# CLOUD ENTERPRISE RESOURCE PLANNING DEVELOPMENT MODEL BASED ON SOFTWARE FACTORY APPROACH



**DZULKAFLI ABD JALIL**

**DOCTOR OF PHILOSOPHY**
UNIVERSITI UTARA MALAYSIA
**2018**

Awang Had Salleh
Graduate School
of Arts And Sciences

Universiti Utara Malaysia

**PERAKUAN KERJA TESIS / DISERTASI**
*(Certification of thesis / dissertation)*

Kami, yang bertandatangan, memperakukan bahawa
*(We, the undersigned, certify that)*

DZULKAFLI ABD JALIL

calon untuk Ijazah PhD
*(candidate for the degree of)*

telah mengemukakan tesis / disertasi yang bertajuk:
*(has presented his/her thesis / dissertation of the following title):*

**"CLOUD ENTERPRISE RESOURCE PLANNING DEVELOPMENT MODEL BASED ON SOFTWARE FACTORY APPROACH"**

seperti yang tercatat di muka surat tajuk dan kulit tesis / disertasi.
*(as it appears on the title page and front cover of the thesis / dissertation).*

Bahawa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : ***10 Oktober 2017.***
*That the said thesis/dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on:*
***October 10, 2017.***

| | | |
|---|---|---|
| Pengerusi Viva: *(Chairman for VIVA)* | Assoc. Prof. Dr. Fauziah Baharom | Tandatangan *(Signature)* |
| Pemeriksa Luar: *(External Examiner)* | Prof. Dr. Rosni Abdullah | Tandatangan *(Signature)* |
| Pemeriksa Dalam: *(Internal Examiner)* | Assoc. Prof. Dr. Azizah Hj Ahmad | Tandatangan *(Signature)* |
| Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)* | Dr. Muhamad Shahbani Abu Bakar | Tandatangan *(Signature)* |
| Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)* | Assoc. Prof. Dr. Zulkifli Mohamed Udin | Tandatangan *(Signature)* |

Tarikh:
*(Date)* *October 10, 2017*

# Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the University Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUM College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok

# Abstrak

Kajian literatur menunjukkan bahawa Perancangan Sumber Perusahaan Awan (Cloud ERP) telah berkembang dengan pesatnya. Namun, dari perspektif pembangun perisian, ia masih dibelenggu masalah seperti pengurusan yang kompleks, beban kerja yang tinggi, kualiti perisian yang tidak konsisten dan masalah pengekalan ilmu. Kajian terdahulu masih kekurangan penyelesaian yang holistik dalam menangani kesemua komponen masalah dalam kajian ini. Pendekatan Pengilangan Perisian (Software Factory) telah dipilih untuk disesuaikan dengan teori yang berkaitan bagi menghasilkan suatu model yang dirujuk sebagai Cloud ERP Factory Model (Model CEF), yang bertujuan untuk menyelesaikan permasalahan tersebut. Terdapat tiga objektif khusus dalam kajian ini iaitu (i) untuk membangunkan Model CEF dengan mengenalpasti komponen dan elemen terlibat dan mengabungkannya kepada persekitaran cloud, (ii) untuk menentusahkan pengagihan bagi kebolehsanaan teknikal Model CEF, dan (iii) untuk menentusahihkan medan kebolehgunaan penghasilan Model CEF dalam kajian kes sebenar. Kajian ini menggunapakai metodologi Sains Reka Bentuk beserta pendekatan penilaian kaedah campuran (Mixed Methods). Model CEF yang dibangunkan mengandungi lima komponen iaitu Barisan Produk, Pelantar, Aliran Kerja, Kawalan Produk dan Pengurusan Pengetahuan yang boleh digunakan untuk menyediakan persekitaran CEF CEF bagi mensimulasikan persekitaran produksi perisian berorientasikan proses dengan ciri-ciri perancangan sumber dan kapasiti. Model CEF ini telah ditentusahihkan melalui penilaian pakar, dan ditentusahkan kebolehsanaan teknikal nya dengan kejayaan pengagihan model ini kepada komersil terpilih bagi kemudahan produksi Cloud ERP. Tiga kajian kes untuk pengagihan Cloud ERP komersil telah dijalankan menggunakan persekitaran prototaip yang dibangunkan. Dengan menggunakan instrumen tinjauan yang telah dibangunkan, dapatan min skala Likert mencapai 6.3 daripada 7 mata keseluruhan yang menentupastikan model CEF adalah boleh digunapakai dan objektif kajian telah dicapai. Model CEF dan proses pengesahan pengagihan perisian Cloud ERP dalam persekitaran komersil melalui kajian kes sebenar merupakan sumbangan utama kajian ini. Kedua –dua sumbangan ini turut dapat digunakan oleh pengamal industri perisian dan ahli akademik sebagai rujukan untuk membangunkan kemudahan penghasilan Cloud ERP yang lebih mantap.

**Kata Kunci:** Perancangan Sumber Perusahaan Awan (Cloud ERP), Kilang Perisian, Senibina Berorientasikan Perkhidmatan (SOA), Litar Produk Perisian

# Abstract

Literature reviews revealed that Cloud Enterprise Resource Planning (Cloud ERP) is significantly growing, yet from software developers' perspective, it has succumbed to high management complexity, high workload, inconsistency software quality, and knowledge retention problems. Previous researches lack a solution that holistically addresses all the research problem components. Software factory approach was chosen to be adapted along with relevant theories to develop a model referred to as Cloud ERP Factory Model (CEF Model), which intends to pave the way in solving the above-mentioned problems. There are three specific objectives, those are (i) to develop the model by identifying the components with its elements and compile them into the CEF Model, (ii) to verify the model's deployment technical feasibility, and (iii) to validate the model field usability in a real Cloud ERP production case studies. The research employed Design Science methodology, with a mixed method evaluation approach. The developed CEF Model consists of five components; those are Product Lines, Platform, Workflow, Product Control, and Knowledge Management, which can be used to setup a CEF environment that simulates a process-oriented software production environment with capacity and resource planning features. The model was validated through expert reviews and the finalized model was verified to be technically feasible by a successful deployment into a selected commercial Cloud ERP production facility. Three Cloud ERP commercial deployment case studies were conducted using the prototype environment. Using the survey instruments developed, the results yielded a Likert score mean of 6.3 out of 7 thus reaffirming that the model is usable and the research has met its objective in addressing the problem components. The models along with its deployment verification processes are the main research contributions. Both items can also be used by software industry practitioners and academician as references in developing a robust Cloud ERP production facility.



**Keywords:** Cloud Enterprise Resource Planning (Cloud ERP), Software Factory, Service Oriented Architecture (SOA), Software Product Line

# Acknowledgement

In the Name of Allah, the Most Gracious the Most Merciful

Alhamdulillah, I have finally managed to complete this research and if it wasn't with the help of those supporting me, it might not be possible to come to this stage. My sincere gratitude and appreciation goes to my primary supervisor, Dr. Muhamad Shahbani Abu Bakar and my secondary supervisor, Assoc. Prof. Dr. Zulkifli Mohamed Udin, who have put tremendous effort and guidance in making this research a success. Thank you again for believing in me.

I would like to thank the Ministry of Higher Education (MOE) for my PhD study sponsorship and also for the Knowledge Transfer Program Grant between Smart Lab Sdn Bhd and Universiti Utara Malaysia. This research has involved a lot of industry experts, academicians, research assistants, and respondents who have passionately participated. To them, I will forever owe a gratitude, which hardly can be repaid.

To family members; my parents Abd Jalil Abdullah and Siti Hasnah Johar, my wife Adelina and daughter Delilah, my siblings, relatives, and friends. I wish for this research to make you proud and happy. Cheers!

# Table of Contents

# List of Tables

# List of Figures

# List of Abbreviations

| | |
|---|---|
| CEF | Cloud ERP Factory |
| IaaS | Infrastructure as a Service |
| PaaS | Platform as a Service |
| SaaS | Software as a Service |
| CQI | Continuous Quality Improvement |
| ERP | Enterprise Resource Planning |
| SPLE | Software Product Line Engineering |
| MDA | Model Driven Approach |
| SOA | Service Oriented Architecture |
| CRM | Customer Relationship Management |
| SCM | Supply Chain Management |
| UML | Unified Modeling Language |
| SF | Software Factory |

# List of Publications

The following is a list of the publications related to this research that have been published in journals and other proceedings.

**JOURNALS**

1. Jalil, Dzulkafli, Abu Bakar, Shahbani (2017). Adapting Software Factory Approach into Cloud ERP Production Model. *International Journal of Computer Science and Information Security*, *15*(1)
2. Jalil, Dzulkafli, Abu Bakar, Shahbani (2017). Enabling Software Factory with Job Workflow. *International Journal of Computer Science and Information Security*, *15*(4)
3. Muhamad Shahbani Abu Bakar, Dzulkafli Jalil, Zulkifli Mohamed Udin (2017). Knowledge Repository: Implementing Learning Management System into Corporate Environment *International Conference on Applied Science and Technology*, *8*(8) JTEC
4. Jalil, Dzulkafli, Shahbani Abu Bakar, Mohd Khir, Mokhzani Fauzi (2017). Integrated Facility Platform for Next-Gen Aircraft Maintenance, Repair and Overhaul (MRO). *International Journal of Computer Science and Information Security*, *15*(5)

**PROCEEDINGS**

1. Muhamad Shahbani Abu Bakar, Dzulkafli Abd. Jalil, Azman Ta'a1, Zulkifli Md. Udin, Said Ashari (2017). Adapting Learning Management System into Corporate Environment: Knowledge Repository in Cloud Enterprise Resource Planning Software Provider . *3rd National Conference on Knowledge Transfer*, *17*(1)
2. Muhamad Shahbani Abu Bakar, Dzulkafli Jalil (2017). Corporate Knowledge Repository: Adopting Academic LMS into Corporate Environment. *International Conference on Applied Science and Technology, 17(4)*

# Awards and Recognitions

1. Grant Knowledge Transfer Program, Implementation of Cloud Enterprise Resource Planning. Ref No: I-ECO/44(UUM-15)/Code SO:13211/Project Cost: RM 143, 086.00.

2. SILVER Medal, IUCEL 2017, International University Carnival, Cloud Learning Management System (LMS) in Corporate Environment. 26-27 Sept 2017, University Sains Islam Malaysia.

# CHAPTER ONE
# BACKGROUND OVERVIEW

## 1.1 Overview

The first chapter offers a brief overview of the research, which focuses on the data that lead to the motivational aspect of the research, specifications of the problem, extraction of research gaps with research questions, and the formulation of research objectives. The scope and limitations of the research and its expected contributive elements will be clearly defined. The research theoretical framework diagram will describe the theoretical approach of the research. Finally, operational terminologies and the overall thesis structure utilized will be explained.

## 1.2 Background Study

Enterprise Resource Planning (ERP) is undoubtedly a critical component of any business operation, thus making it almost a mandatory requirement to set up a business (Panorama Consulting Solutions, 2016). ERP refers to an enterprise business strategy and a set of industry-domain-specific applications that promote customer and shareholder value by enabling and optimizing collaborative operational and financial processes (Bond, Genovese, Miklovic, Zrimsek, & Rayner, 2005). The term ERP was initially defined by Gartner in 1990 and was later revised to the term ERP II in 2000, which expanded the scope to include almost every business facet within an organization. Figure 1.1 illustrates a typical commercial ERP package.


*Figure 1.1* Typical Commercial ERP package.

In the context of this research, the term ERP refers to all systems within an organization, such as the Financial Management System (FIS), Human Resources Management (HRMS), Manufacturing Resource Planning (MRP), as well as systems with external interaction to other systems; such systems are CRM and SCM (Shaul & Tauber, 2013).

In an attempt to reduce cost, ERP software providers have been trying to standardize their ERP packages into standard common modules with configuration capabilities (Dittrich & Vaucouleur, 2008; Uppström et al., 2015). Over the years, with the knowledge gained from previous implementations, ERP vendors have been trying to come up with  a configurable standard version of ERP software, which has potential to be delivered to future clients (Dittrich, Vaucouleur, & Giff, 2009). By using configuration features, it is hoped that customer requirements can instead be solved with configuration options, rather than with software code customization.

Theoretically, a future client 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 between the standard ERP software package and that of any new customer requirements (Pollock, Comford, Neil, & Comfordnclacuk, 2002).

This after all, is the main struggle of ERP companies that is to keep updating the new software version 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 a century of updated software revision, customization during the new implementation is inevitable (Timm Seitz, 2010). So, despite the highly configurable ERP packages, code based customization remains a major requirement for most ERP implementations (Panorama Consulting, 2015). Besides, ERP customization also reflects that every business has its own unique business model, which contributes to the organization's competitive advantage (Pollock et al., 2002).

The emergence of the Internet in previous decades has tremendously affected the ERP solution technology and its deployment methods (Kiadehi & Mohammadi, 2012). The gaining momentum of Internet Cloud technology provides a natural progression of ERP applications into a Cloud model (Salim, 2013). Traditional mainframe and client server computing model have been made obsolete with the new web-based technology (Bhattacharya, 2009). ERP adaptation into a cloud computing model has shaped the cloud model into a few models, aptly named private cloud, public cloud, and hybrid cloud (Armbrust, Joseph, Katz, & Patterson, 2009). When the solution is accessible only within a privately managed network such as on-premise solution, it is called private cloud. Public cloud solution is the delivery of an ERP package over a public cloud, such as Amazon, Google, and Apple Cloud, etc. Hybrid cloud model is the solution that allows bridging ERP over private and public clouds (Moens & De Turck, 2014).

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 a cloud based solution, advantages such as 'on demand capability', scalability and maintainability, and lower Total Cost of Ownership (TCO) for customers. The Cloud's SaaS technology has turned ERP into a system that is expected to be on demand in nature (Purohit, Jaiswal, & Pandey, 2012). In order to leverage the Cloud economic of scale potential, a Cloud ERP system should be in multi-tenancy mode (Ashalatha & Agarkhed, 2016; Cai, Wang, & Zhou, 2010; Jepsen, 2015; Sellami, Kacem, & Kacem, 2014; W.-T. Tsai, Shao, Huang, & Bai, 2010). 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 (Mijač, Picek, & Stapić, 2013; W.-T. Tsai et al., 2010). 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 dimension (Zaidman, 2010).

Another approach is the multi-tenancy model with composite add-on variants (Mietzner, Leymann, & Papazoglou, 2008). Figure 1.2 illustrates how Cloud ERP single tenancy differs from that of the Multi Tenancy model.

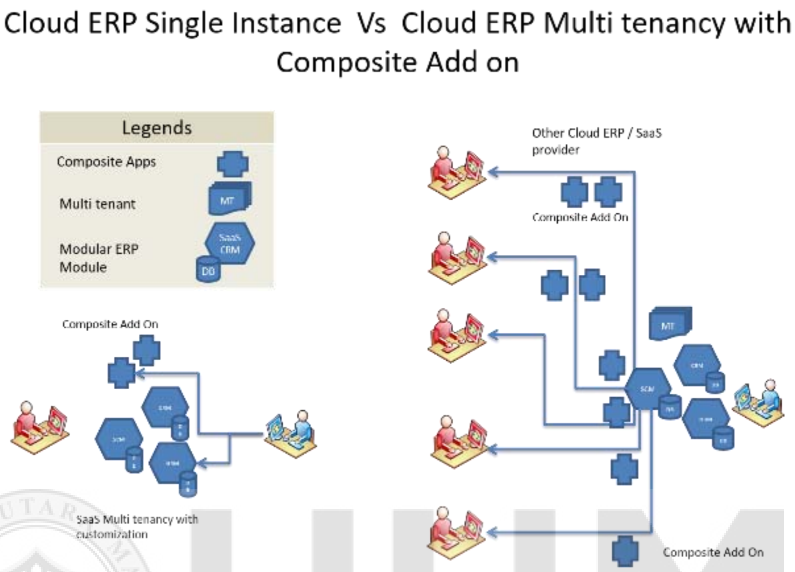



*Figure 1.2* Cloud ERP Single Instance VS Cloud ERP Multi-tenancy with Composite Add-ons



One possible approach to simplify and reduce the complexity is to break down the software development production into a process based approach, just like in standard product manufacturing or a factory. The closest concept of this nature in the software industry is the Software Factory approach. The Software Factory term was coined in 1975 by (Bratman & Court, 1975). In their terminology, it is an integrated set of tools that supports the concepts of structured programming, program development, program production libraries, and incorporates hierarchically structured program modules as the basic unit of production (Bratman & Court, 1975). Today, Software Factory is a part of Software Product Line in the field of Software Engineering, which refers to an approach that configures extensive tools, processes, and content, while 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 (Greenfield & Short, 2003). In simpler terms, software factory is an approach of software development life cycle to mimic factory manufacturing processes.

**1.3 Research Motivation**

In a press release made in 2016, Gartner predicted that in 2018, Postmodern ERP with a smaller on-premise ERP core and a multi-vendor SaaS solution will start replacing the legacy ERP model. Postmodern ERP refers to a Hybrid approach of ERP with customizable on premise or Private Cloud Solutions integrated with loosely coupled SaaS modules from multiple providers over the cloud (Gartner, 2016).

Despite the growing awareness and the acceptance of ERP, history has shown that the implementation success rate of ERP has been a major setback. The failure rate of the implementation is significantly high. According to KPMG Canada Survey in 1997, over 61% of the projects investigated were perceived to have failed (Whittaker, 1999). In 2001, Robbins-Gioia Survey found that 51% of the participants considered their ERP implementation as unsuccessful (Ghosh, 2012). Even in this recent year, research done by Panorama Consulting shows that the ERP failure rate is as high as 72%, and has in fact increased from previous years (Panorama Consulting, 2015). Among others, incompatibility of business processes or resistance to change are reported as two of the major contributing factors (Ghosh, 2012).

Nevertheless, the need of implementation of an ERP system has never been more critical for most organizations. More and more organizations require an ERP system setup even before the start of operations (Xu, 2011). By today, almost all financial institutions, both large and small medium enterprises are relying on ERP systems. The globalization trend and its effect has increased the need for Small Medium Enterprises (SME ) to consider an ERP adoption in order for them to remain in business (Park & Lee, 2006). The undeniably strong growth of cloud computing is directly related to cloud ERP growth. Since Cloud ERP itself is the major product of Cloud computing, the ERP market is also one of the fastest growing and most lucrative businesses within the software industry (Xu, 2011). However, the adaptation of cloud ERP is also still relatively low (12%), as shown in Figure 1.3 reported by Panorama Consulting. This fact implies that research in this area can be further explored. In summary, the problem regarding the high ERP failure rate, critically high demand of Cloud ERP, and a relatively low adoption rate of Cloud ERP serve as motivational factors for further research exploration within the domain of Cloud ERP.



