into specific jobs and tasks. Workflow model for the Cloud ERP Product can be considered as the engine as each task would be defined with specific checklist of tasks. The important outcome for the Workflow management is to plan the programmer's daily workload. Instead of ad-hoc requests, programmers can rely on the list of jobs specified to be completed accordingly.

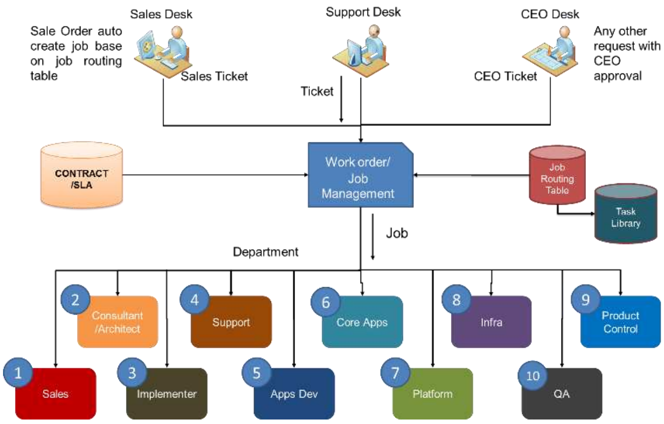

Figure 8: Workflow Management Model

Product Control Model

Product Control is essential in the Model development of CEF Factory model. Product control represents the ability to ensure all the products released by the factory follows stringent step by step procedures as illustrated in Figure 9. The model should take into consideration that each software product is registered as well as the ability to monitor the different version or revision done on the specific product based on client's need or request. Activation key and product serial number activation are included in the model to emphasize the security aspect of product control.

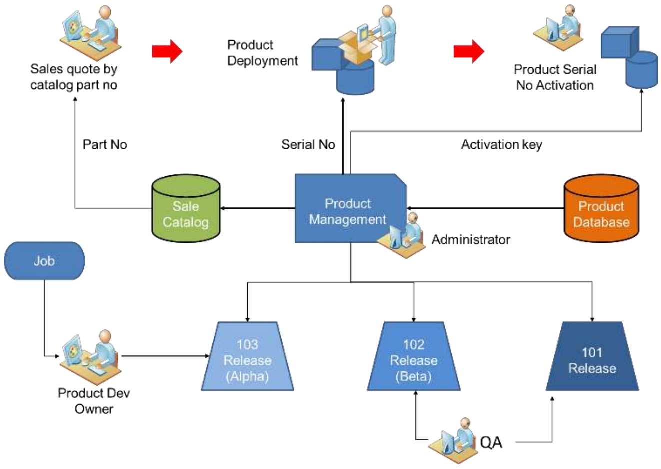

Figure 9: Product Control Model

Knowledge Management

As the Figure 10 below shows, the Knowledge Management model is greatly dependent on the ability of the system to pool the necessary resources and matches it with the needs of the department within the organization. Each department will require different sets of courses to be imparted to their respective staff. With the On-Demand concept, staff/employee will be given the ability to choose and undergo the training when it is required.

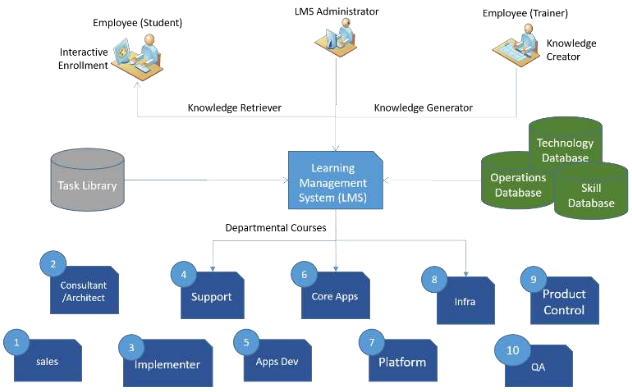

Figure 10: Knowledge Management Model

V. MODEL VALIDATION

The proposed CEF model has been validated through expert reviews process which is the first line of validation process. In order to further validate the model in real commercial environment, we have simulated the working Cloud ERP production environment to fit the CEF model guidelines. In doing so, the CEF model and its sub components are translated into software application that represents each sub-model. The simulation of the model is heavily dependent on the organization size and human resources available. Figure 11 below clearly indicates the segregation of tasks within the

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

designated department within the overall software production cycle. The process flow from each work center designates a form of controlled procedures to be followed stringently in order for the prototype to be usable.