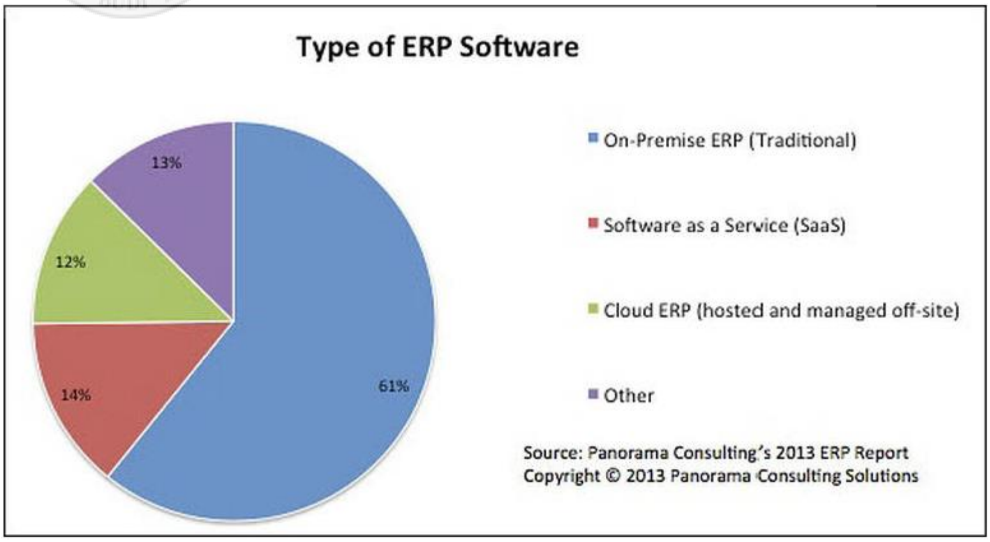



*Figure 1.3* Cloud ERP market share

Another motivation factor is the fact that software factory demonstration has been lacking especially in the Cloud ERP production. Despite the long history of software factory, Helton (2010) cited that the demonstration of software factory as a whole has been very inadequate, due to its nature of complexity and an inadequacy of required tools. From this research view, this statement creates a very good research opportunity.

In addition to the issues presented, the ability to retain the knowledge in knowledge based industries is another factor that contributes to the research motivation. Gulati & Srivastava (2014) reiterates this widely known fact that being overly dependent on employees especially in the IT services companies would pose a brain drain when an employee eventually leaves the company, bringing along the knowledge.



## 1.4 Problem Statement

Cloud ERP has promised a more affordable solution of acquiring an ERP (Sahin, 2013). It delivers a simpler way of implementing the system without elevated expenditures on infrastructure (Appandairajan, Zafar Ali Khan, & Madiajagan, 2012; Castellina, 2011). The on-demand feature of the Cloud services also hinted at a more scalable model for customers than ever before. (Corrall, 2010). At least for Enterprise SaaS application, customers can now order online a ready system to be deployed in a shorter time compared to that of the traditional ERP (Appandairajan et al., 2012). The simplification of acquisition from customers' side does not come free of charge. The ERP developers or ERP vendors are now facing a more complicated system to manage (Uflacker & Busse, 2007). 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

(Castellina, 2011; Corrall, 2010). The nightmare of Cloud ERP management from vendor's view is further worsened by the fact that most customers will not simply settle with the supposedly proven configurable system (Appandairajan et al., 2012; Arnesen, 2013; Timm Seitz, 2010). They want to add more functionality to the system or in a way customize it. In fact, they want to do it regularly, not just one time. Even with a system, which allows for a high level of configuration features, customers' business processes are dynamic and ever changing; thus, customization and updates are still required (Rittik & Ghosh, 2012). This in turn poses a big challenge to ERP providers in developing and maintaining the software.

The challenge is then obviously greater, considering that most Cloud software is expected to be multi-tenancy with single code version model for it to achieve economy of scale benefit (Cai et al., 2010; Sellami et al., 2014; W.-T. Tsai et al., 2010). Therefore, Cloud ERP with multi-tenancy customization model carries a more serious challenge for researchers and ERP practitioners alike (Mijač et al., 2013).

The complexity problem in the back-end Cloud ERP management is also related to technology and industrial business process advancement, which require frequent cloud software version updates (K. E. Vaniea, Rader, & Wash, 2014; K. Vaniea & Rashidi, 2016).

The on-demand nature of Cloud ERP enables customers to subscribe to an ERP solution by module. Consumers can choose to add another module to integrate with their existing module(s). 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 instances. The modules need to be loosely coupled and yet be integrable when being deployed. This is another factor that causes complexity for the Cloud ERP back end management (Jegadeesan & Balasubramaniam, 2009; Liao, Chen, & Chen, 2013).

As many software programmers and analyst are involved is developing ERP software, the concern about software quality consistency is real (Bryan, 2012). Without a proper measuring structure, there will be different software quality standards among the modules as programming and knowledge skills vary between programmers. Experienced programmers will tend to produce better software quality in term of functional and structural specification than that of junior programmers (Kamma, G, & J Neela, 2013).

Since Cloud ERP, as a product itself is about providing a business solution to enterprises, processed knowledge of a business will be part of the company's assets, along with computing and programming knowledge. Technically, it reflects a vast knowledge pool that resides within the business operation that introduces a knowledge management challenge. As this is a typical knowledge-based company, another problem is without a systematic management model, this knowledge is coupled to the workers. Over dependency on knowledge-workers could be a serious problem, as inevitably an employee will leave the organization, thus introducing knowledge gap in the Cloud ERP production. While one of the challenges is how to manage vast amounts of knowledge, retaining that knowledge against staff turnover is another challenging task (Ghahfarokhi & Zakaria, 2009).

In summary, from the ERP software providers' perspective, Multi Tenancy features, dynamic customization requests, frequent software version updates, and upholding modularity features of Cloud ERP have contributed to the high complexity and heavy workload of Cloud ERP production processes. In addition, craftsmanship effects due to varying staff skills creates functional and structural software quality problems 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 Cloud ERP providers. From this point forward, the current research will refer the problems mentioned above, which are complexity, inconsistent software quality, heavy workload, and knowledge retention management as research problem components.

## 1.5 Research Gaps and Research Questions

In solving the research problems mentioned above, the previous research that attempted to solve the specific challenges mentioned have been gathered and grouped.

Table 1.1

*Grouping of Relevant Research on Cloud ERP Challenges*

| No | Problem Addressed | Research Authors | Limitations/Relevancy |
|---|---|---|---|
| 1 | Multi-Tenancies Management | Chang-Hao Tsai, Yaoping Ruan, Sambit Sahu, Anees Shaikh, Kang Shin, Alexander Clemm, Lisandro Granville, Rolf Stadler, 2007<br>Hong Cai, Ning Wang, Ming Jun Zhou, 2010<br>R Ashalatha, Jayashree Agarkhed, 2016<br>Sanjukta Pal, Amit Kr Mandal, Anirban Sarkar, 2015<br>Wael Sellami, Hatem Hadj Kacem, Ahmed Hadj Kacem, 2014 | • Focus only on Multi Tenancy architecture as the most suitable method<br>• Believes that Multi-Tenancy able to achieve economies of scale<br>• Migrating Single Tenant to Multi Tenancy |
| 2 | Customization – Dynamic requests | Zhu, Xiyong & Wang, Shixiong, 2009<br>Borovskiy, Vadym, Zeier, Alexander, Koch, Wolfgang, Plattner, Hasso 2009 | • Tries to enable customization and configuration<br>• Proposes several customization |

Table 1.1 continued

| | | | |
|---|---|---|---|
| | | Khadija Aouzal, Hatim Hafiddi, Mohamed Dahchour, 2015<br>Elin Uppstrom, Carl Mikael Lonn, Madeleine Hoffsten, Joakim Thorstrom, 2015<br>Marko Mijač, Ruben Picek, Zlatko Stapić, 2013 | framework.<br>• Focus only on allowing customization in multi-tenancy |
| 3 | Software Updates | Giuffrida, Cristiano & Tanenbaum, Andrew S., 2009<br>Kami Vaniea, Yasmeen Rashidi, 2016<br>Kami E. Vaniea, Emilee Rader, Rick Wash, 2014<br>R. Jhanwar, T. Yaryan, 2007 | • Software update is critical<br>• Prefers to live update instead of stopping system to update any patches, fix bus<br>• Uses SaaS as one entity for most proposed solution |
| 4 | Modular Cloud ERP | Ali, Nasr, Gheith, 2016<br>Schackmann, Holger & Lichter, Horst, 2006<br>Kumara, Indika, Han, Jun, Colman, Alan Nguyen, Tuan & Kapuruge, Malinda, 2013 | • Aims for modularity using Software Product Line approach |
| 5 | Resource Provisioning | Atul Gohad, Karthikeyan Ponnalagu, Nanjangud G. Narenda, 2012<br>Shi, Jiyuan Dong, Fang, Zhang, Jinghui, Jin, Jiahui & Luo, Junzhou, 2016<br>Truong, Hong-linh, 2016 | • Addressing mostly infrastructure as a service |
| 6 | Security Measures | Anwar, Mohd & Imran, Ashiq, 2015<br>Kiadehi, Elias Fathi & Mohammadi, Shahriar, 2012<br>Maheshwari, Shivang & Sharma, Charu, 2015<br>Alouane, Meryeme & Bakkali, Hanan E L, 2015 | • Addressing security concerns and multi-tenancy isolation<br>• Disaster Recovery, redundancy |
| 7 | Knowledge Management | Gulati, V. P., Srivastava, Shilpa, 2014;<br>Sridharan, Bhavani, Kinshuk, 2003;<br>Schilling, A, Laumer, S<br>Weitzel, T, 2012 | • Address the knowledge management and retention as employee leaves the organization |
| 8 | Inconsistent Software Quality | Bryan, G E, 1980;<br>Boehm, B W 1976;<br>Sison, Raymund 2009 | • Explains different programmers have different skills |

Table 1.1 lists groupings of research that are related to the present research query on Cloud ERP challenges; those are Multi Tenancy, customization, software updates, software modularity, resource provisioning, security measures, knowledge management, and inconsistent software quality. These are important features of Cloud ERP that make it vary from the traditional ways of the software or ERP industries. Obviously, Cloud ERP management is not just a software development lifecycle, but rather it is a software development solution that sits in an outsourcing business lifecycle. Although it is similar to SaaS solution, ERP as a service typically

offers multiple SaaS solutions that are well integrated when it is delivered. Figure 1.4 shows the difference between ERP as a Service when compared to SaaS. Technically, most of the previous research mentioned above did not address ERP as a modular SaaS solution. In reality as well as in this research, ERP can be acknowledged as a modular SaaS solution; thus, making it much more complex and difficult to manage.

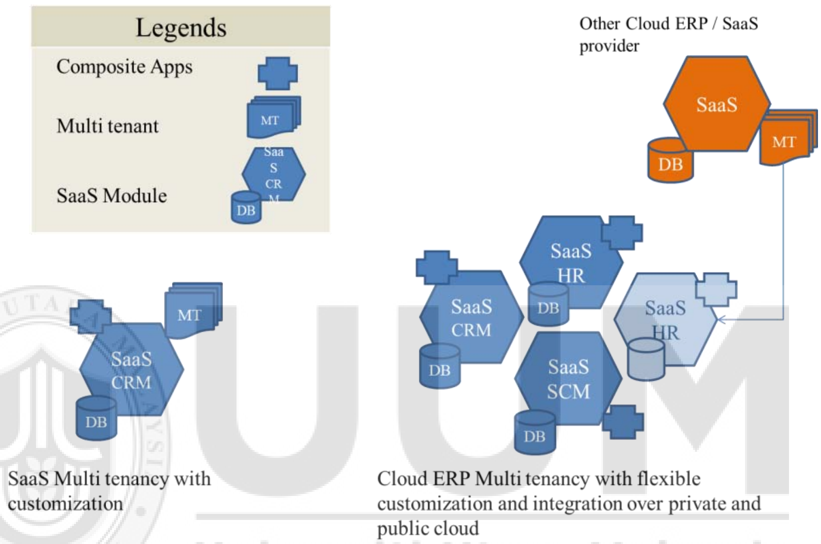



*Figure 1.4* Cloud ERP with customization as compared to SaaS solution.

While all of the research mentioned above did address its objectives, they did not address the overall view from the back-end management point of view. What is needed now is a model that provides a systematic management reference, which can indirectly provide a solution to the list of problems above. One of the software engineering approach in solving large enterprise system development is called Software Factory. It is an attempt to industrialized software development by mimicking manufacturing model.

Previous research regarding the Software Factory approach have proven successful in solving similar problems in various contexts. Table 1.2 lists down 18 research that has incorporated Software factory in their solution in attempting to solve various software production problems. Group one researchers defined and demonstrated Software Factory in general a view, Group two in an educational environment, Group three in solving problems from within the company, while Group four in solving the problems in a software production company.

Table 1.2:
*Previous Software Factory approach solution grouped by their scopes and objectives*

| Group | Authors | Grouping | Limitation/Relevancy |
|---|---|---|---|
| 1 | (Bratman & Court, 1975; Canuel & Robert, 2012; Fernstrom, Narfelt, & Ohlsson, 1992; Helton, 2010; Nomura, Tonini, Hikage, & Tonini, 2007) | General Purpose | • Provides definition of Software Factory<br>• Addressing Software Development methodologies<br>• Did not cover Cloud ERP environment |
| 2 | (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<br>• 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<br>• Management Information System view<br>• Internal development model<br>• Did not cover Cloud ERP environment |
| 4 | (Greenfield & Short, 2003; Li, Li, & Li, 2001; Lim, Ang, & Parvi, 2000; Rockwell & Gera, 1993) | Software Vendor | • Adaptation of Software Factory in software production for software vendors<br>• Mass customization approach<br>• Cover outsourcing model<br>• Did not cover Cloud ERP environment |



Although all the researchers listed in Table 1.2 may have variations in their Software Factory concepts and models, their basic principles are similar in that they are mainly targeting a way to provide a systematic approach of developing software by reducing human craftsmanship effects as well as promoting reusability. In the table, the research has been categorized into four groups. Group 1 authors provide an insight and definition of Software Factory. Group 2 incorporated Software Factory in learning educational environment. Group 3 were focused on Software Factory in corporate environment, and thus, was categorized with Management Information System as its focus. Researchers from Group 4 provide the most relevant Software Factory insight from the software developers' view. However, their solutions are intended for a traditional software production model not a Cloud SaaS model or an ERP model. Moreover, most of the research did not provide a production model that can be simulated to produce Software Production environment. In addition, a specific Software Factory approach focusing on Cloud ERP production is also lacking.

In summary, although the researchers listed in Table 1.1 have provided various solutions in solving some of our problem statement components, their approaches rarely provided a solution model from holistic development model concept and obviously not from that of software factory approach. While researchers in Table 1.2 focused on Software Factory, their research did not directly address our problem statement components. Researchers from both Table 1.1 and Table 1.2 also did not focus on Cloud ERP production model, and generally did not provide a production model that can be simulated to any Cloud ERP production environment in solving development workload, complexity, consistence quality, and knowledge retention management problems.

Therefore, to bridge the research gap, the researcher seeks to find a solution to the problem statement in a Cloud ERP software production environment model, which can be used to solve the problem stated earlier. In formulating the model, which can now be referred to as Cloud ERP Factory model, the following research questions were triggered to facilitate the research.

i. What are the components effected and how can the Cloud ERP Factory model be constructed?
ii. How to verify that the Cloud ERP Factory model is technically feasible to be deployed into a Cloud ERP production environment?
iii. Is the proposed model usable in solving the research problem components?

**1.6 Research Objectives**

The main objective of the research is to propose a Cloud ERP Factory (CEF) Model from an ERP solution providers' view that can tackle the complexity of software production, minimize the workload, improve software quality consistency, and provide knowledge retention management capabilities by using Software Factory approach. In order to develop the main objective, the following specific objectives are as listed below:



i. To develop the model by identifying the components with its elements and compile them into the CEF Model.
ii. To verify the model's technical feasibility by deploying it into an existing Cloud ERP Production environment.
iii. To evaluate the model field usability by using a prototype environment in a real Cloud ERP production case studies.

## 1.7 Research Scope and Limitation

This study is proposing a Cloud ERP model from a developer's perspective. The research scope was to construct a model that can be used as a reference model for any commercial ERP providers and corporate users. However, the scope of this research was limited by the following constraints:

i. This study will only focus on the architectural, workflow, and product management aspect of the models. Cloud ERP Resource provisioning and security aspects were not focused on, as they are mostly covered in the mainstream research.

ii. This study focuses on major technical components in ERP development and excludes the other factors such as social, financial, and political influence.

iii. The model will be developed comprehensively in order to tackle the solution to the problem statements. However, during the model simulation or prototyping in the commercial Cloud ERP production environment, some of the model component element aspects such capacity planning capability may not be fully implemented as it required more data and time before it could be effectively enabled.

iv. The research scope is to develop the model that can tackle the research problem components, but not the correlation or relationship among the research problem component variables.

v. Model validation will employ field usability case study and will only focus on evaluating the model using Perceived Usefulness and Ease of Use (PUEU) questionnaires, and its ability to address the research problem components of the prototype environment based on the proposed model.

vi. The intended audience for this study should be familiar with ERP software development processes especially Cloud environment.

## 1.8 Significance of the Study

This study is significantly important, as it is a novel attempt to develop a Model of the Cloud ERP Factory model that can help in reducing the inherent problem of Cloud ERP backend management. Continuously improving business processes is important for any organization, yet most of the existing Cloud ERP models are driving toward minimizing Cloud ERP customization with SaaS Multi-Tenancy Model. Instead of trying to trade off the customization to achieve SaaS economy of scale, this model provides an alternative approach that can enable continuous ERP customization without sacrificing business scalability. CEF promises to provide a new paradigm shift regarding the views concerning Cloud ERP development. The model can serve as a reference guideline to industrial practitioners to implement a Cloud ERP production model with a higher rate of success. Using CEF model in developing Cloud ERP may directly mean higher productivity with fewer resources, thus indirectly may lower the cost of acquiring a Cloud ERP solution. This study is also meant to become the base research foundation, which would promote more intensive research pertaining to the Cloud ERP Factory (CEF) model, the Software Factory model, as well as dynamic software product line as discussed within this research.



## 1.9 Research Theoretical Framework

This research was carried out based on concepts and theories related to developing, managing, and maintaining software development. The solution was then expressed in the Model using the Theoretical Model shown in Figure 1.5 and five phases shown in Chapter Three. The research and theoretical framework is visualized in Figure 1.5. Facts and figures from a literature review elicitation were conducted to identify the research problems, scope, questions, and objectives. Literature studies, expert survey, content analysis regarding relative concepts and theories were also performed in order to determine the components of the CEF model affecting Cloud ERP lifecycle management. Work system and Resource Based Theories will be integrated with Software Factory approach to come out with a Theoretical Model of developing the Model. In addition, comparative analysis of existing models of relevant Cloud ERP models have been carried out to support the research gap and identify critical Cloud ERP components.

With the information gathered and Theoretical Model guides, the intended model has been developed based on the outcome of Phase One. All the component variables were then expressed using proven concepts such as Abstraction Layers, Feature Model, Workflow diagrams, System Context Diagrams, and other appropriate forms that could be utilized to enhance the model's readability. The model was then deployed to create a prototype environment of the real CEF development, implementation, and support in order to validate its usability and the model's ability to address the research problem components. These were iterative processes in informal experimental form, which later were evaluated and confirmed by expert review. The prototyped environment was then systematically tested for its usability and ability to tackle the research problem components in three real case studies of actual Cloud ERP implementations. Finally, in the conclusion stage, the results were analyzed against the research objectives and the predicted outcome of the current exploration.