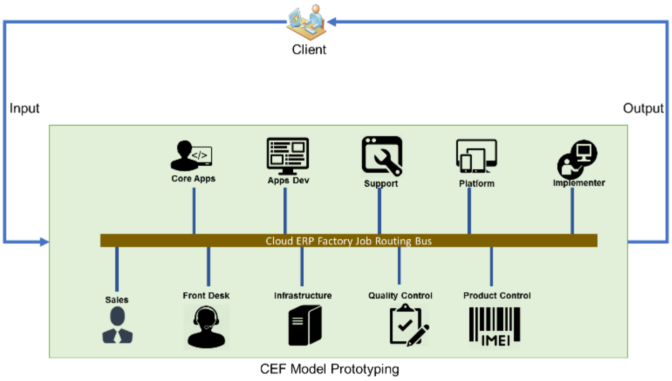

Figure 11: CEF Model Prototype

Figure 12, 13 and 14 shown below are some of the prototype modules developed to achieve the CEF model objectives. Each prototyped module manages its own designated function and provides the necessary procedures to allow a Cloud ERP production environment to have a more systematic and factory-like precision.

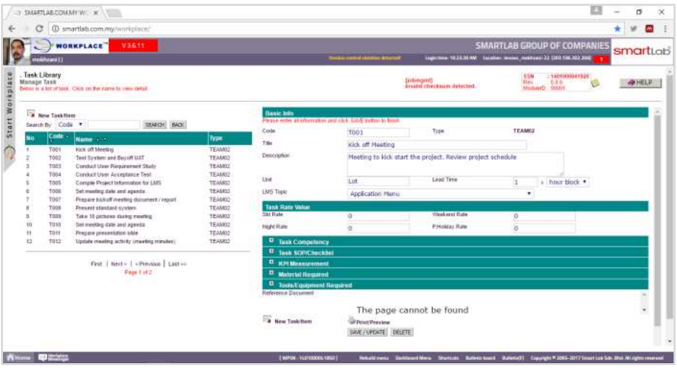

Figure 12: Job Task Checklist

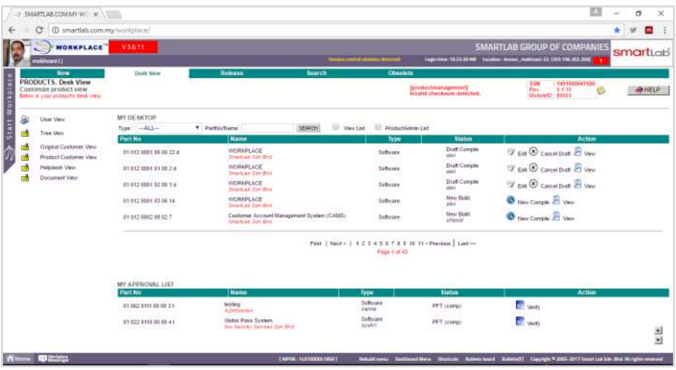

Figure 13: Product Registration Module

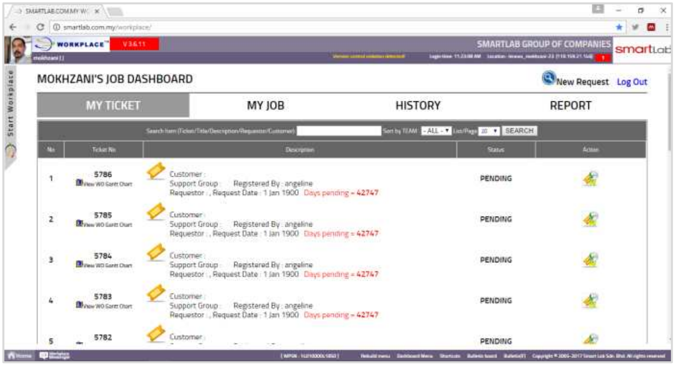

Figure 14: Request/Complaints Ticket Dashboard

VI. FIELD DEMONSTRATION & FINDINGS

Once the prototype is successfully verified, we have conducted three separate concurrent field demonstration to validate its usability in a real software deployment environment. The Figure 15 shows that in this research, each field demonstration is provided with a survey as to the impact of the implemented CEF with regards to its usability based on the Perceived Usefulness and Ease of Use (PUEU) and Usefulness, Satisfaction and Ease of use (USE) questionnaires sample with a Likert Scale rating options.