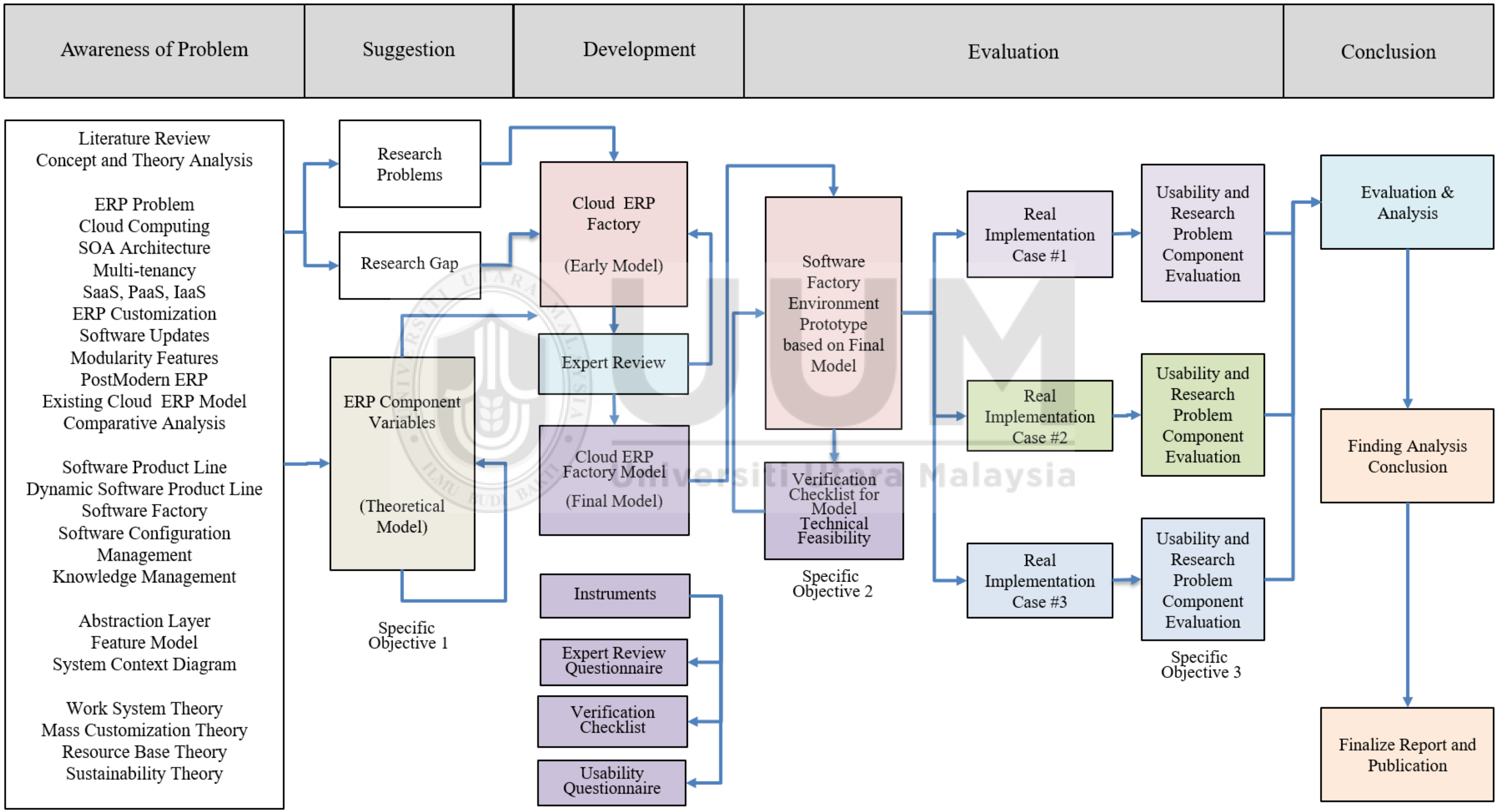



*Figure 1.5* Research and Theoretical Framework

## 1.10 Operational Definition of Terminologies

**ERP**

An ERP system transforms an organization’s entire business process into integrated software applications embracing all modules such as finance, accounting, human resource, and manufacturing, etc.

**Cloud Computing**

Cloud computing is the delivery of computing services to the end users across the internet using a service oriented architecture model.

**SOA**

Service Oriented Architecture refers to an architectural pattern in computer software design in which application components provide services to other components via a communications protocol, typically over a network. The principles of service-orientation are independent of any vendor, product, or technology.



**SaaS**

Software as a Service is terminology used to describe a type software delivery solution, which uses an internet cloud platform as its deployment base. It is also sometimes referred to as “On demand Software “solution.

**PaaS**

Platform as a Service refers to software platform underlying the software service that can be delivered to customers by an on-demand service.

**IaaS**

Infrastructure as a Service refers to the hardware or software infrastructure underlying the software platform that can be delivered to customers as an on-demand service.

**MDA**

Model Driven Architecture (MDA) refers to a software design approach for the development of software systems. It provides a set of guidelines for the structuring of specifications, which are expressed as models.

**Postmodern ERP**

Postmodern ERP is a technology strategy that automates and links administrative and operational business capabilities (such as finance, HR, purchasing, manufacturing, and distribution) with appropriate levels of integration that balance the benefits of vendor-delivered integration against that of business flexibility and agility. This definition highlights that there are two categories of ERP strategy: administrative and operational.

**CRM**

Customer relationship management (CRM) is an approach to managing a company's interaction with its customers and prospects. The CRM approach tries to analyse data about customers' history with a company in order to better improve business relationships with customers, specifically focusing on retaining customers, in order to drive sales growth.

**SRM**

Supplier relationship management (SRM) is the discipline of strategically planning for, and managing, all interactions with third party organizations that supply goods and/or services to an organization in order to maximize the value of those interactions.

**SCM**

Supply Chain management is the management of the flow of goods and services. It includes the movement and storage of raw materials, work-in-process inventory, and finished goods from point of origin to point of consumption.

**SPLE**

Software Product Line Engineering is a field of study that refers to software engineering methods, tools, and techniques for creating a collection of similar software systems from a shared set of software assets using a common means of production.

**Software Factory**

A Software development model approach that mimics product-manufacturing concepts in order to industrialize the software industry.

**Mass Customization**

A theory of producing personalized or customized products of services at near mass production efficiency.

**CEF**

Cloud ERP factory. The main artifact of this research, which is a model base on Software Factory that is used to solve inherent Cloud ERP production problems.

## 1.11 Thesis Structure

The following chapters contents are listed below.

**Chapter Two: Literature Review**

This Chapter presents the overall literature review done prior to the research. It explores the existing research related to Cloud ERP and the theories and concepts that lead to the need for this research.

**Chapter Three: Research Methodology**

Chapter Three provides an overview of the method of the research and processes involved in this study from the beginning to the end of the research in order to achieve the research objectives.

**Chapter Four: Model Development**

Chapter Four focuses on the model development phase for the CEF Model. This includes guidelines, components, specific activities, and deliverables. It explains the overall development and validation process of the CEF Model. The main focus in mind is to provide a simplified blueprint for Cloud ERP providers on how to run a standardized Cloud ERP production by following the steps explained in the CEF Model.

**Chapter Five: Model Verification**

Chapter Five focuses on the activities to provide a technical feasibility of the CEF Model application finalized in Chapter Four by developing a CEF environment. The development of a CEF environment is designed to capture the essence of the CEF Model including its subcomponents. The objective of this phase is to verify the finalized model by adopting it into the existing commercial Cloud ERP production environment.

**Chapter Six: Model Validation**

Chapter Six elaborates on how the CEF Model was validated using real-world field usability case studies. The developed prototype CEF Model environment is used as a validation vehicle in order to conduct the field usability validation. case studies have been conducted to focus on the usability of the CEF Model implemented in the selected Cloud ERP company. The results for the three field usability case studies were compiled and analyzed to form a research conclusion regarding the ability of the model to tackle the research problem components.

**Chapter Seven: Conclusion**

Chapter Seven summarizes all the research activities that has been carried out and further review the outcome based on the objectives and the research questions. The Research Findings and Discussion section revisits the three research questions and the research objectives with indepth reviews. The chapter then elaborates the research contribution artifacts and discuss the research limitation and recommendation before finally concluding the research with a closing summary.

# CHAPTER TWO
# LITERATURE REVIEW

## 2.1 Introduction

This chapter attempts to provide a structured review of ERP, Cloud ERP, Software Factory, and its related technological publications. The objective is to look for concepts, theories, methods, and approaches relevant to the current research documentation needs. The reviewed literature has been analyzed to formulate the motivation for the research justification. Figure 2.1 describes the overall structure of the literature review that leads to the formation of the problem statements and research gap.

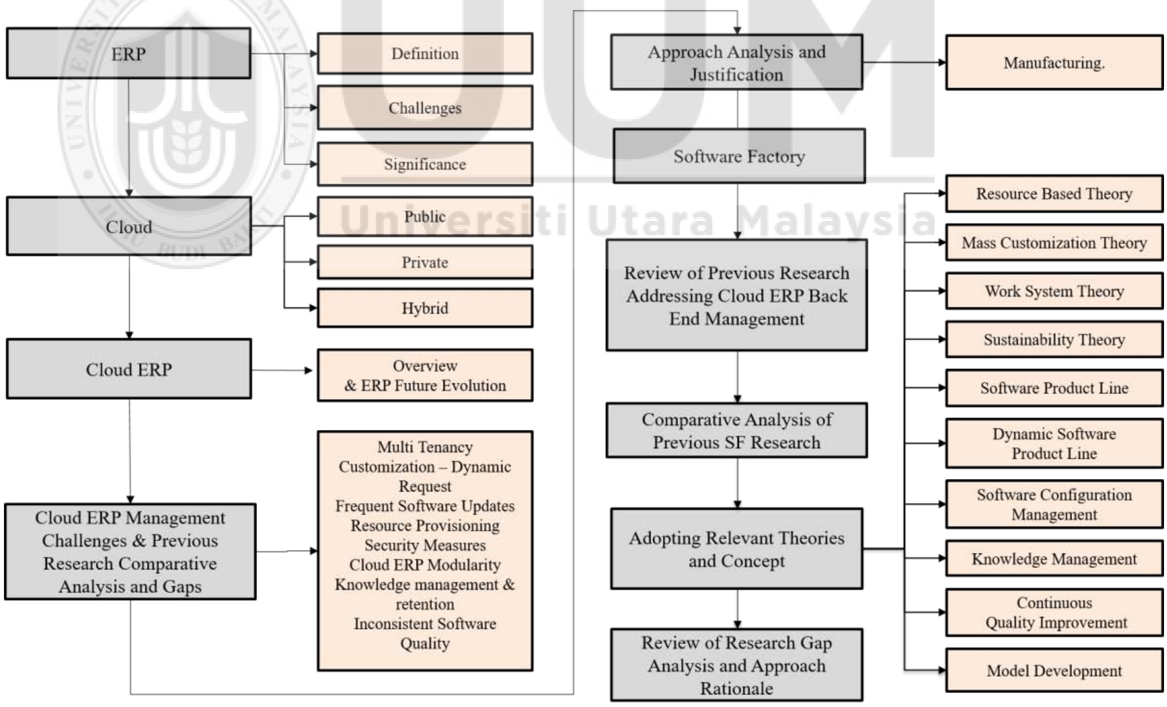



*Figure 2.1* Literature Review Structure

## 2.2 ERP Overview

ERP has an extensive history starting from 1990, since Gartner Group created the acronym to describe it as an extension of Manufacturing Resource Planning (MRP). At the time, ERP was sometimes described as a back-office system along with other terminologies, such as Customer Relationship Management (CRM) and Supplier Chain Management (SCM); consequently, they were coined later to fill up the role gaps in dealing with customers and suppliers respectively (Cisl, 2013). By the end of the mid-1990s, in one of its publications, Gartner Group defined the term ERP II as a system that covers the entire functional business process of an enterprise (Bond et al., 2005).

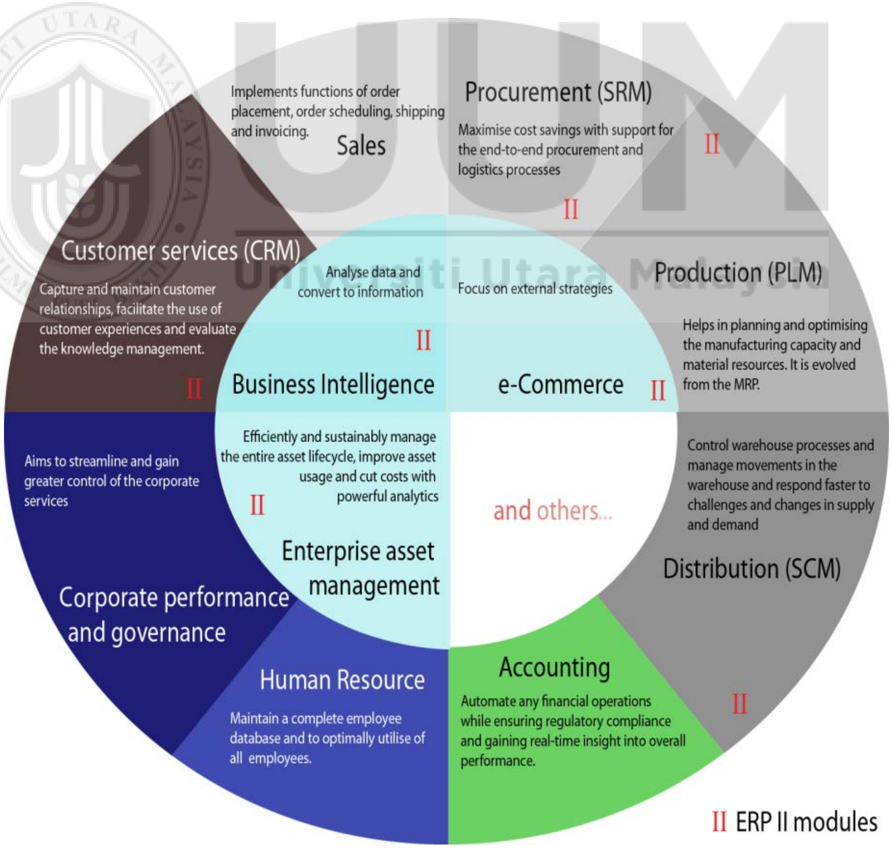




*Figure 2.2* ERP II package, which includes almost all business software

As described in Figure 2.2, the functionality of an ERP system practically covers the entire business operation in any organization today. Essentially, this has made ERP the main product of computer science's application in the business world (Panorama Consulting Solutions, 2016). Furthermore, it is typically an enormous system that is designed to translate a firm's normal business processes into computerized data entry and reporting alone. From the coverage of ERP II shown in Figure 2.2, Sales, CRM, Corporate Performance, SCM, Accounting, Human Resource, Production, Asset Management, Business Intelligent, as well as ecommerce are among the standard modules of any given ERP system (Bond et al., 2005).

ERP implementation failures have been studied throughout numerous research endeavors. According to the KPMG Canada Survey, in 1997 over 61% of the implementation projects that were analyzed were deemed to have failed (Whittaker, 1999). In 2001, Robbins-Gioia Survey found that 51% of the participants viewed their ERP implementation as unsuccessful (Ghosh, 2012). In 2008, the UK's Office for National Statistics found that only 6.5% of the 8000 businesses surveyed were using an ERP system as of 2006. In an article by Bob Lewise in infoworld.com, he even claimed that up to 70% of the ERP implementation attempts failed.



Panorama-Consulting.com in their web site article, the Top Ten Predictions for the ERP Software Industry in 2014 predicted that ERP implementation failure is not going to see a reduction anytime soon, but emerging technologies, such as cloud services and mobile computing will create a positive outlook regarding a higher rate of successful implementation (Panorama Consulting, 2015).

Despite the high implementation failure rate, the need for an ERP system by enterprise organizations and even SMEs is getting more and more important and demanding (Panorama Consulting Solutions, 2016). ERP has become a mandatory infrastructure for serious startup business and a business can scarcely survive the competition without an ERP deployment (Kumar & Hillegersberg, 2000). ERP implementation in an organization represents a unified view of the management processes. In essence, an ERP system provides an organization with valuable information relating to the interaction of its consumers and suppliers with the organization (Shaul & Tauber, 2013). In 2010, the ERP market was estimated to be worth nearly one trillion dollars USD (Bento & Costa, 2013) and this estimation is further heightened with the emergence of cloud computing, which provides a more accessible solution, not only to the large multi-national organizations but even to the small-medium enterprises (Raihana, 2012).

Looking at the significance of an ERP acquisition, this study acknowledges that as the market value for ERP increases, there is a need to address the failure rate of ERP implementation. Although the present research does not focus on the implementations of ERP per se, the proposed model would hopefully allow for much better implementation and deployment approaches in the future.



## 2.3 Cloud Computing

Cloud Computing has been discussed in many publications and has been defined in several ways. Rimal, Jukan, Katsaros, and Goeleven (2011) defined Cloud Computing as a model of service delivery and access, which is dynamically scalable and virtualized all of the resources are provided as a service over the Internet. Cloud offers new operating and business models that allow consumers to pay for the resources they effectively use without heavy upfront investment (Rimal et al., 2011). Armbrust, Joseph, Katz, and Patterson (2009) from U.C. Berkeley referred Cloud Computing as the application, hardware, and system software delivered as a service over the Internet. The term cloud refers to the datacenter, hardware, and software, thus the services offered in the cloud environment are made available in a pay-as-you-go manner (Armbrust et al., 2009).

Cloud Computing architecture is somewhat similar to the timesharing computer model of the 60s; wherein, there is a central server relaying data to dumb terminal located within the network (Kim, Kim, Lee, & Lee, 2009). Instead of a time-sharing server, a Cloud Computing server utilizes a web stateless server as its central server. Cloud Computing Architecture can be categorized into three different models. The aforementioned models are Public Cloud, Private Cloud, and Hybrid Cloud (Raihana, 2012). Figures 2.3 show a sketch of a Cloud ERP model. Some documentations include Community Cloud as the 4th Cloud Computing Model (Bakshi, 2014). Therefore, Community Cloud is introduced as a 4th Cloud Computing Model, Hybrid Cloud is defined as a Cloud Infrastructure Model that consists of three distinct cloud infrastructures (private, community, or public) that remain unique, separate entities, but are bound together by standardized or proprietary technology that enables data and application portability (Schubert & Adisa, 2011).

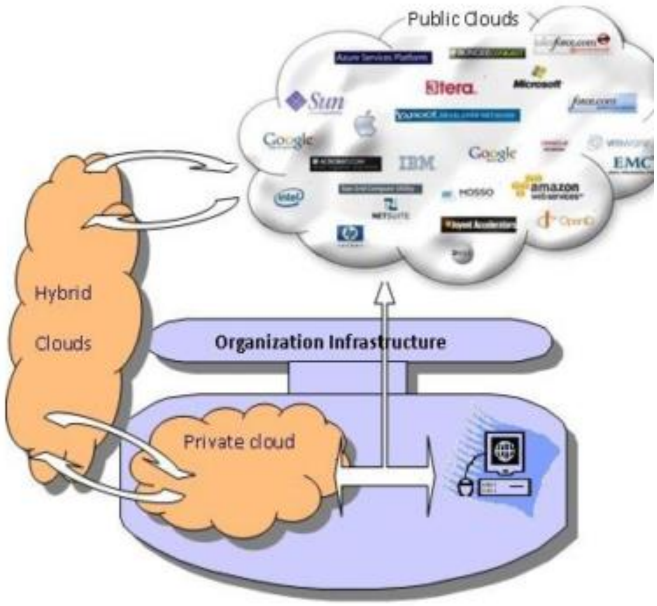



*Figure 2.3* Cloud Computing Model (Raihana, 2012)

When the cloud is made available as pay per use over the internet, it is referred to as a public cloud (Armbrust et al., 2009). It is an internet application hosted over the open internet. Typical a public cloud is owned by an organization that is selling its services and its infrastructure in such a way as to be made available to the general public or large industry groups (Raihana, 2012). The obvious examples of public cloud are the iCloud, Google, and Microsoft 360, etc.