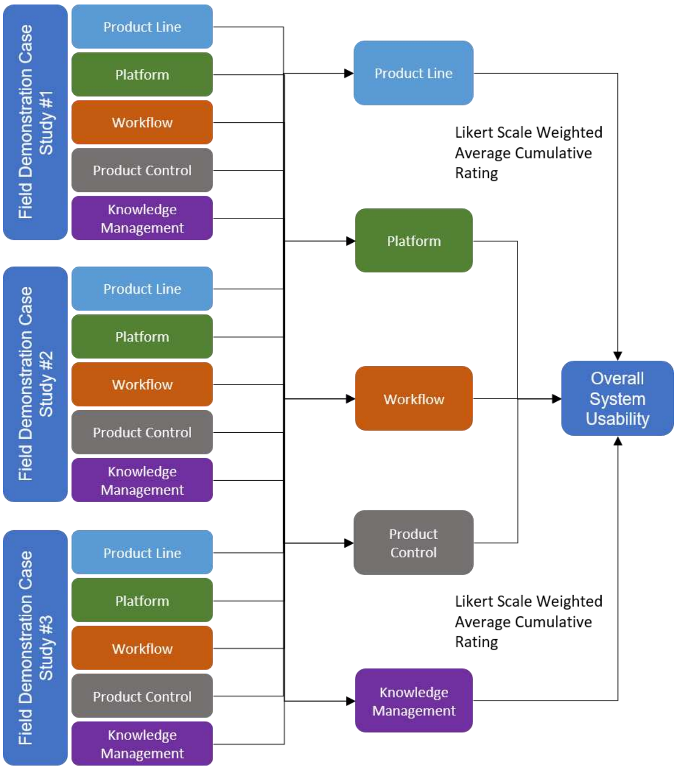

Figure 15: Field Demonstration Usability Test

The PUEU and USE questionnaires consists of 12 and 30 questions respectively which focuses on the perceived usefulness of the implemented system for the designated players for each sub component. The questionnaires were distributed in two sessions, first before the CEF is implemented, and the second session once the CEF has been implemented. In below Table 2, all the data is tabulated based on the five sub component elements.

Before Implementation	FD #1	FD #2	FD #3
Product Line	2.3	3.1	2.4
Platform	3.0	3.5	3.7
Workflow	2.4	2.2	3.1
Product Control	3.0	3.0	2.0
Knowledge Management	1.0	2.0	1.0

After Implementation	FD #1	FD #2	FD #3
Product Line	5.5	4.8	6.2
Platform	4.1	5.3	6.7
Workflow	5.3	6.3	6.6
Product Control	4.0	5.1	4.7
Knowledge Management	4.4	5.1	4.1

Table 2: Tabulation of Survey Response to CEF Implementation in Field Demonstration Case Studies