Private cloud, however, varies from public cloud in the way it is made available to its clients. It is meant to made available solely within a specific community or organization. Functionality wise, private cloud is similar to that of public cloud; accordingly it is operated for a particular organization and its physical infrastructure may be managed by themselves, by a third party, on site, or even off site (Raihana, 2012).

As its name suggest, the hybrid cloud is a combination of both public and private cloud models in terms of its service availability. There is the existence of data portability among the cloud by using a proprietary technology (Raihana, 2012). For authors who include Community Cloud as another Cloud Model under the concept of hybrid cloud, the term is defined as a composite of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound by standardized or proprietary technology that enables data and application portability (Badger, Grance, Patt Corner, & Voas, 2011). For the purpose of simplification, as well as to avoid ambiguity, this paper accepts that Hybrid Cloud is a Cloud Model that is composed of Private Cloud and Public Cloud, as well as Community Cloud with managed accessibility controlled by a private organization.



Cloud Computing provides an opportunity to deploy a Service Oriented Architecture (SOA) model into practice (Rabelo, Noran, & Bernus, 2015). SOA is a computer software design architectural pattern, in which application components provide services to other components via a communications protocol over a network. The principles of SOA are independent of any vendor, product, or technology (MSDN, 2014). In Cloud computing, the SOA model is used to define the services offered over the Cloud in various layers (Rabelo et al., 2015).

Despite the many published versions of the definition, generally significant services over the cloud can be categorized into three core types, those are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) (Raihana, 2012; Torkashvan, 2009). Figure 2.4 shows a diagram depicting a typical service model. Some authors use XaaS to define everything as a Service that is delivered over the Cloud, which may make it possible to create an ERPaaS (Rimal et al., 2011). In this paper, the term will mainly be used to denote the three service models, namely IaaS, PaaS, and SaaS to illustrate that Cloud ERP is, in fact, a Cloud Service that involves IaaS, PaaS, and SaaS.

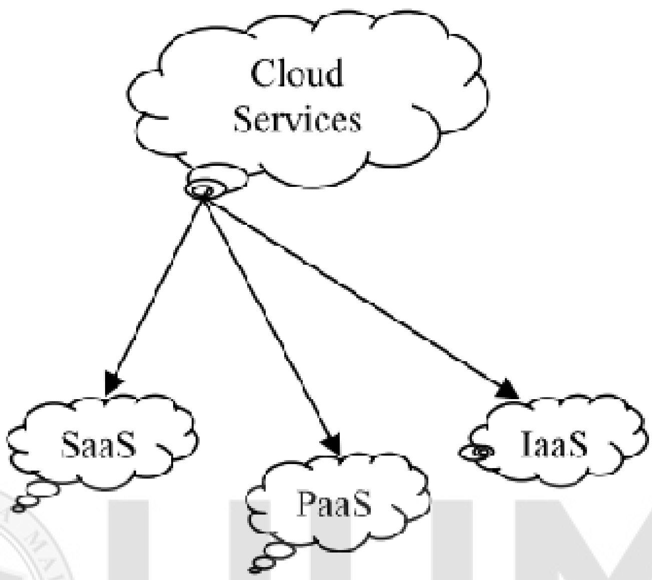



*Figure 2.4* Cloud Services (Raihana, 2012)



SaaS is defined as a cloud application delivered over the cloud or internet with a specific Service Level Agreement (SLA) between that of the provider with that of the subscriber (Zhu & Wang, 2009). The main part of the software remains in the cloud server infrastructure, thus making maintenance, support, and upgrades easier. Typically, a web-based application is an example of a SaaS model application. (Ju, Wang, Fu, Wu, & Lin, 2010; Mell, Grance, & NIST, 2011).

PaaS is a software layer service offered as a platform that can be used to construct higher level platform, which provides interconnecting application objects, programming scripting tools, and storage services (Badger et al., 2011). One of the most prevalent capabilities of PaaS is to provide scalability to ensure computing resources are readily available as the workload increases (W. Tsai, Li, & Esmaeili, 2012). Example of PaaS is Google the chrome platform, Apple IOS platform, and Android platform (Ju et al., 2010).

Typically, IaaS is the lowest level of service offered by a cloud, as it offers hardware server infrastructure virtualization as a service over the cloud (Badger et al., 2011). The customers can use the IaaS to run their own software applications; generally, this type of platform is a pay-per-use or on-demand model. These models simplify the client's need to acquire and manage hardware server infrastructure (NIST, 2011).

In order to achieve the economics of scale of Cloud Computing, a SaaS developer has to leverage a multi tenancy feature (Betts, Homer, Jezierski, Narumoto, & Zhang, 2012). Software Multi-tenancy refers to software architecture, in which a single instance of software from the server operates to serve multiple tenants (Bezemer, Zaidman, Platzbeecker, Hurkmans, & Hart, 2010; Cai et al., 2010). A tenant is a group of users who share common access to specific resources related to the software instance (Schroeter, Mucha, Muth, Jugel, & Lochau, 2012). On the contrary, Multi-Instance architecture terminology refers to a multiple tenant environment that uses a separate software instance for every tenant (Krebs, Momm, & Kounev, 2012). Multi tenancy in SaaS has been a hot topic for discussion ever since the emergence of Cloud applications. Krebs et al. (2012), in his paper discusses about the Architectural concern of SaaS Multi-tenancy. The above researchers mentioned about various layers of sharing; as such, data center, virtualization, and middleware does not constitute a Multi-Tenancy Application (MTA). Instead, a MTA is about the sharing of a single instance of application code among multi tenants. The authors further differentiate a MTA from closely related terms, such as Multi Instance, Multiple Application Deployment, and Tenant Space. Lastly, the authors discuss the higher-level design concerns, such as Affinity, Persistence Design, Performance Isolation, QoS differentiation, and Customization. They emphasize that Customization capabilities are the key enabler to a MTA. In their terminology, Customization is the ability to handle different tenant specific configurations related to the User Interface, system functions, and the services referenced (Krebs et al., 2012).

Krebs et al. (2012) also addressed the importance of a MTA in a SaaS application and likewise stressed that Customization is the key enabler for a MTA. In their paper, a MTA has been specified as a software instance sharing capability and not the hardware and middle ware sharing sectors. They have eliminated other layers and have left only the software layer as a MTA. From this context, if a layer is created within the software itself, it will be part of a MTA. Jepsen (2015) argues that the objective of a MTA is to achieve economic of scale of the SaaS applications. With cheap hardware and platform resources, a MTA is not so relevant anymore. In a Cloud ERP deployment, the project value involved is much higher than any single, individual SaaS application. Typical an ERP customer will try to avoid a MTA, as it creates a data security concern. The previous research studies relating to Cloud ERP and its challenges will be discussed in the following section.

## 2.4 ERP in Cloud Computing

ERP system deployed in the Cloud SOA model is referred to as Cloud ERP (Duan, Faker, Fesak, & Stuart, 2012). It serves the purpose of traditional ERP but with the advantages of SaaS architecture. As the cloud-computing environment is taking over the traditional software model, cloud ERP is the future of ERP model. Raihana

(2012) defines Cloud ERP as the SaaS solution model of ERP delivered in Cloud environment and deploying virtualization and load balancing technology making it adaptable, flexible, agile, scalable, efficient, and affordable. Due to the ease of acquisition, Cloud ERP is regarded as a game changer for innovative organizations, allowing small and midsize enterprise access to advanced business applications (Lenart, 2011). This is further supported in a trending report that claims although on premise implementations are prevalent among big corporations, it is common to see the adoption of Cloud ERP as the organizations get smaller (Castellina, 2011).

Generally, a few main factors that drive the adoption of Cloud ERP are similar to those factors for SaaS. Among those factors are low entry cost,  flexibility as well as scalability (Timm Seitz, 2010). However, as our research looks at Cloud ERP from Cloud ERP provider's perspective, the next subsections focus on several examples of challenges that contribute to the overall complexity of Cloud ERP backend management.



## 2.5 Postmodern ERP

Gartner has been a major research organization that influences the Cloud Computing and ERP market direction. It is very informative for this research to take into account Gartner view of Cloud ERP and its forecast. In their research report, (Gartner, 2016) has created a term Postmodern ERP and predicted that highly customized legacy ERP system will be replaced by a hybrid model of on premise ERP blended with Cloud SaaS applications. They forecasted that in the years to come, the future ERP model is made of on premise ERP integrated with cloud ERP or SaaS solution.

Since Gartner has originally coined the term ERP, the Postmodern ERP mentioned by Gartner in this report is not to be taken lightly. By today, many ERP company has tried to incorporate the characteristic of Postmodern ERP mention in this report. It is perhaps a marketing strategy but for the purpose of this research to create a Conceptual Model of Cloud ERP, it is wise to incorporate the features of Postmodern ERP as well. Below are the lists of characteristics of Postmodern ERP, interpreted from this report.

i. Flexible and dynamic Customization.

ii. Loosely coupled, modular, and readily integrated with other Cloud applications.

iii. Hybrid in nature that support integration with on premise ERP system.

iv. Supported by business process outsourcer.

One important note taken from the report is the word by Andy Kyte, vice president and Gartner Fellow is that,

> "Business stakeholders still want these same qualities, but now they assume that these qualities will be present in any software solution, and their requirements have switched to the twin concerns of lowering IT costs and seeking increased flexibility. A system that is not sufficiently flexible to meet changing business demands is an anchor, not a sail, holding the business back, not driving it forward" (Gartner, 2016).

From the statement, the need for high flexibility software updates in the ever-changing business process need is vital, thus translating it to the need of continuous customization of the system.

Taking into account Gartner's prediction of future ERP evolution, in the context of our research, the Cloud ERP referred in this paper is of private or public or hybrid type with dynamic customization and outsourcing features. In the next section, the challenges of current Cloud ERP management are described.

### 2.6 Cloud ERP Management Challenges and Issues

In a journal released in 2010, Hofmann presents the challenges faced by organizations when trying to migrate from on premise set up to the cloud computing environment (Hofmann & Woods, 2010). The challenges described in the journal are also applicable to the migration of traditional ERP system to Cloud ERP as discussed in a research paper by Appandairajan, Zafar Ali Khan, & Madiajagan (2012), who discuss about Cloud ERP – Implementation Strategies and Challenges. In this paper, they mention about the challenges of Cloud ERP as the following:



i. Internet bandwidth and routing management.
ii. SLA on the service availability.
iii. Scalability of the resources provided based on demand.
iv. Risk of Open Accessibility.
v. Security issues such as information secrecy, encryption security and access management.
vi. Data physical location.
vii. Pricing versus business model for ERP delivery.

viii. Lack of industry standards, which make users reluctant to be tied up to a particular provider.

ix. The need for System Customization for larger organization.

The authors made a point about customization level as an important deterrent for adopting Cloud ERP for large organizations. Timm Seitz (2010) emphasizes that lack of customization capability stops large enterprise into moving to Cloud ERP solution. As time progresses, this fact is further reiterated by (Arnesen, 2013) noting that major customizations are not allowed in Cloud ERP solutions. In summary, the technology comparison between Cloud ERP and Conventional ERP is shown in Table 2.1.

Table 2.1
*ERP Technology Comparison*

| Conventional ERP | Cloud ERP |
|---|---|
| High capital expenditure | No capital expenditure |
| Direct & indirect expenditure | No investments |
| Ballooning costs | Low-cost subscription model |
| Long implementation time | Implementable in weeks |
| Rigid | Scalable |
| Limited access | Anytime, anywhere access |
| Upgrade at extra cost | Free upgrade |
| Limited licensing | Flexible licensing |

Most of the research published deals with the challenges of adopting Cloud ERP from implementation point of view. In the context of this research however, the exploration is focused on the challenges to Cloud ERP providers in managing the complexity of Cloud ERP.

Traditional ERP implementation was primarily conducted on premise (Purohit et al., 2012). The concept of the implementation has always been fitting the correct applications to enhance the existing business process of the organization (Pollock et al., 2002). However, with the introduction Cloud ERP, several deployment strategies exist depending on the offering provided to the clients. These deployment architectures are referred to Single Tenant, Multi-Instance, Multi-Tenant with Composite Add-ons and finally, Multi-Tenancy.

Single Tenant cloud application refers to the fact that each tenant capitalizes on its own software instance (Mijač et al., 2013). This is closely related to how a normal traditional ERP is deployed. In a Single Tenant application, the management in terms of customization are carried out by creating subsets in the development tree itself. Multi-Instance architecture terminology refers to multiple tenant environment that use a separate software instance for every tenant (Krebs et al., 2012). Composite add-ons are practically applications built on top of existing core application. In Multi-tenancy architecture, there exists a subset model, which allows for add-ons to be added onto the core applications. In order to achieve the economics of scale of Cloud Computing, SaaS developer has to leverage multi tenancy feature (Betts et al., 2012). Software Multi-tenancy refers to software architecture in which a single instance of software from server operates to serve multiple tenants (Pal, Mandal, & Sarkar, 2015). A tenant is a group of users who share a common access to specific resources to the software instance (Kriouile, El Asri, El Haloui, & Benali, 2015).

Multi tenancy in SaaS has been a hot topic for discussion ever since the emergence of Cloud application. Krebs et al. (2012) in their paper discuss about architectural concerns of SaaS Multi-tenancy. They mentioned about various layer of sharing such as data center, virtualization and middleware does not constitute as a Multi-Tenancy Application (MTA). Instead, the MTA is about the sharing of single instance of application code among multi tenants. The authors further differentiate the MTA from closely related term such as Multi Instance, Multiple Application Deployment, and Tenant Space and finally discuss about higher-level design concern such as Affinity, Persistence Design, Performance Isolation, Quality of Service (QoS) differentiation and customization. They emphasize that customization capability is the key enabler to the MTA. In their terminology, customization is the ability to handle different tenant specific configurations  related to User Interface, systems function and the services referenced (Krebs et al., 2012). The author addressed the important of MTA in SaaS application and also stressed that customization as the key enabler for MTA. In their paper, the MTA has been specified as software instance sharing capability and not the hardware and middle ware sharing. They have eliminated other layers and left only with software layer as MTA. From this context, if a layer is created within the software itself, it will be part of MTA.



Jepsen (2015) argues that the objective of MTA is to achieve economic of scale of SaaS applications. With cheap hardware and platform resources, MTA is not so relevant anymore. In a Cloud ERP deployment, the project value involved is much higher than individual SaaS application whereby typical ERP customers will try to avoid MTA as it creates data security concern.

From the Cloud ERP providers' perspective, Multi-Tenancy solution invokes a different aspect of management headache whereby Cloud ERP providers would then be burdened with trying to manage different tenancy approaches to ensure that the market gets what it wants.

Customization in ERP is almost inevitable to some degree in all ERP implementation project (Scheer & Habermann, 2000). It is required to match between system package and business processes of an organization (Uppström et al., 2015). In the early days of ERP, implementations normally involve a heavy customization process. In fact, early system assumes that ERP providers will adhere to the organization business processes. Following System Development Life Cycle (SDLC), user requirements will precede the process and system analysts will create a System Analysis report for programmers to code. This is normally true during the Client Server age in which the programming is mostly done at Client PC. Over the years, ERP providers have managed to create standard packages of software with the ideal intention to redeploy it to other customers. Configuration option is added to provide minimum variant within a software version. Unfortunately, research show that hardly any organization can fit any other organization system model (Rothenberger & Srite, 2009; Uppström & Hoffsten, 2015). Rothenberger and Srite (2009) in their paper relate the failure of ERP to heavy customization but also mention that it is still unclear why customer still choose to customize the ERP system.

In their paper, Uppström and Hoffsten (2015) presents various ERP Customization options ranging from setting parameters to developing new functionality by modifying source code. Luo and Strong (2004) in their research describes three type of ERP customization options; those are module customization, table customization, and code customization. Module customization refers to the option of selecting which module to implement; table customization involves changing table parameters; and code customization is the code modification to add or modify functionality of the module.

ERP customization can be categorized into a few options. Uppström and Hoffsten (2015) provide an extensive review of the ERP Customization option for ERP for Traditional ERP and also recommend the Customization option applicable for Cloud ERP. Table 2.2 shows their report about On-Premise ERP Customization option review. In their paper, the authors have commented on the terminologies use for the Customization options and then create the effectiveness study for their use in Cloud ERP Customization. Their finding on the effectiveness of all the On-Premise customization options however shows that configuration and interface development are the only applicable or recommended to be discussed within the Cloud ERP domain. They also recommended that new customization option namely packaged customization, conversions, and mobile platforms to be included into the customization option for Cloud ERP. In conclusion, they proposed the need for a new model for On Premise ERP and Cloud ERP Customization Model to be developed to solve the premise ambiguity on the option type (Uppström et al., 2015).



Table 2.2
*On Premise ERP Customization Option Review (Uppström et al., 2015)*

| **Customization Option** | **Status** |
|---|---|
| Configuration | Common customization option. Configuration encapsulates more possible changes than before, e.g. role interfaces. Lack of consensus in regards to definition of configuration. |
| Bolts-On | Common customization option but encapsulates more possible changes than before. Development and dissemination facilitated through marketplaces. |
| Screen Masks | Old fashioned, not commonly used and has been replaced with other techniques such as web and mobile interfaces. Configuration as to a large extent replaced the need for screen masks. |
| Extended Report | Outdated customization option. Replaced with configurations, workflow layers and workflow engines. |
| Workflow Programming | Outdated customization option. Replaced with configurations, workflow layers and workflow engines. |

Table 2.2 continued.

| | |
|---|---|
| User Exit | Valid and a (very) common customization option. Most classical option of customization, and you cannot do implementation projects without using them. |
| ERP Programming | Status unclear. Differs between different vendors and perceived as an unclear option among the respondents. |
| Interface Development | Important and common customization option but the naming is outdated. Interfaces are often simpler to develop today due to more tool. |
| Package Code Modification | Regarded as a still valid option as well as difficult to perform and should be avoided. In almost all implementation project some kind of coding is necessary, but an attempt must be made to keep it at a minimum to facilitate upgrades and support of the system. |

Configuration on the other hand, is a method of personalizing (or sometime referred to customizing) user requirement using the standard version of software. This is a major difference between SaaS software and cloud ERP as opposed to standard ERP or traditional ERP.

It is interesting to see the types of ERP customization options presented in the paper although the author agreed that the customization options extracted from previous research were hardly understood by ERP deployment practitioners. In Cloud computing, in order to achieve economy of scale, the software should be provided in single code base and operated as a single instance with multi tenancy. In order to meet these constraints and requirements from various customers, SaaS is deemed to be highly configurable. To develop configurable SaaS, it is vital to know and analyze configuration requirements from the beginning of the software development (Shim et al., 2011).

This is a very challenging task as it is almost impossible to analyze requirements from millions of different versions of customer requirements. Thus as of today, ERP or cloud ERP is still bound to customization (Dittrich & Vaucouleur, 2008; Pollock et al., 2002; Davenport, 2016). The dilemma created here is that, as Cloud ERP is a SaaS application, it is expected to have a standard version with high configurability, while ERP requirement still demands customization that is impossible to achieve with configurability. Every new implementation, ERP providers are bound to face a new business processes that cannot be achieved using standard version of software.

Another point is, even if one can accept that ideally Cloud ERP implementation should be done with configuration and minimum code customization, for the developers, they need to keep updating their standard module to include newly encountered functionality variant from customer requirement into the configurability option. As the Cloud ERP technology is growing and maturing, ERP developers are trying to adapt all possible functionality variant into the standard modules. This can be done in discrete updating like SAP R1 to R2 and R3 or dynamic versioning as fast as typical android application today. In this paper, as the Cloud Computing itself is growing at amazing rate, the assumption is made that the need for dynamic or continuous Cloud ERP customization features are critical. Dynamic or interactive customization pertains to the concept of customizations activities done constantly due to the fact that business requirements are no longer static. This is also true with regards to frequent software updates, which allows Cloud ERP applications to be up-to-date in terms of technological advancements and business practice requirements.

Software updates is one of the most crucial maintenance activity to be carried out as a measure to ensure all software are updated and patched (K. Vaniea & Rashidi, 2016). Updating software is no longer due to just fixing bugs but developers have to keep up with any technological advancements as well as any new business practice introduced (Gu et al., 2012). Software itself is an intangible product, which basically indicates there is no finished product and thus making it not perfect (Chen, Yu, Hang, Zang, & Yew, 2011). As with any unfinished product, a software would always be required to be patched or improved upon the discovery of new technological advancements or a new business practice takes effect (Chen et al., 2011). The simplest example would be the introduction of Goods and Services Tax (GST) imposed in Malaysia in 2015 as software providers scrambled to ensure that their Accounting system be it cloud or otherwise to be compliant to the necessary guidelines. Therefore, it can be observed the frequent software updates whether required or imposed poses a management complexity for Cloud ERP providers.

When looking at the necessity to frequently update Cloud based systems, to update Cloud ERP systems would be even more complex as it is understood that Cloud ERP is not just one complete entity, but instead a collection of SaaS applications due to the need for system modularization.



Modularity is a highly important factor in designing software systems as it dictates the quality of a particular software in terms of the different aspects of maintaining it as well as being able to accommodate potential changes (Huynh & Cai, 2007). Traditional ERP applications are known to be designed with modularity features to facilitate implementation in stages (Nagpal, Khatri, & Kumar, 2015; Stoilov & Stoilova, 2008). In the context of Cloud ERP, modularization provides a system which are segmented and integrated on-demand depending on the tenant’s preference. To allow for such loosely coupled and integrate-ready applications generally means more workload for the Cloud ERP providers and ultimately contributes to the back-end management complexity.

As more cloud applications become complex and utilized by the public, cloud application providers are forced to rethink their strategies in system deployment. The ability to allocate the necessary resource to ensure that all clients are able to utilize the application during peak hours is a challenge on its own (Shi, Dong, Zhang, Jin, & Luo, n.d.). Cloud application providers are also required to develop a way to provide a reliable infrastructure which promises users enough resources when needed (Rimal et al., 2011). With the introduction of cloud computing, the allocation of resources has become one of the major discussion factor. The peak-time usage of application on the cloud presents a complexity to application providers to ensure that users are able to access and use the application as though the resources are fully dedicated to them.

By providing a ready-made easy to select templates, the cloud provider ensures that additional new tenancy is able to be allocated with the necessary resources quicker and more streamlined. However, according to a research journal by Gohad, Ponnalagu and Narenda (2012), this would pose a long-term challenge for the providers. In the effort to overcome this challenge, the authors proposed an extensible dynamic provisioning framework. The framework is based by defining a Tenancy Requirements Model (TRM) and the Quality of Service (QoS) characteristics to help map provisioned resources with tenants.

The research presented by (Shi et al., n.d.), however, chose to shift the attention on to how to effectively place the virtual machines into physical servers with multidimensional resource requirements. The authors proposed a novel resource provisioning strategy, which allows for virtual machine provisioning for hosting service and the placement of virtual machines into servers. The method considers the number of virtual machines to be provided for each service using a queuing theory thus converting the placement problem as a variant of cutting stock problem. This is turn allows the method to decide how many servers should be provided to solve the issue.

Authors in the research paper titled "SINC – An Information-Centric Approach for End-to-End Internet of Things (IoT) Cloud Resource Provisioning" (Truong, 2016) try to address the growing need for effective provisioning and management of resources in an IoT-based system. To illustrate their proposed solution, the authors suggests an information-centric approach towards the resource provisioning and management. The underlying concept of the model is by virtualizing access to underlying IoT resources and thus leveraging APIs to manipulate the resources.

Another critical factor for Cloud ERP offerings is the security management in which it provides. Security management involves controlled access, prevention of unlawful entry as well as data integrity (Maheshwari & Sharma, 2015). This factor alone would cause pause for financial organizations to adopt Cloud ERP migration and prefers to employ an on premise set up which subsequently allows for the organization to manage its own security precautions. Cloud ERP security issues can be classified as physical security, transmission security and storage security (Appandairajan et al., 2012). Even before the existence of cloud applications, data sensitivity and security has been a major concern with big multi-national corporations. This has generally led to the implementation of on premise systems, which supposedly provides better security features as well as being in a closed loop system. Several researches have pointed out the security concerns in which resides in cloud computing. In the research paper published by Anwar and Imran, (2015), the authors raised the concern that cloud computing, particularly in multi-tenancy architecture, poses privacy and security issues specifically relating to personal health information. The proposed model suggested by the author takes the form of an ontological model for healthcare workflow and multi-tenancy, which are then applied with the necessary requirements to generate stricter access control policies. To achieve this, the authors capitalized on the Semantic Web Rule Language to represent access control policies as rules which in turn be simulated in a cloud-based healthcare environment. The result as claimed by the authors were able to address the existing vulnerabilities of multi-tenancy. Agreeing to that notion of cloud computing flaws, (Alouane & Bakkali, 2015) discussed the serious risks related to the use of Cloud computing. The authors perceived that temporary or permanent loss of data, data security, lack of traceability, and accountability are some of the pertinent risks faced while adopting a Cloud computing architecture. In their research, the authors focused on three major elements of cloud computing in terms of security, privacy, and trust.



As more and more complexities are defined, they begin to present a very prominent issue. Stemming from the need of having separate yet loosely coupled modules and the flexibility of customizations, Cloud ERP applications in general are segmented based on the functionalities and business process that it represents, which eventually requires Cloud ERP providers to have separate team of programmers to handle each module. These factors although offers simplicity to the users, present a problematic situation to Cloud ERP providers whom are required to employ more programmers/developers to maintain it. With the employment of new developers, Cloud ERP providers are now faced with a crisis, which involves the ability to ensure that the newly hired programmers are able to adopt and adapt to the operations and technological standard upheld. This is also a deciding factor in determining the productivity and quality of the programmer compared to experienced programmers within the organization. Every programmer would then bring his/her own expertise, knowledge and skills, which the current research will refer to as “craftsmanship” into the workplace (Bryan, 2012). The different abilities of these developers would ultimately be reflected in the quality of the system or application developed.

As with any organization, staff turnover is a normal occurrence. However, the complication is magnified in a knowledge based organization such as software developer due to the knowledge loss once a senior and experienced programmer leaves the company (Schilling, Laumer, & Weitzel, 2012). This is a situation, which can be referred to as “knowledge retention management”, whereby as the turnover increases, so does the amount of the knowledge loss to the organization. The resignation of an experienced programmer leaves behind a sizeable gap in terms of the quality of the application to be maintained. A newly hired programmer would have to relearn the methods and familiarization of the coding structure, which would subsequently impact the quality of the assigned application.

The research has identified multi-tenancy, customization with dynamic requests, frequent software updates, system modularity, resource provisioning, security measures, knowledge management, and inconsistent software quality as some of the challenges that contribute to the overall complexity in managing Cloud ERP as shown in Table 2.3, which will be presented in the following section.

Table 2.3

*Cloud ERP Management Challenges and Issues*

| **No** | **Journal Titles/Cloud ERP Challenges** | **Multi-Tenancy** | **Customization** | **Frequent Software Updates** | **Modularity Features** | **Resource Provisioning** | **Security Measure** | **Knowledge Management** | **Inconsistent Software Quality** |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Virtualization-Based Techniques for Enabling Multi-tenant Management Tools; *Chang-Hao Tsai, Yaoping Ruan, Sambit Sahu, Anees Shaikh, Kang Shin, Alexander Clemm, Lisandro Granville, Rolf Stadler, 2007* | x | | x | x | | | | |
| 2 | A transparent approach of enabling saas multi-tenancy in the cloud; *Hong Cai, Ning Wang, Ming Jun Zhou, 2010* | x | | | | | | | |
| 3 | Multi Tenancy Issues in Cloud Computing for SaaS Environment; *R Ashalatha, Jayashree Agarkhed, 2016* | x | | | | | | | |
| 4 | Application Multi-Tenancy for Software as a Service; *Sanjukta Pal, Amit Kr Mandal, Anirban Sarkar, 2015* | x | | | | | | | |
| 5 | Elastic Multi-tenant Business Process Based Service Pattern in Cloud Computing *Wael Sellami, Hatem Hadj Kacem, Ahmed Hadj Kacem, 2014* | x | | | | | | | |
| 6 | Software Customization Based on Model-Driven Architecture Over SaaS Platforms *Zhu, Xiyong & Wang, Shixiong, 2009* | | x | | | | | | |
| 7 | Enabling Enterprise Composite Applications on Top of ERP Systems *Borovskiy, Vadym, Zeier, Alexander, Koch, Wolfgang, Plattner, Hasso 2009* | | x | | | | | | |
| 8 | An Overview of Variability Management in Cloud Services *Khadija Aouzal, Hatim Hafiddi, Mohamed Dahchour, 2015* | | x | | | | | | |
| 9 | New implications for customization of ERP systems *Elin Uppstrom, Carl Mikael Lonn, Madeleine Hoffsten, Joakim Thorstrom, 2015* | | x | | | | | | |

Table 2.3 continued

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 10 | Cloud ERP System Customization Challenges<br>*Marko Mijač, Ruben Picek, Zlatko Stapić, 2013* | | x | | | | |
| 11 | Cooperative update: a new model for dependable live update<br>*Giuffrida, Cristiano & Tanenbaum, Andrew S., 2009* | | | x | | | |
| 12 | Tales of Software Updates: The process of updating software<br>*Kami Vaniea, Yasmeen Rashidi, 2016* | | | x | | | |
| 13 | Betrayed by updates<br>*Kami E. Vaniea, Emilee Rader, Rick Wash, 2014* | | | x | | | |
| 14 | Dynamic software update<br>*R. Jhanwar, T. Yaryan, 2007* | | | x | | | |
| 15 | A Requirements Elicitation Approach for Cloud Based Software Product Line ERPs<br>*Ali, Nasr, Gheith, 2016* | | | | x | | |
| 16 | A Cost-Based Approach to Software Product Line Management<br>*Schackmann, Holger & Lichter, Horst, 2006* | | | | x | | |
| 17 | Sharing with a difference: Realizing service-based SaaS applications with runtime sharing and variation in dynamic software product lines<br>*Kumara, Indika, Han, Jun, Colman, Alan Nguyen, Tuan & Kapuruge, Malinda, 2013* | | | | x | | |
| 18 | Model Driven Provisioning in Multi-tenant Clouds<br>*Atul Gohad, Karthikeyan Ponnalagu, Nanjangud G. Narenda, 2012* | x | x | | | x | |
| 19 | Resource Provisioning Optimization for Service Hosting on Cloud Platform<br>*Shi, Jiyuan Dong, Fang, Zhang, Jinghui, Jin, Jiahui & Luo, Junzhou, 2016* | | | | | x | |
| 20 | SINC – An Information-Centric Approach for End-to-End IoT Cloud Resource Provisioning<br>*Truong, Hong-linh, 2016* | | | | | x | |
| 21 | Access Control for Multi-tenancy in Cloud-Based Health Information Systems<br>*Anwar, Mohd & Imran, Ashiq, 2015* | x | | | | | x |
| 22 | Ten security practices to a formidable ERP system<br>*Maheshwari, Shivang & Sharma, Charu, 2015* | | | | | | x |

Table 2.3 continued

| | | | | |
|---|---|---|---|---|
| 23 | Security, Privacy and Trust in Cloud Computing: A Comparative Study<br>*Alouane, Meryeme & Bakkali, Hanan E L, 2015* | x | | |
| 24 | Knowledge Management Strategy and Structure in Service Sector<br>*Gulati, V. P., Srivastava, Shilpa, 2014* | | x | |
| 25 | Reusable active learning system for improving the knowledge retention and better knowledge management<br>*Sridharan, Bhavani, Kinshuk, 2003* | | x | |
| 26 | Train and retain: The impact of mentoring on the retention of FLOSS developers<br>*Schilling, A, Laumer, S Weitzel, T, 2012* | | x | x |
| 27 | Not all programmers are created equal<br>*Bryan, G E, 2012* | | | x |
| 28 | Quantitative evaluation of software quality<br>*Boehm, B W 1976* | | | x |
| 29 | Investigating the effect of pair programming and software size on software quality and programmer productivity<br>*Sison, Raymund 2009* | | | x |

In summary from the literature review, multi tenancy features, dynamic customization requests, frequent software version updates, and upholding modularity features of Cloud ERP have contributed to the high complexity and heavy workload of Cloud ERP production processes. On the other hand, craftsmanship effects due to varying staff skills creates functional and structural software quality problems faced by the Cloud ERP provider has resulted in inconsistent software quality issue. Finally, due to the vast amount of knowledge and over dependency on knowledge workers, knowledge management and retention poses another big problem to Cloud ERP providers. The problem mentioned has no doubt been included in the previous researches that were tabulated in Table 2.3. While all the research mentioned above did address its objectives, they did not address the overall view from the back-end management point of view. The solution provided was focused solely on the specific scope of the issue, i.e. multi-tenancy alone or a model to solve customization requirements. Each of the solution presented did not take into consideration other aspects of Cloud ERP software production which in turn was unable to be adopted to provide a complete solution model to solve all the research problem components highlighted in this research. Moreover, the listed research was also focused on either traditional software development or a SaaS application which basically meant they were not intended for Cloud ERP production environment which is basically a group of integrated SaaS solutions. Hence, what is needed now is a model that provides a systematic management reference, which can indirectly provide a solution to all of the problems above.

This ultimately corresponds perfectly to our research in formulating a model, which aims to improve the management complexities faced by Cloud ERP providers, high workload, inconsistent software quality and knowledge retention management. In the next section, this research has reviewed several approaches to be taken into consideration to formulate the best approach in an attempt to solve the research problem components.



## 2.7 Approach Analysis and Justification

This research has considered several approaches which was felt that could be used to formulate the model. The several approaches which were considered those are model-driven approach, resource-based theory approach, software product line approach and manufacturing approach. From all the approaches considered, the manufacturing model was discovered to provide a method to address each of the research problem components (product determination, standard operating procedures, resource allocation, industrialization vs craftsmanship). The closest approach that is similar to manufacturing in the software development industry is software factory. In order to formulate the model however, understanding the concept of a standard manufacturing model was crucial as software factory is also based on the manufacturing-typed elements and is discussed next.

A standard manufacturing or production system has many factors of consideration based on the type of product to be manufactured (Menascé, Krishnamoorthy, & Brodsky, 2015). However, there are elements, which are common to every single type of product. Generally, to start up a manufacturing facility, several factors have to be taken into account. Referring to Table 2.4, most of the research on manufacturing processes points to four major areas in dealing with manufacturing a product, which are Product Development, Platform Commonality, Process Flow, and Quality Control. Due to the nature of Software Factory approach is to mimic manufacturing model, these four major areas were used as our reference in model development. Other factors have been excluded, such as human resources and cost consideration due to the time frame for the whole research.



Product development is the most basic consideration in a manufacturing process (Gabriel, 2013). The type of product manufactured defines the manufacturing capability of the factory. As an example, to manufacture an electronic product, the factory needs to be equipped with machinery that is capable of producing and assembling that particular electronic product. To compete in a very competitive marketplace, having a product alone is not enough. Manufacturers are also expanding the product lines with the belief that developing product families are able to promote economy of scale through platform commonality (Jiao, Simpson, & Siddique, 2007).

Once the product development is complete, the process to produce or manufacture the said product is the next crucial element for any manufacturers (Camisón & Villar López, 2010). The processes defined to manufacture a specific product dictates the amount of time required, the process needed to provide the outcome, the types of process required and which station is needed (Kamrani, Smadi, & Salhieh, 2012). Within the context of the processes however, quality assurance is another element which manufacturers pay attention to (Wuest, Irgens, Thoben, Wuest, & Thoben, 2014). Quality assurance has its own set of processes to be completed to make sure that the product manufactured follow the quality standard set by the company (Jain, 2012).

Knowledge Management System has been included as a component as a means to support the sustainability and knowledge retention management component of the CEF Model. Due to the vast amount of knowledge in the Cloud ERP domain, it is essential to include a knowledge management component to ensure that all knowledge obtained or learned in the organization is able to be stored and kept for future references.

Table 2.4

*Consideration Factors for Product Manufacturing*

| **Journal Titles/Manufacturing Consideration Factors** | **Product Development** | **Platform Commonality** | **Process Flow** | **Quality Control** |
|---|---|---|---|---|
| The process of formation of manufacturing in small and medium-sized enterprises in Iran (Talebi & Khaksar, 2011) | √ | | √ | |
| Two-phase methodology for customized product design and manufacturing (Kamrani et al., 2012) | √ | | | |
| An examination of the relationship between manufacturing flexibility and firm performance (Camisón & Villar López, 2010) | √ | | | |
| Product family design and platform-based product development: A state-of-the-art review (Jiao et al., 2007) | √ | √ | | |
| Implementation of Mass Customization Tools in Small and Medium Enterprises (Stojanova, Suzic, & Orcik, 2012) | √ | | | |
| An integrated approach to optimize design, marketing, and manufacturing objectives during product development. (B Nepal, 2014) | √ | √ | | |
| Manufacturing Complexity: The Effects of Common Attributes of Manufacturing System Design on Performance. (Gabriel, 2013) | | | √ | |
| Component and process commonalities in production system under various uncertain factors (Wazed, Ahmed, & Nukman, 2010) | | √ | √ | |
| A manufacturing-oriented approach for multi-platforming product family design with modified genetic algorithm (Z. Liu, Wong, & Lee, 2011) | √ | √ | | |
| Strategic inventory allocation for product platform strategy (Kristianto, Helo, & Takala, 2010) | √ | √ | | |
| Improving manufacturing process for biomedical products: a case study (Bimal Nepal, Natarajarathinam, & Balla, 2011) | | | √ | √ |
| A strategic and operational approach to assess the lean performance in radial tire manufacturing in India (V. Gupta, Acharya, & Patwardhan, 2013) | | | √ | √ |
| A review of agile and lean manufacturing as issues in selected international and national research and development programs and roadmaps. (Castro, Putnik, & Shah, 2012) | | | √ | √ |

Table 2.4 continued

| | | | | |
|---|---|---|---|---|
| An evaluation of just in time (JIT) implementation on manufacturing performance in Indian industry (Singh & Ahuja, 2014) | | | √ | √ |
| A framework for the quality-oriented design of micro manufacturing process chains (Rippel, Ltjen, & Scholz-Reiter, 2014) | | | | √ |
| Monitoring processes through inventory and manufacturing lead time (Cuatrecasas-Arbós, Fortuny-Santos, Ruiz-de-Arbulo-López, & Vintró-Sanchez, 2015) | | | | √ |
| Manufacturing Strategy Design Process Improvement Through Enterprise Engineering Recommendations (Moura, Lima, Deschamps, & Gouvea, 2015) | | | √ | √ |
| The impact of TQM implementation on the organizational performance of Iranian manufacturing SMEs (Valmohammadi, 2011) | | | | √ |
| Process capability – a surrogate measure of process robustness: a case study (Mondal, 2016) | | | √ | |
| Core process management practices, quality tools and quality improvement in ISO 9001 certified manufacturing companies (Psomas, Fotopoulos, & Kafetzopoulos, 2011) | | | √ | √ |
| An integrated model for performance management of manufacturing units (Parthiban & Goh, 2011) | | | | √ |
| Autonomic smart manufacturing (Menascé et al., 2015) | √ | √ | √ | |



## 2.8 Software Factory Model and Its Adoption

In order to tackle the complexity of Cloud ERP challenges discussed in the previous section, a Software Factory approach was chosen to be employed. It is a concept introduced by Bratman and Court (1975) referring to it as “an integrated set of tools that supports the concepts of structured programming, program development, program production libraries, and incorporates hierarchically structured program modules as the basic unit of production”. With traditional development method, software production remains reliant on the craftsmanship of a development team or programmers. The process involved is also labor intensive and manual in nature. In order to reduce cost , delivery time, and to improve software quality, there is a need for a transition to a more automated approach (Greenfield & Short, 2003). In principle, Software Factory promotes the automation of software development cycle in commercial environment to be like a manufacturing process model. Helton (2010) describes component assembly is central to the software factory approach, but also includes modeling techniques, standardization requirements, and SPL concepts, to develop applications with available technology platforms. He also describes the motivating factors that may drive the adoption of Software Factory. Among those are software crisis and ERP failure problem. The authors also explained that Software Factory adoption faces challenges, such as immature underlying methodology, inadequate tools, the complexity of the concept, and inadequate demonstration.