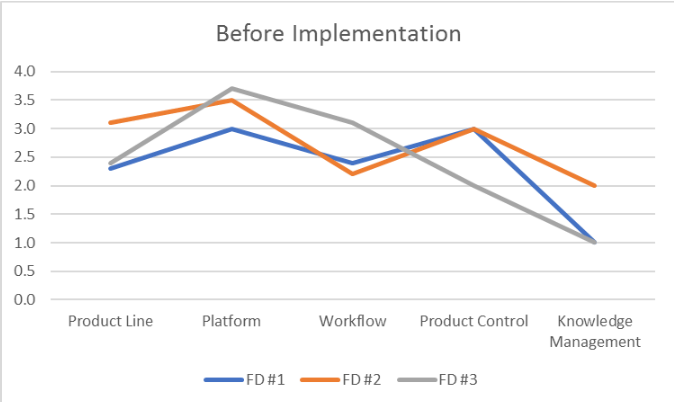

Figure 15: Field Demonstration Usability Test

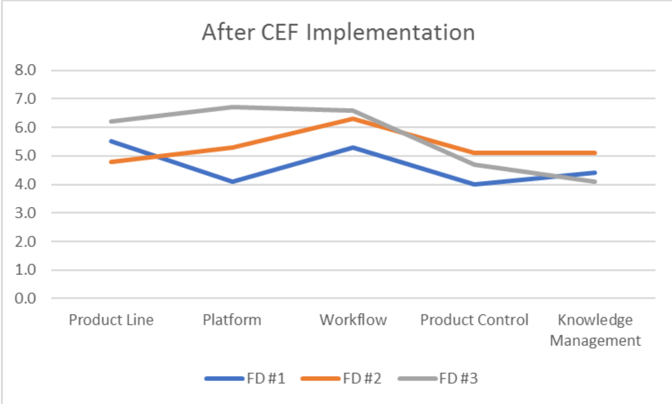

Figure 16: Field Demonstration Usability Test

From the Figure 15 & 16 derived from the Table 2, we were able to verify that by implementing the CEF model in an existing Cloud Software production environment, there is a significant increase in the overall perception of how the Cloud ERP production ability in providing a better, systematic and efficient way in deploying and supporting their products.

VII. CONCLUSION

This research has demonstrated a Cloud ERP Factory model with its five sub models Software Product Line, Platform Model, Workflow Model, Product Control Model and Knowledge Management Model. The model has been validated with expert reviews and simulated with prototyped in the commercial Cloud ERP development environment. The three field demonstration case studies conducted were able to validate and prove that the CEF model can be translated into Cloud ERP production environment which simplifies the complexity of managing Cloud ERP system, high workload, the consistency of software quality and introduces a more efficient method in managing and retaining the knowledge within the operation and technical domain. The model is a significant contribution to Cloud ERP and software factory research domain as it has produced demonstration which can be replicated and expanded into new related researches. It is also highly possible to expand the model into other industry domains solving similar problems. The author acknowledges that future research can also be conducted into each of the sub-components in the CEF Model to further improve the model in its repeatability and archiving the targeted objectives.

REFERENCES

[1] Panorama Consulting Solutions, “2016 Report on Erp Systems and Enterprise Software,” pp. 1–32, 2016.

[2] B. Bond, Y. Genovese, D. Miklovic, B. Zrimsek, and N. Rayner, “4 October 2000 ERP Is Dead — Long Live ERP II,” no. October 2000, 2005.

[3] L. Shaul and D. Tauber, “Critical success factors in enterprise resource planning systems,” *ACM Comput. Surv.*, vol. 45, no. 4, pp. 1–39, 2013.

[4] Y. Dittrich and S. Vaucouleur, “Practices around customization of standard systems,” *Proc. 2008 Int. Work. Coop. Hum. Asp. Softw. Eng. - CHASE ’08*, pp. 37–40, 2008.

[5] E. Uppström and M. Hoffsten, “New Implications for

Customization of ERP Systems," 2015.
[6] Y. Dittrich, S. Vaucouleur, and S. Giff, "ERP customization as software engineering: Knowledge sharing and cooperation," *IEEE Softw.*, vol. 26, no. 6, pp. 41–47, 2009.
[7] N. Pollock, J. Comford, P. Neil, and J. Comfordnclacuk, "Fitting Standard Software to Non-Standard Organisations," pp. 721–725, 2002.
[8] Timm Seitz, "SAP ERP in the Cloud," *Oracle white Pap.*, no. April, pp. 19–37, 2010.
[9] Panorama Consulting, "Key Findings from the 2015 ERP Report," 2015. .
[10] E. F. Kiadehi and S. Mohammadi, "Cloud ERP: Implementation of Enterprise Resource Planning Using Cloud Computing Technology," *J. Basic. Appl. Sci. Res*, vol. 2, no. 11, pp. 11422–11427, 2012.
[11] S. A. Salim, "Cloud ERP adoption-A process view approach," *Proc. - Pacific Asia Conf. Inf. Syst. PACIS 2013*, no. November, p. n. pag., 2013.
[12] P. Bhattacharya, "Dynamic Updates for Web and Cloud Applications," 2009.
[13] M. Armbrust, A. D. Joseph, R. H. Katz, and D. A. Patterson, "Above the Clouds : A Berkeley View of Cloud Computing," *Science (80-.).*, vol. 53, no. UCB/EECS-2009-28, pp. 07–013, 2009.
[14] H. Moens and F. De Turck, "Feature-based application development and management of multi-tenant applications in clouds," *Proc. 18th Int. Softw. Prod. Line Conf. - SPLC '14*, vol. 1, pp. 72–81, 2014.
[15] G. N. Purohit, M. P. Jaiswal, and M. S. Pandey, "Challenges Involved in Implementation of ERP on Demand Solution : Cloud Computing," *Int. J. Comput. Sci. Issues*, vol. 9, no. 4, pp. 481–489, 2012.
[16] R. Ashalatha and J. Agarkhed, "Multi Tenancy Issues in Cloud Computing for SaaS Environment," 2016.
[17] H. Cai, N. Wang, and M. J. Zhou, "A transparent approach of enabling saas multi-tenancy in the cloud," *Proc. - 2010 6th World Congr. Serv. Serv. 2010*, pp. 40–47, 2010.
[18] C. Jepsen, "Does multi-tenancy really matter anymore?," 2015. .
[19] W. Sellami, H. H. Kacem, and A. H. Kacem, "Elastic Multi-tenant Business Process Based Service Pattern in Cloud Computing," *2014 IEEE 6th Int. Conf. Cloud Comput. Technol. Sci.*, pp. 154–161, 2014.
[20] W.-T. Tsai, Q. Shao, Y. Huang, and X. Bai, "Towards a scalable and robust multi-tenancy SaaS," *Proc. Second Asia-Pacific Symp. Internetware - Internetware '10*, pp. 1–15, 2010.
[21] M. Mijač, R. Picek, and Z. Stapić, "Cloud ERP System Customization Challenges," *Cent. Eur. Conf. Inf. Intell. Syst.*, pp. 132–140, 2013.
[22] A. Zaidman, "Multi-Tenant SaaS Applications : Maintenance Dream or Nightmare ? Position paper," *Iwpse-Evol '10*, pp. 88–92, 2010.
[23] R. Mietzner, R. Mietzner, A. Metzger, A. Metzger, F. Leymann, F. Leymann, K. Pohl, and K. Pohl, "Variability Modeling to Support Customization and Deployment of Multi- Tenant-Aware Software as a Service Applications," *Princ. Eng. Serv. Oriented Syst. (PESOS 2009)*, vol. 215483, pp. 18–25, 2009.
[24] H. Bratman and T. Court, "The Software Factory," *Computer (Long. Beach. Calif).*, vol. 8, no. 5, pp. 28–37, 1975.
[25] J. Greenfield and K. Short, "Software factories: assembling applications with patterns, models, frameworks and tools," *Syst. Lang. Appl.*, pp. 16–27, 2003.
[26] N. Y. Sahin, "Cloud ERP Security : Guidelines for Evaluation," pp. 11–38, 2013.
[27] P. Appandairajan, N. Zafar Ali Khan, and M. Madiajagan, "ERP on Cloud: Implementation strategies and challenges," *2012 Int. Conf. Cloud Comput. Technol. Appl. Manag.*, pp. 56–59, Dec. 2012.