This statement indirectly applies to this research's objectives as the objectives are to develop a Model using Software Factory approach. It can be used as a guidance in constructing the CEF Model.



## 2.9 Review of Existing Journals on Software Factory and Its Adoption Models

In this section, existing journals dealing with the definition and adoption of Software Factory models are reviewed to further allow for the discovery of the applicable approach for the current research's Cloud ERP Factory (CEF) model.

### 2.9.1 General Purpose Software Factory

The term Software Factory was coined in an article published by Harvey Bratman and Court, (1975) as a part of a software engineering project conducted by the Software Development Corporation. According to Bratman (1975), the term refers to an integrated set of software development tools to support a disciplined and repeatable approach to software development. The tools believed to be able to increase software reliability and control software production costs were intended as a replacement to the “ad-hoc” nature of software development with standard engineering techniques. Bratman even suggested that the existing software development process lacks methodological and well-founded body of knowledge. These factors were listed as lack of discipline and repeatability, lack of development visibility, changing performance requirements, lack of design and verification tools, and lack of software reusability. To tackle all these factors, Bratman proposed a revolutionary architecture of components to illustrate the integrated and extensible facility of software development tools.

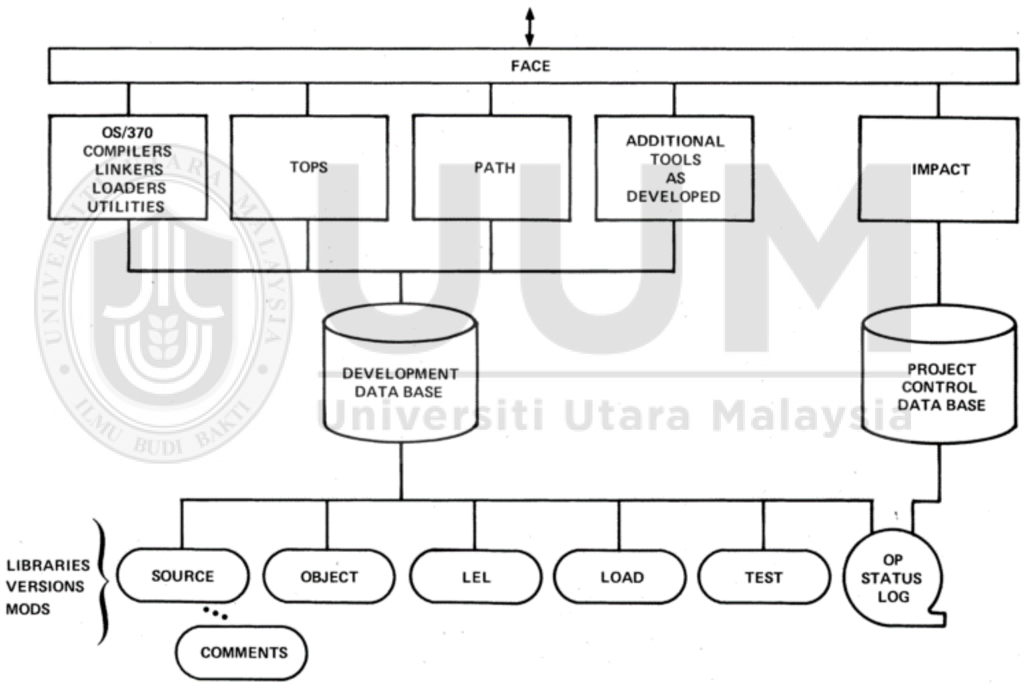



*Figure 2.5* Software Factory Architecture (Bratman & Court, 1975)

With reference to Figure 2.5, Bratman suggests a structure that consists of Factory Access and Control Executive (FACE), Integrated Management, Project Analysis, and Control Technique (IMPACT), and Project Development Database. Each component plays an important role as a control structure for the various Factory processors. To complement the processors in the structure, Bratman also introduces

Automation Documentation Tool (AUTODOC), Program Analysis and Test Host (PATH), Test Case Generator (TCG), and Top Down System Developer (TOPS) each with its own role within the proposed structure. Bratman clearly defines software reusability as one of the outcome by the model enhanced with the application of the architecture proposed.

In a research journal published by Fernstrom et.al., (1992), the authors elaborate that the software factory concept symbolizes a desired paradigm shift in which substantial investments can be made at a more acceptable risk level. Fernstrom claims instead of looking into code producing and documentation element that is traditional software development, software factory shifts the focus to the information coordination between the related components. Fernstorm argues that the factory analogy can only be applied in the context of a software production as means to industrialize the production style and disregards the implementation element. To clarify this paradigm shift, Fernstrom introduces Eureka Software Factory (ESF), which combines process modeling with aid integration architecture and reviews CASE environment architecture and two factory experiments to study its implications.

There exist varying defining factors that make Software Factory the ideal candidate in trying to solve the software crisis. In a research journal, Helton, (2010) examines the factors driving developers to consider adopting software factories and the challenges faced while employing this model. Helton suggests that software factory provides a complete framework of software development process, end to end but comes with a huge reliance to other inadequate methodologies and tools. Generally, Helton defines Software Factory as a complete overall framework to improve the entire software development process, from start to finish as means to address the prolonged software crisis, attraction of a coherent approach and inadequacies of related methodologies. Although the adoption of Software Factory is ideal in most cases, Helton views it as only applicable as a conceptual model and unable to be demonstrated effectively. This is due to the challenges in which Helton argues are due to the immaturity of underlying methodologies, inadequacies of required tools and the complexity of software factory itself. Although Helton claims software factory application in a software development is not demonstrable, there is some existing application of software factory in education as well as corporate environment to support this approach in utilizing software factory to the current research.

Since its inception, Software Factory has evolved into a more complex methodology. Authors Canuel & Roberts (2012) in their research and development article defined Software Factory as tools required to allow development and automation of the overall software building process and even added the concept of tools to aid the software development process which includes documentation, wiki, and source code manager. The Figure 2.6 illustrates how the complete workflow is represented and Canuel claimed it would be able to answer the needs of many projects. The authors also discussed the many benefits of adopting Software Factory and how adopting the approach would ultimately provide automatic software deployment and driving of developments. The benefits would then be very appealing as it creates more structured steps in achieving its targets and even more so to the affected stakeholders. Canuel suggested although Software Factory is primarily intended for developer, other stakeholders including business teams, project management, and production teams would greatly benefit as well. The adoption of Software Factory as good as it sounds, also comes with its own set of challenges. One of the most interesting challenge suggested by Canuel is whether Software Factory can be implemented in a cloud computing environment.

The authors further suggest a tool as a set of practices to improve collaboration between development and operation teams named DevOps. It is believed with DevOps, a better Software Factory could be achieved due to the fact software factory is mixed together with the tool which would be able to help solving the issues such as improvement of workflow, environment management, platforms setup, automated steps, software quality, and performance detection among others.

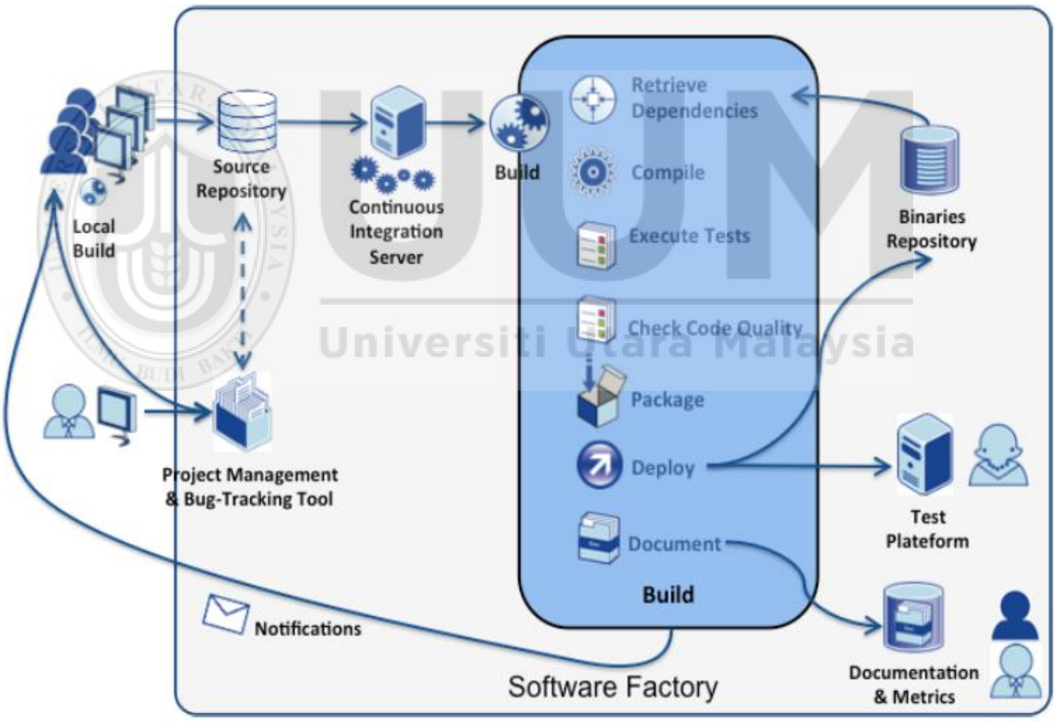



*Figure 2.6* Software Factory Workflow Model (Canuel & Robert, 2012)

In a research journal, Nomura, Tonini, Hikage, and Tonini (2007) argue that the software factory architecture has been adopted by a growing number of software houses, which facilitates outsourcing through segmentation of activities to enable a more flexible and dynamic production system. The authors propose a Model for

Defining Software Factory Processes to resolve one of the main issues in mapping the correct process within a Software Factory environment. To achieve this, the authors apply the model defined in the field by means of an action research with regards to a system improvement project. The outcome clearly defines the view of the process, which allows the identification and improvement of critical points as well as aiding the management of service demand, people, processes, and outsourcing activities. Figure 2.7 is the representation of the Software Factory map processing discussed by the authors.

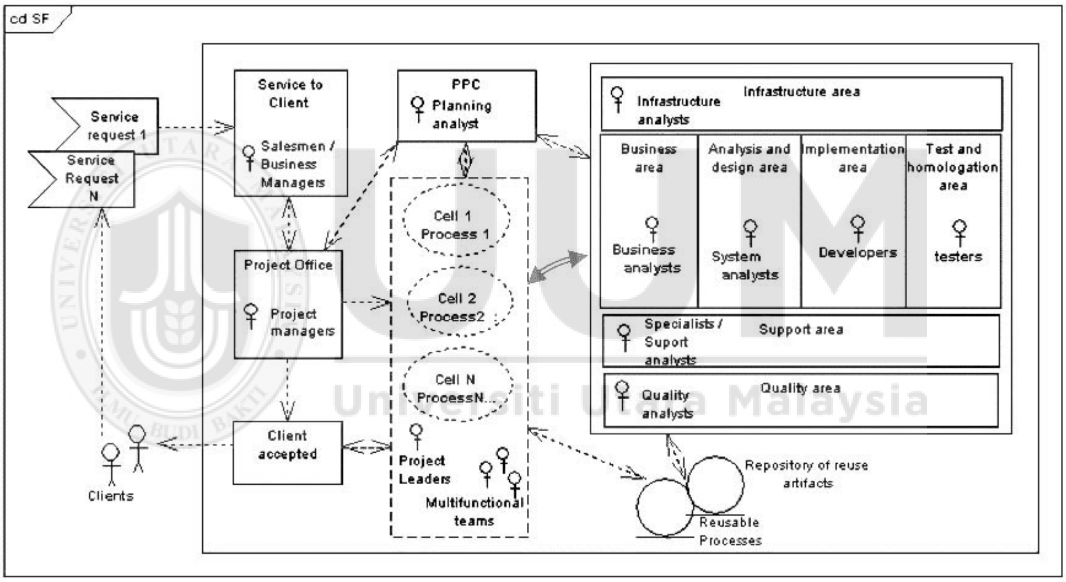



*Figure 2.7* Software Factory Process Modelling (Nomura et al., 2007)

### 2.9.2 Education Purpose

In the second group of research dealing with software factory, the adoption rate is growing in numbers for educational purposes. Ahmad, Liukkunen, and Markkula, (2014) states in their research that having the knowledge and skills to handle real life practical situations are the traits demanded by the industry for university graduates. This is particularly more important in the field of software engineering and to this effect, the University of Oulu Finland has built a Software Factory Laboratory (SWF).

The authors conducted a study, which examines the factors in the SWF learning environment that affect learning of a SWF course by the students. Based on the study, the authors posed the findings in which all the factors studied play an important role in software engineering students learning, academic achievements and professional skills gaining as Figure 2.8 shows the competencies gained by the students in the SWF project course. With the findings made from the study, the authors believe that it would develop a better understanding on learning environments, which ultimately can be used for the improvement of software engineering learning environment as well as to develop a better understanding of the different landscape of learning environments.

| **Competen-cies gained** | **Percentage** | | | | | **M** | **Mdn** |
|---|---|---|---|---|---|---|---|
| | *Strongly disagree* | *Dis-agree* | *Neut ral* | *Agree* | *Strongly Agree* | | |
| Effective task management | 5 | 10 | 10 | 55 | 20 | 3.68 | 4.00 |
| Solving complex problems | 5 | 5 | 25 | 50 | 15 | 3.58 | 4.00 |
| Sharing responsibilities | 5 | 5 | 25 | 35 | 30 | 3.74 | 4.00 |
| Developing a shared | 5 | 5 | 25 | 40 | 25 | 3.68 | 4.00 |
| Building a positive relationships | 10 | 0 | 15 | 40 | 35 | 3.84 | 4.00 |
| Negotiating with other groups | 5 | 5 | 10 | 50 | 30 | 3.89 | 4.00 |
| Use of rational argument to persuade others | 15 | 5 | 25 | 35 | 20 | 3.32 | 4.00 |
| Resolving conflict | 15 | 5 | 40 | 15 | 25 | 3.21 | 4.00 |

*Figure 2.8* Competencies Gained in SWF Project Course (Ahmad, Liukkunen, and Markkula, 2014)

Siqueira et al., (2008) published a proposal, which looks at how Software Factory is applied in a software engineering laboratory. The authors affirm that productivity, quality, scale, and control within the software development environment is one of the aims of a software factory. By applying the concept of a software factory with well-defined processes as shown in Figure 2.9 for software development and management the authors, believe that it allows learning in practice of project management activities based on standardized quality. To further stress the significance of defined and controlled processes, Siquera et.al. claims students find it easier to understand the activities to be done and shows students examples of the many problems commonly faced in enterprises.

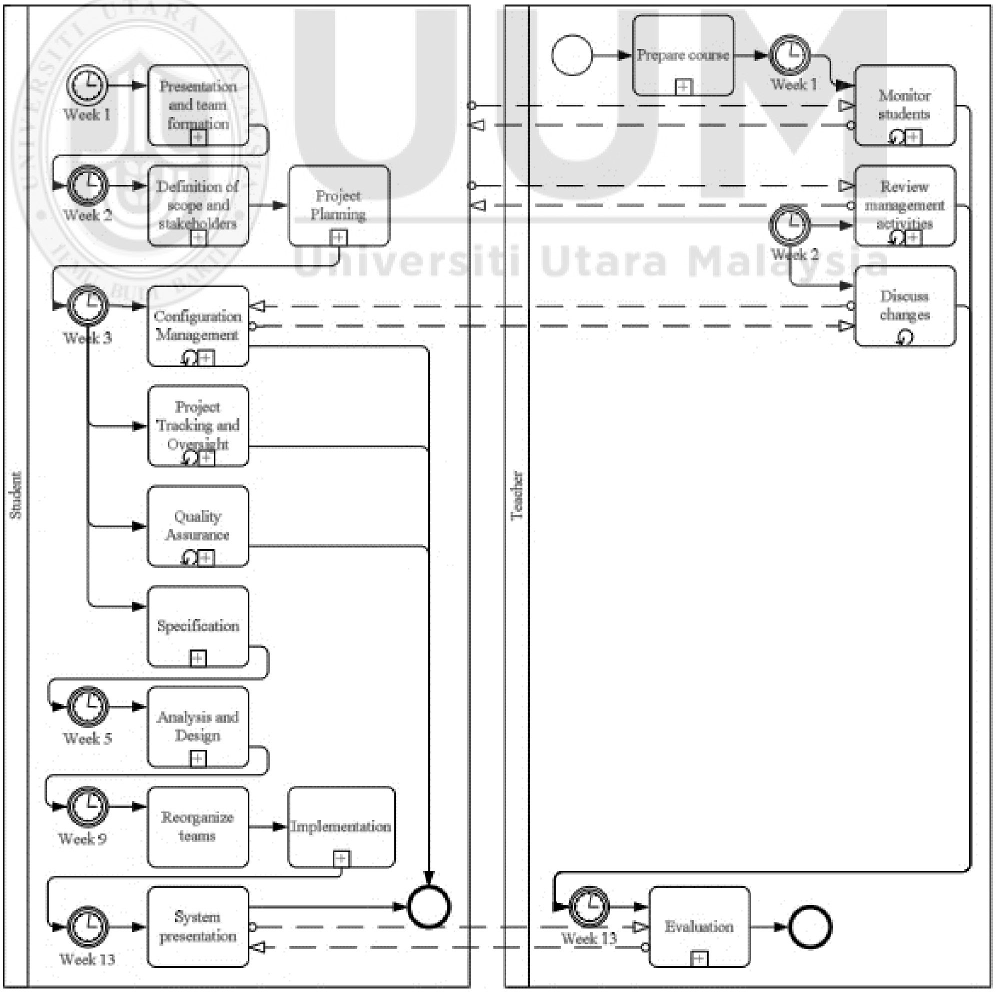



*Figure 2.9* Overview of the Educational Factory Processes (Siqueira et al., 2008)

### 2.9.3 Management Information System (MIS) Solution

As Software Factory concept is gaining acceptance, it is noted that several adoptions of Software Factory approaches can be grouped to service as a Management Information System (MIS) solution.

To overcome the complexity dealing with non-standard data representations used by computing systems in the aviation domain, Comitz and Pinto (2006) conducted a study to evaluate a prototype software factory which provides the capability to create data and interface models used in the air traffic domain. The authors propose a model, which will allow the user to specify various entities allowing the factoring to automatically create machine usable data representation. Comitz et.al. (2006) makes use of developing software factories research to introduce three major work areas pertaining to its research, which are creation of aviation metadata repository, creation of aviation data software factory using the metadata repository, and finally specifications of Domain Specific Language (DSL) for aviation data. By using software factory approach, the authors hope that the management of complexity in system acquisition and procurement can be reduced.