[28] N. Castellina, "SaaS and Cloud ERP - Trends, Observations, and Performance," no. December, p. 27, 2011.
[29] S. Corrall, "Cloud ERP: a New Dilemma to Modern Organisations?," *J. Comput. Inf. Syst.*, vol. 64, pp. 49–67, 2010.
[30] M. Uflacker and D. Busse, "Complexity in Enterprise Applications vs. Simplicity in User Experience," *Human-Computer Interact. HCI ppl. Serv. Lect. Notes Comput. Sci.*, pp. 778–787, 2007.
[31] B. S. Arnesen, "Is a Cloud ERP Solution Right for You?," *Strateg. Financ.*, vol. 95, no. 2, pp. 45–50, 2013.
[32] G. Rittik, "A Comprehensive Study on ERP Failures Stressing on Reluctance to Change as a Cause of Failure," vol. 3, no. May, pp. 123–134, 2012.
[33] K. Vaniea and Y. Rashidi, "Tales of Software Updates : The process of updating software," *Chi*, pp. 3215–3226, 2016.
[34] K. E. Vaniea, E. Rader, and R. Wash, "Betrayed by updates," *Secur. Hum. Behav. 2014*, pp. 2671–2674, 2014.
[35] H. Jegadeesan and S. Balasubramaniam, "A method to support variability of enterprise services on the cloud," *CLOUD 2009 - 2009 IEEE Int. Conf. Cloud Comput.*, pp. 117–124, 2009.
[36] C.-F. Liao, K. Chen, and J.-J. Chen, "Modularizing tenant-specific schema customization in SaaS applications," *Proc. 8th Int. Work. Adv. Modul. Tech. - AOAsia'13*, pp. 9–12, 2013.
[37] G. E. Bryan, "Not all programmers are created equal #x2014; Redux #x00A9;," *Aerosp. Conf. 2012 IEEE*, no. March 1980, pp. 1–10, 2012.
[38] D. Kamma, G. G, and P. J Neela, "Countering Parkinson ' s Law for Improving Productivity Categories and Subject Descriptors," 2013.
[39] Abu Bakar, M.S., & Shiratuddin, N., Conceptual Design Model Using Operational Data Store (CoDMODS) for Developing Business Intelligence Applications. International Journal of Computer Science and Network Security. 11(3), 161 - 169.,2011.
[40] C. Fernstrom, K. H. Narfelt, and L. Ohlsson, "Software factory principles, architecture, and experiments," *IEEE Softw.*, vol. 9, no. 2, pp. 36–44, 1992.
[41] D. Helton, "The Pontential of Software Factories," vol. 3, no. 7, pp. 260–266, 2010.
[42] S. Alter, "Work System Theory: overview of core concepts, extensions, and challenges for the future," *J. Assoc. Inf. Syst.*, vol. 14, no. 2, pp. 72–121, 2013.
[43] B. Wernerfelt, "A Resource based view of the firm," *Strateg. Manag. J.*, vol. 5, no. 2, pp. 171–180, 1984.

Dzulkafli Jalil has 25 years' experience in Electronics Design and developing Enterprise Resource Planning (ERP) solution and is currently the CEO at Smart Lab Sdn Bhd Malaysia. He received BSc Electrical Engineering from Purdue University, Indiana, USA in 1993 and MSc IT from USM, Malaysia 2002. He is currently a PhD candidate and researcher at School of Computing, Universiti Utara Malaysia. His research interests are on Cloud ERP Software, Internet of Things (IoT) and Business Intelligence.

Muhamad Shahbani Abu Bakar received the PhD Computer Science, M.Sc (Information Technology) and BSc. Computer Science in 1990 and 1999, respectively. Currently, he is a Deputy Director University Teaching and Learning Center, Universiti Utara Malaysia. After working as an analyst programmer and system analyst (from 1990-2000) in private and government sector and a lecturer (from 2000 - 2009), he has been a senior lecturer in School of Computing, Universiti Utara Malaysia since 2009. His research interest includes Software Engineering, Web Application, Cloud Computing, Enterprise Resource Planning, Big Data, Data Warehouse, Business Intelligence, and Educational Technology.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500