One of the most prominent research in Software Factory was conducted by Greenfield and Short, (2003a), as it looks in a methodology developed, which is referred to as Software Factories. The authors view Software Factory as a development environment configured to support the rapid development of a specific type of application and claims it would change the overall view of software industry through introducing patterns of industrialization. In further defining software factory, the authors refer Software Factory as "*A software product line that provides a production facility for the product family by configuring extensible tools using a* *software template based on a software schema*". The authors address the ability of a software development environment, which allows for mass customization and utilization of templates to support an automated software development environment. Figure 2.10 illustrates the typical Software Factory model that the authors propose.

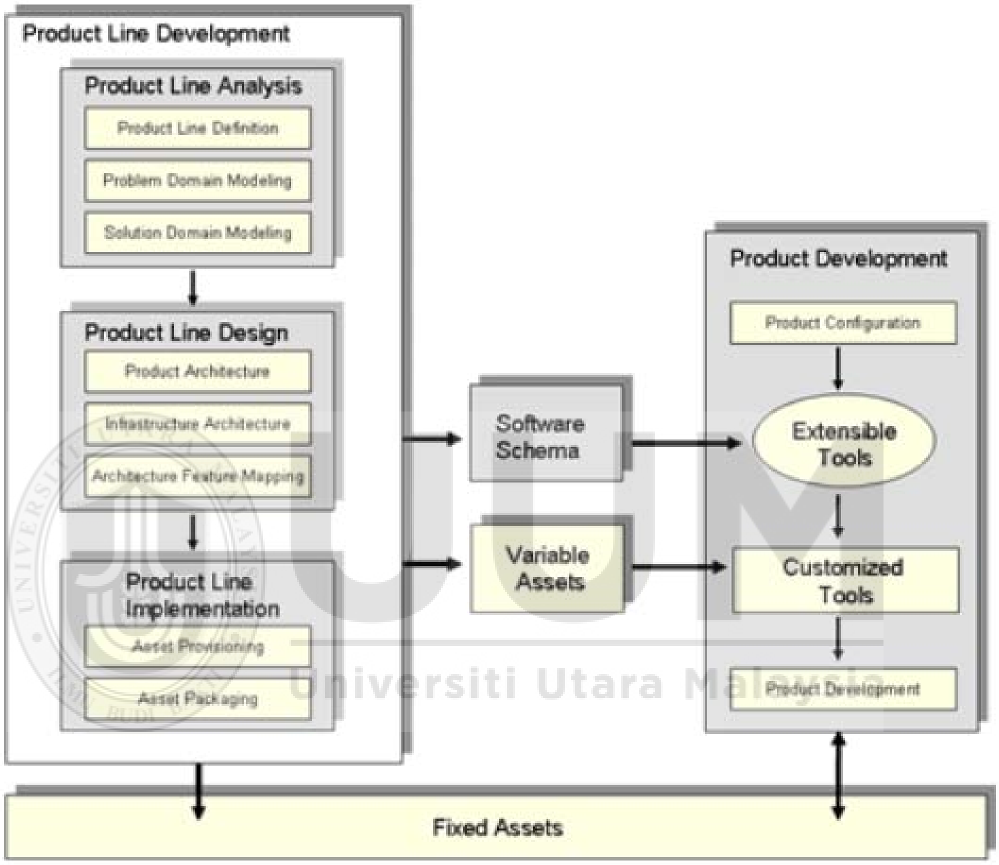



*Figure 2.10* Software Factory Model (Greenfield & Short, 2003)

### 2.9.4 Software Vendor

The most interesting concept of Software Factory is the application to the software development environment for software providers. In their research paper, Li et al., (2001) proposes a general Software Factory Model (SFM) which is presented through a static and dynamic view which ultimately is used as a base of the improvement of software product quality and productivity. Figure 2.11 illustrates the combination of static and dynamic view of the SFM.

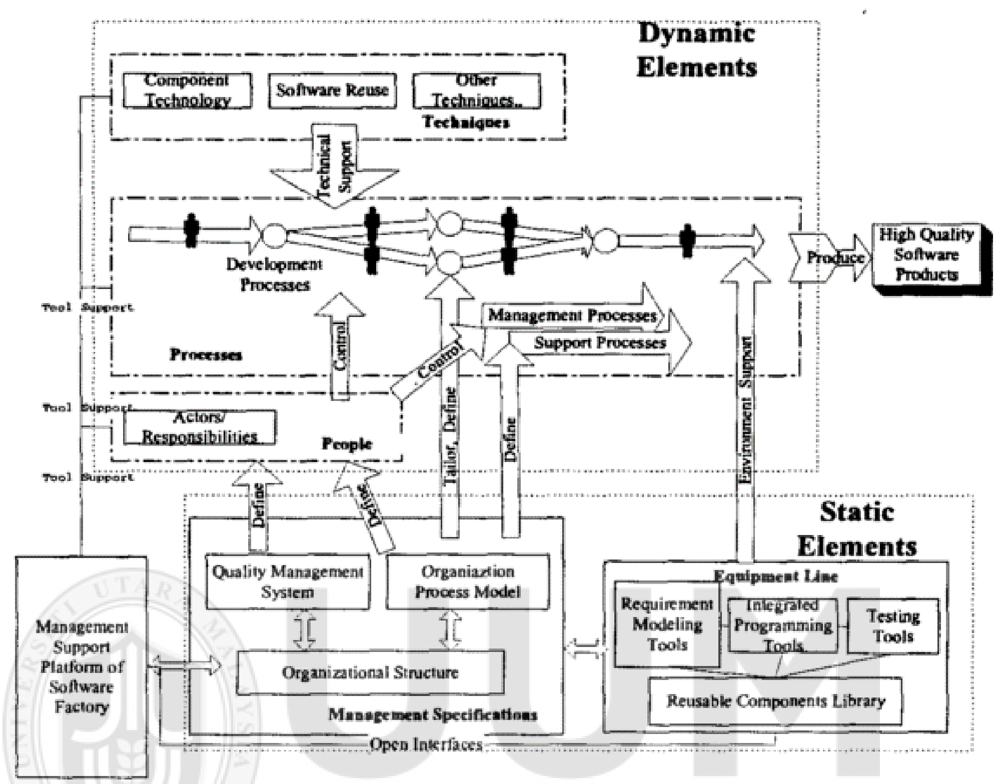



*Figure 2.11* Static and Dynamic Elements of SFM-CSO (Li et al., 2001)



Another example of Software Factory adoption for software development was presented in a journal published by Lim et al., (2000). The authors while acknowledging that high software throughput is desirable by diffusing software-based innovations, a robust, consistent, and efficient means of producing quality and reliable software is needed. The authors claim that adopting a Software Factory model would allow the ability to develop software in a production-like manner and within a factory-like environment. To achieve this, the authors propose a Software Factory model utilizing industry practices, derived theoretical framework, existing literature, and the traditional Source-Message-Channel-Receiver-Effect (S-M-C-R-E) communication model for diffusion of innovations. In this research, Lim et.al. focus on the objectives to formulate a hypothesis that a Software Factory model can be attained to diffuse software-based innovations and concurrently testing the model through the demonstration of the model. Figure 2.12 exemplifies the use of Software Factory approach by focusing on the diffusion of innovations from the providers to the adopters.

Although the research points out that attaining a high throughput using Software Factory approach is attainable, in order to achieve a production-like manner with manufacturing environment within the scope of software development process requires a more in depth look as how to manage the human capital as well.

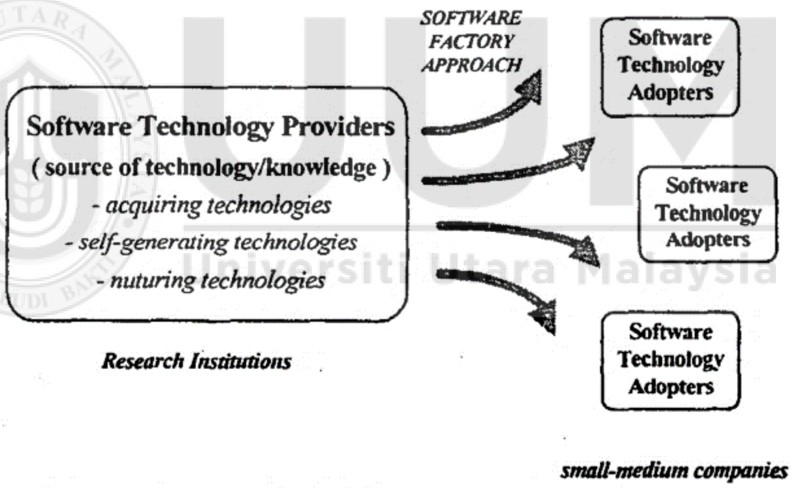



*Figure 2.12* Software Factory Approach in Diffusing Innovations (Lim et al., 2000)

### 2.10 Comparative Analysis of Existing Software Factory Models

Comparative analysis of the relevant existing model of Cloud ERP is shown in Table 2.5, which shows the comparison of existing research in regard to their focus emphasizing on selected components of Cloud ERP management.

Table 2.5

*Existing Software Factory (SF) Models and Approach*

| No | Authors | Research Title | Comments |
|---|---|---|---|
| 1 | Bratman, Harvey<br>Court, Terry, 1975 | The Software Factory | • Provides definition of Software Factory<br>• Addressing Software Development methodologies<br>• Did not cover Cloud ERP environment |
| 2 | Canuel, Vincent<br>Robert, Mikael, 2012 | Toward a better software factory | • Provides definition of Software Factory<br>• Addressing Software Development methodologies<br>• Did not cover Cloud ERP environment |
| 3 | Fernstrom, Christer<br>Narfelt, Kjell Hakan<br>Ohlsson, Lennart, 1992 | Software Factory Principles, Architecture, And Experiments | • Provides definition of Software Factory<br>• Addressing Software Development methodologies<br>• Did not cover Cloud ERP environment |
| 4 | Helton, David, 2010 | The Potential of Software Factories | • Provides definition of Software Factory<br>• Addressing Software Development methodologies<br>• Did not cover Cloud ERP environment |
| 5 | Nomura, L<br>Tonini, A C<br>Hikage, O K<br>Tonini, Antonio Carlos, 2007 | A Model for Defining Software Factory Processes | • Provides definition of Software Factory<br>• Addressing Software Development methodologies<br>• Did not cover Cloud ERP environment |
| 6 | Ahmad, Muhammad Ovais<br>Liukkunen, Kari<br>Markkula, Jouni, 2014 | Student perceptions and attitudes towards the software factory as a learning environment | • Application of SF into classroom and educational settings<br>• Did not cover Cloud ERP environment |
| 7 | Chao, Joseph<br>Randles, Mark, 2009 | Agile software factory for student service learning | • Application of SF into classroom and educational settings<br>• Did not cover Cloud ERP environment |
| 8 | Siqueira, Fábio Levy<br>Barbarán, Gabriela M Cabel<br>Becerra, Jorge Luis Risco, 2008 | A software factory for education in software engineering | • Application of SF into classroom and educational settings<br>• Did not cover Cloud ERP environment |

Table 2.5 continued

| | | | |
|---|---|---|---|
| 9 | Mayra Pariata, Nora Montaño, 2014 | Software Factory: from professional environment to academic environment | • Application of SF into classroom and educational settings<br>• Did not cover Cloud ERP environment |
| 10 | Comitz, Paul H. Pinto, Avinash, 2006 | A software factory for air traffic data | • Application of SF into classroom and educational settings<br>• Did not cover Cloud ERP environment |
| 11 | Piho, G<br>Tepandi, J<br>Roost, M<br>Parman, M<br>Puusep, V, 2011 | From Archetypes Based Domain Model Via Requirements to Software: Exemplified by LIMS Software Factory | • Adaptation of Software Factory in corporate environment<br>• Management Information System view<br>• Internal development model<br>• Did not cover Cloud ERP environment |
| 12 | Siy, Harvey P<br>Herbsleb, James D<br>Mockus, Audris<br>Tucker, George T<br>Krishnan, Mayuram, 2001 | Making the Software Factory Work: Lessons from a Decade of Experience | • Adaptation of Software Factory in corporate environment<br>• Management Information System view<br>• Internal development model<br>• Did not cover Cloud ERP environment |
| 13 | Sharilyn A. Thoreson, 1989 | The Automated Software Development Project at McDonnell Aircraft Company (The Software Factory) | • Adaptation of Software Factory in corporate environment<br>• Management Information System view<br>• Internal development model<br>• Did not cover Cloud ERP environment |
| 14 | Greenfield, Jack<br>Short, Keith, 2003 | Software Factories: Assembling Applications with Patterns, Models, Frameworks and Tools | • Adaptation of Software Factory in corporate environment<br>• Management Information System view<br>• Internal development model<br>• Did not cover Cloud ERP environment |
| 15 | Li, Chao<br>Li, Huaizhang<br>Li, Mingshu, 2001 | A Software Factory Model based on ISO9000 and CMM for Chinese small organizations | • Adaptation of Software Factory in corporate environment<br>• Management Information System view<br>• Internal development model<br>• Did not cover Cloud ERP environment |
| 16 | Lim, N. K.<br>Ang, James S K<br>Parvi, F. N., 2000 | Diffusing software-based innovation with a software factory approach for software development | • Adaptation of Software Factory in software production for software vendors<br>• Did not cover Cloud ERP environment |

Table 2.5 continued

| | | | |
|---|---|---|---|
| 17 | Rockwell, Robert<br><br>Gera, Michael H, 1993 | The Eureka Software Factory CoRe: A Conceptual Reference Model for Software Factories | • Adaptation of Software Factory in software production for software vendors<br>• Mass customization approach<br>• Cover outsourcing model<br>• Did not cover Cloud ERP environment |
| 18 | Navarro, J J, 1993 | Characteristics of a Flexible Software Factory - Organization Design Applied to Software Reuse | • Adaptation of Software Factory in software production for software vendors<br>• Mass customization approach<br>• Cover outsourcing model<br>• Did not cover Cloud ERP environment |

In summary, the comparative studies of the existing relevant research provide an overview of the existing research coverage and approaches to the Cloud ERP challenges. However, most of the articles only focus on a specific area of Software Factory approach and rarely provide a Software Factory model that mimics a manufacturing environment. Particularly, demonstration of the Software Factory tool or machinery is lacking. Most of the research describes Software Factory approach as a general concept that possibly able to industrialized software development. As claimed by Helton (2010), Software Factory demonstration is still inadequate and not available to be acquired by the industry players.

Although all the researchers listed in Table 2.5 may have variations in their Software Factory concepts and models, their basic principles are similar in that they are mainly targeting a way to provide a systematic approach of developing software by reducing human craftsmanship effects as well as promoting reusability. However, their solutions were mostly intended for a traditional software production model, and not a Cloud SaaS model or an ERP model. Moreover, most of the research did not provide a production model that can be simulated to any Cloud ERP production environment in solving development workload, complexity, consistence quality, and knowledge retention management problems. In addition, a specific Software Factory approach focusing on Cloud ERP production environment is also inconclusive.

For this reason, this research would be able to bridge the gaps by providing a more detailed and thorough demonstration of Software Factory though the development of our model includes the incorporation of Software Factory with other theories and concepts which are discussed in length in the next section.

## 2.11 Applicable Theories and Concepts

In this section, all the related theories and concepts applicable to this research are reviewed.

### 2.11.1 Resource Base Theory

Resource based theory or Resource base view is a concept that emphasize about organization's tangible and intangible resources that can be used to formulate a strategy for competitive advantage (Wernerfelt, 1984). The ability to decide the value of the available resource within the organization can provide a company direction for business venture. Wade and Hulland (2004) explain that resources are valuable and rare that their benefits can be utilized with a temporary competitive advantage. That advantage can be sustained to the extent that the firm is able to protect against resource depletion limitation, transfer, or substitution.

Resource based theory is useful for this study as it can be used to be integrated into the model for competitive advantage of the Cloud ERP provider. This can be achieved by enabling skilled manpower to handle multiple and/or all projects compared to just one project.

### 2.11.2 Mass Customization Theory

Mass Customization is a theory published in 1989 by Stan Davis in Future Perfect (Davis, 1989) and developed by Pine in 1993 (Pine, 1993). The theory provides a paradigm shift for businesses to offer variants of products and services to meet individual personalized needs, while keeping the near mass production efficiency (Tseng, Hu, & Wang, 2013). Among the core strategies of the theory is common product family platform and flexible manufacturing which in turn create economic of scale advantage; which has been widely implemented today and has become a marketing strategy for competitiveness (Jiao & Tseng, 2004). Although widely implemented in other industries, Mass Customization concepts in the software industry has not been well received, which rings true for ERP industry as well.



### 2.11.3 Work System Theory

According to Alter (2013), a work system is a system in which human participants and/or machines perform processes and activities using information technology and other resources to produce products/services for internal or external customers. In most organizations, there should be multiple work systems in practice, such as material procurement, product delivery, employee hiring, and invoicing customers, etc. Special casework systems include information systems whose activities are devoted to processing information.

Projects are also work systems designed to produce specific products/services and then go out of existence. Work systems with human participants are called sociotechnical work systems in contrast with totally automated work systems that operate autonomously and automatically when launched.

Work system theory consists of three components those are:

i. The definition of the term work system, in which human participants work to produce a specific product of service for an internal or external customer.
ii. Work system framework, a static view of a work system as it exists during a time interval when it retains its identity and integrity even though it may change slightly through small adaptations, workarounds, personnel changes, and even unintentional drift.
iii. Work System Life Cycle (WSLC) model, a dynamic view of how a work system changes over time.
iv. Work system lifecycle model, a dynamic view of how work systems change over time through a combination of planned and unplanned change (Alter & Ph, 2011).

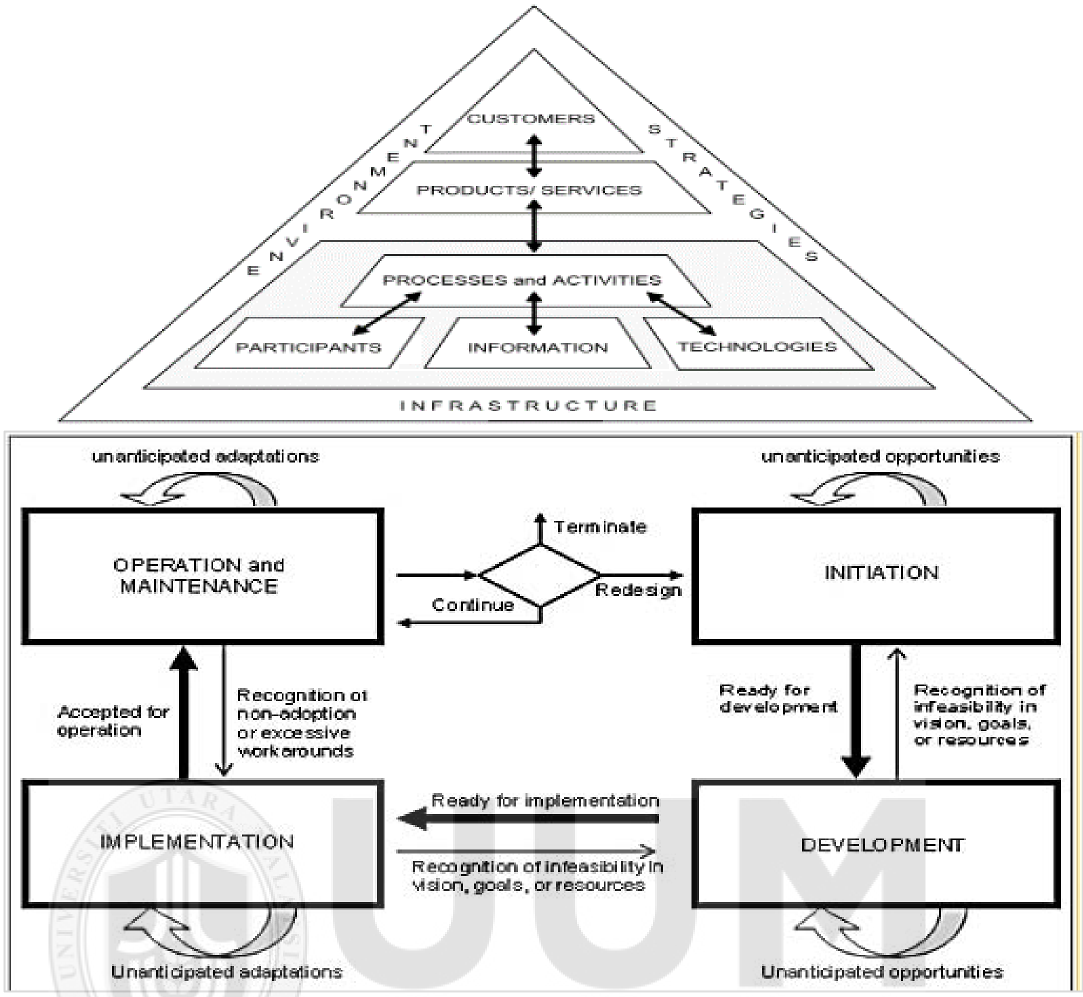



*Figure 2.13* Work System Framework and Work System Lifecycle.

From general perspective, Work System Theory describes work procedure in an organization. Figure 2.13 illustrates the Work System Framework and its Lifecycle (Alter, 2013).

In formulating our CEF Model, Work System Theory is utilized to represent the model and submodel. Sociotechnical Work System Theory will be particularly of interest, as the CEF Model is mainly about a human interaction machine likened to a software production facility.

### 2.11.4 Sustainability Theory

Sustainability theory is focused on the awareness of an organization with regards to its ability to maintain viable in the business world (P. a. C. Smith, 2012). Sustainability has been a very prominent topic discussion, gaining momentum in recent years as claimed by Müller and Pfleger (2014). It is believed that a business organization should be thinking about business sustainability in order to be competitive in the market. There have also been research that argues that there exist a small yet powerful relationship between sustainability and company financial performance (Peloza & Yachnin, 2008). However, the main argument is that for a business to be sustainable, an organization business model must be transformed by social and environmental priorities, apart from the classical bottom line of optimizing stakeholder's equity.

It is relevant to acknowledge the fact that this concept was included in the research for the CEF Model to point out that part of the main reason for this model is to allow for business sustainability in terms of providing a long lasting, sustainable system.

### 2.11.5 Software Product Line

Software Product Line Engineering or commonly known as SPLE or SPL is a concept, which promotes the reusability of common features of a software system that satisfy a specific market segment needs (Northrop, 2008). This concept is very much a predominant research area conducted in the Software Engineering Institute at Carnegie-Mellon (Northrop, 2008). In a research that focuses on the variability issues, Bosch (2002) claims that a standard software product line consists of three major elements which are product line architecture, software components and the product family. This is very much a standard that most software developers adhere to with regards to software product development (Gomaa & Saleh, 2006; Matinlassi, 2004). Based on this, the three components were chosen as one of basis in our CEF Model component.

### 2.11.6 Dynamic Software Product Line

Dynamic Software Product Line (DSPL) is basically an extension of SPL which supports runtime variability (de Jesus Souza, Magno Lua Santos & de Almeida, 2015). It is a methodology which is heavily used to develop software capable of flexible adaptations, runtime capabilities, reconfigurations, and post deployment activities (Banerjee & Kumari, 2016). Therefore, DSPL provides a very good platform to develop a product, which is highly customizable and configurable depending on the runtime of system according to the changes within the environment itself. With regards to our research, DSPL was used to enhance the workflow system to be self-autonomous and self-managing to suit the dynamic changes of a software system.



### 2.11.7 Software Configuration Management

Software Configuration Management (SCM) is defined primarily as the mechanism required to control the evolution of software products (Estublier, 2000). In his research, Estublier (2000) states that a typical SCM manages several areas such as repository of components, process control and support as well as workspace control. A part of SCM is version control, which is defined as the technique of managing the dynamic changes made to a software source code and/or objects (Junqueira, Bittar, & Fortes, 2008). In software development, it is accepted that a version is always followed by its revision number therefore it can be concluded version and revision control are always paired together to actually make sense. Release engineering or release management is a part of Software Engineering process, defined as the release of the software artifact whether it would be in the form of executable, installer, library, or source code package (Wright, 2009). Each segment of Software Configuration Management is of interest in our research as it has provided the basic elements for our Product Control component within the CEF Model.

### 2.11.8 Knowledge Management

Knowledge Management consist of several elements such as finding, organizing, selecting, disseminating and transferring of information that can be used for problem solving, dynamic learning, strategic planning and decision making (B. Gupta, S.Iyer, & Aronson, 2000). This concept is crucial in many knowledge-based organizations due to the heavy dependency on employees for both tacit and explicit knowledge. This concept was included as one of the major elements due to the sheer nature of its importance in dealing with knowledge retention management in a knowledge-based working environment.

### 2.11.9 Continuous Quality Improvement

Continuous Quality Improvement (CQI) is a management principle in product and services quality department. In Total Quality Management (TQM), similar emphasize on continuous process improvement is part of the key element (Decker, 1992). From CQI and TQM perspectives, in order to achieve or improve product or service quality, the management has to embark in the continuous business process improvement with the objective to achieve quality target (Mohd Yusof & Aspinwall, 2000).

The TQM and CQI concepts reflect that organization will need to continuously review and reengineer its business processes in order to remain competitive. CQI concept is relevant to this topic in such a way that, by continuously improving the quality directly implies continuous review and improvement of the affected business processes. In return, the business process improvement may directly be translated to ERP requirement change thus triggering the need to update the ERP software periodically.

In other words, ERP Customization itself is not a one-time implementation affair, yet it is a process of adapting software module to an ever changing business processes (Skok, Legge, & Hill, 2002). To address these challenges, an ERP model is needed that not only supports customization, but also possesses the flexibility of continuous updates in order to meet the dynamic addition change of customer requirements. In adapting Mass Customization into ERP industry, personalized ERP product are defined to come with the flexibility of continuous software updates to fit dynamic requirement.



**2.11.10 Model Development**

Model is a composition of concepts, which can be used to explain abstractly and help other to simulate the represented model. The process of constructing a model is referred to as conceptual modeling and the output of the process is called artifact (Kotiadis, 2008) .

Robinson (2010) suggests that a good model should be:

i. Valid - produce accurate result.

ii. Credible – believed by the client.

iii. Feasible – can be simulated with the current technology.

iv. Useful – sufficiently easy to use, flexible, visual, and quick to run.

Conceptual model is different from computer model as it is not software specific. A computer model is software specific and it represents a model. Figure 2.14 further explain the separation of the two models and also shows the artifacts of modeling.

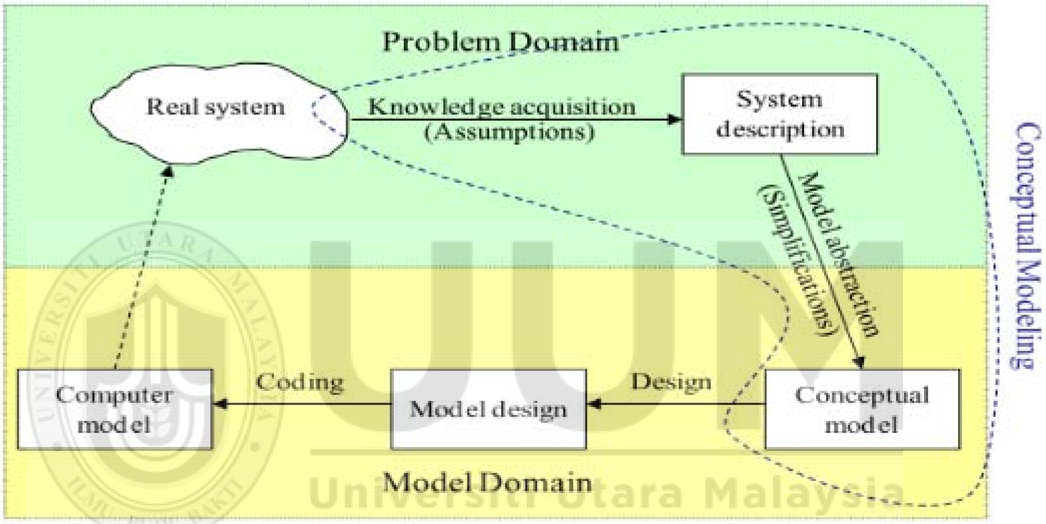



*Figure 2.14* Artifacts of Conceptual Modeling (Robinson, 2010)

Kung and Sölvberg (1986) explains that a conceptual model's main objective is to convey the fundamental principles and basic functionality of the system, in which it represents and it must be easily understood by intended users. A model, when implemented properly, should satisfy the following four fundamental objectives:

i. Enhance an individual’s understanding of the representative system.

ii. Facilitate efficient conveyance of system details between stakeholders.

iii. Provide a point of reference for system designers to extract system specifications.

iv. Document the system for future reference and provide a means for collaboration.

The quality of a model is another method of verifying the proposed model. Several dimensions can be used to gauge the quality of a model such as Understandability, Clear Steps, Relevancy, Flexibility, Scalability, Accuracy, Timeless, Completeness, and Consistency as per presented by Abu Bakar (2012).

All of the authors mentioned above have elaborated the model definition, purpose and also it's important criteria.  The concepts were adopted into this research in developing the model Cloud ERP Factory.

**2.12 Review of Research Gap Analysis and Approach Rationale**

In order to solve the research problem components, review of previous research within the problem domain have been carried out and comparative analysis made have grouped them into four major problems those are, complexity management, high workload, inconsistent software quality and knowledge retention management. However, it is noticeable that the researches were intended to solve its own problem component and missing a research that intended to solve all four research problem components holistically. Notably, in order to tackle technical component of Cloud ERP production model, all four research problems must be addressed within a single development model thus, establishing a research gap that needs to be addressed. Another research gap is that most of the research presented addresses Cloud SaaS rather than Cloud ERP which is actually a group of modular and integrated SaaS solutions. Software Factory was identified as the most suitable software engineering approach to develop the intended solution due to its previous success in research undertaken to industrialize software development. Comparative analysis of previous Software Factory research however has shown a gap notably that is lacking in research in Cloud environment and also Cloud ERP production environment. This research can ultimately be a successful demonstration of Software Factory adaptation in software development as Helton (2010) has mentioned the demonstration for Software Factory is still inadequate.

Concepts and theories elaborated in the previous section have been identified as important element in researching for the intended Cloud ERP production model. Finally, another gap worth mentioning is the fact that from the literature review, there is an obvious lacking of research undertaken combining all those theories and concepts into producing a development model from developer's point of view.



**2.13 Summary**

In summary, the review of literatures in this chapter constitutes the knowledge required for this research. The literature topic presented follows the topic visualization structure shown at the beginning of the chapter in Figure 2.1. This chapter has explained the basic knowledge of ERP, Cloud Computing, Cloud Services, Cloud ERP, Software Factory model, and comparative analysis of previous relevant research. The research gaps from the relevant research were also presented to substantiate that the solution provided in this research could fill in the gaps. Subsequently, it presents the related theories and concepts as well as the brief explanation about modeling. The literature review also formulated the need for a new model for Cloud ERP Factory and presented the Software Factory as a technically feasible approach in delivering the right model. Finally, it also explains the modeling approach taken from the literature that will be taken as guidelines in constructing a good model.

# CHAPTER THREE
# METHODOLOGY

## 3.1 Overview

Chapter Three illustrates the overall research process and approach for this study. In general, the research employed design science or design research with mixed methodology quantitative and qualitative for evaluation phase. The rationale of using design research and mixed methodological approach was reasoned out in the following section. Each phase of the methodology was coherently elaborated with the matching activity phases in order to accomplish each objective of the study, respectively.

## 3.2 Design Research

The design research approach was chosen for the purpose of addressing the issues described in Chapter One. The decision to use the design research methodology was based on the fact that this approach is the most suitable approach to produce the intended outcome of our research. Design research is defined as a process which is able to create effective artifacts by producing and applying scientific knowledge of the specific tasks (March & Smith, 1995). Design research is also preferred because it is able to solve problem and improves the performance at the same time. There are several approaches to design research, with many proposing different set of processes of constructing design science research. In our research however, the design guidelines as proposed by Hevner, March, and Park (2004) that assists researchers in understanding the design science research requirements are preferred. Table 3.1 is a summary of the proposed guidelines.

Table 3.1

*Design Science Guidelines (Hevner et al., 2004)*

| **Guidelines** | **Descriptions** |
|---|---|
| Guideline 1: Design as an Artifact | • Design-science research must produce a viable artifact in the form of<br>• a construct, a model, a method, or an instantiation. |
| Guideline 2: Problem Relevance | • The objective of design-science research is to develop technology-based solutions to important and relevant business problems. |
| Guideline 3: Design Evaluation | • The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods. |
| Guideline 4: Research Contributions | • Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies. |
| Guideline 5: Research Rigor | • Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact. |
| Guideline 6: Design as a Search Process | • The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment. |
| Guideline 7: Communication of Research | • Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences. |



## 3.3 Rationale of Using Design Science Approach

As the artifact that this research hopes to achieve is a model, the processes involved in the model development is mainly a design work and the evaluation of its validity. These activities match with the design science research approach. Therefore, design science research methodology is very much aligned with our research objectives. This approach provides dynamic processes such as development and evaluation strategy that will be conducted in this study. This approach also focuses on delivering effective artifact by using a systematic research.

## 3.4 Rationale of Using Mixed Methods Approaches for Evaluation

In Mixed Methods approaches for evaluation purposes, both Qualitative and Quantitative research methods were used. By their name, both refer to the manner through which the entire research process is carried out. The reasoning for choosing which method depends on the nature of the research problem and research objectives. Qualitative method refers to the unstructured primarily exploratory methodology, wherein the research activities was based on small sample but with broader in-depth understanding of the research area while quantitative research refers to a research methodology, which is mainly designed to quantify the data collected by applying some forms of statistical analysis (Offermann & Platz, 2009).

Due to the huge scope nature of this research and also the objective to validate the model, both Qualitative and Quantitative data are being gathered to support the research findings. As the typical lengthy deployment time of Cloud ERP, the empirical quantitative data was limited to only three case studies. Qualitative data from interviews was also used to formulate the conclusion.

## 3.5 Research Phases

Design science methodology is decided to be the most suitable approach for our research as it is focused on the development of the model artifact (Offermann & Platz, 2009; Vaishnavi & Kuechler, 2004). Following Design Research Science (DSR), this study was divided into five phases, which are (i) awareness of the problem, (ii) suggestion, (iii) development, (iv) evaluation, and (v) conclusion. The overall research activities were outlined illustratively in Figure 3.1.

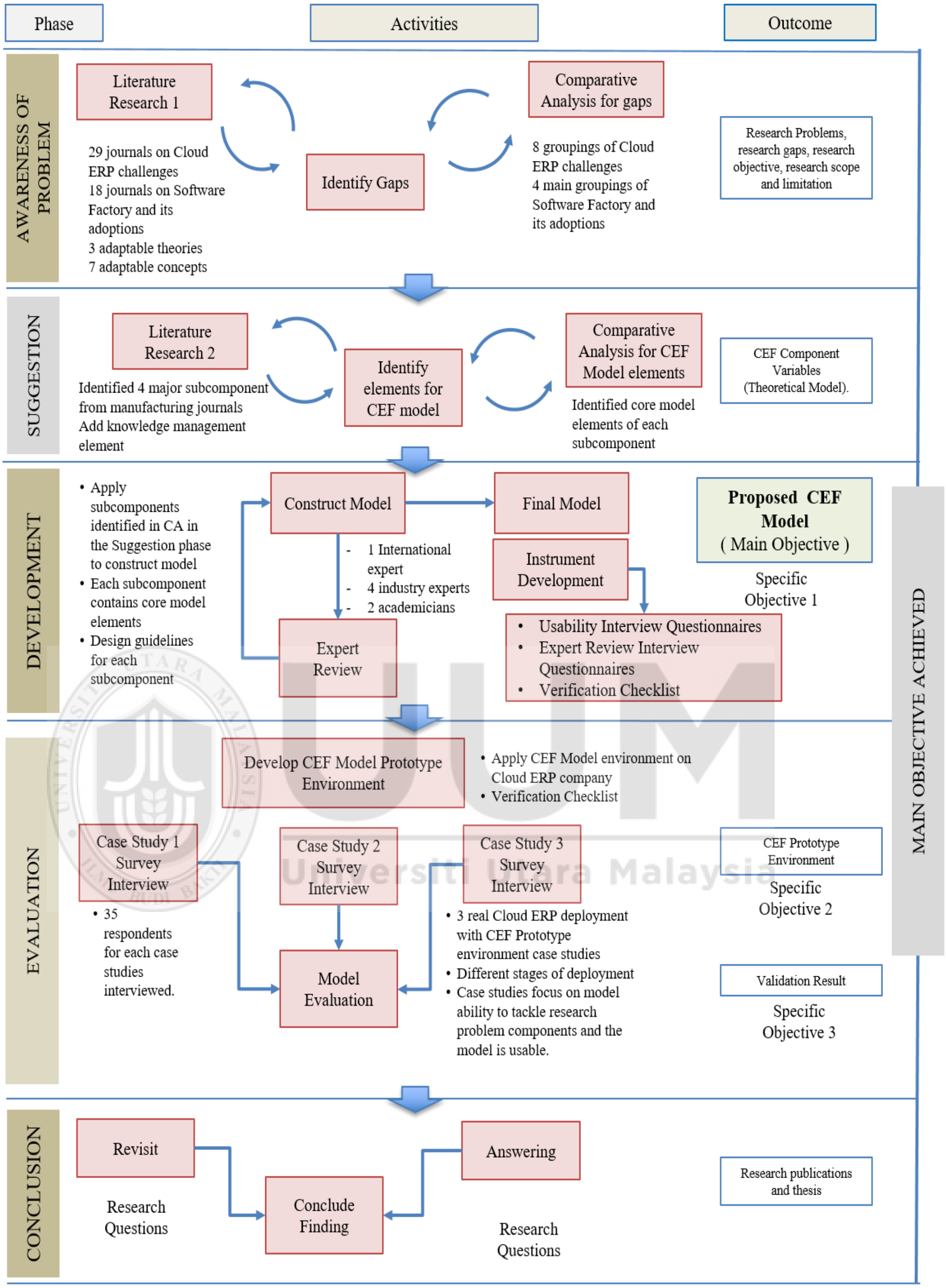




*Figure 3.1* Summary of Research Phases.

### 3.5.1 Phase 1: Awareness of the Problem

This research has three main phases, which start with theoretical study phase. In this phase, a literature review about subject matters, concepts, and theories were performed to identify problem statements, research gaps, research questions, and research objectives. The preliminary study was conducted to reconfirm the research problem and its significance. Finally, the information gathered was used to formulate Theoretical research foundation. Elaboration of the processes involved is discussed in the following sections.

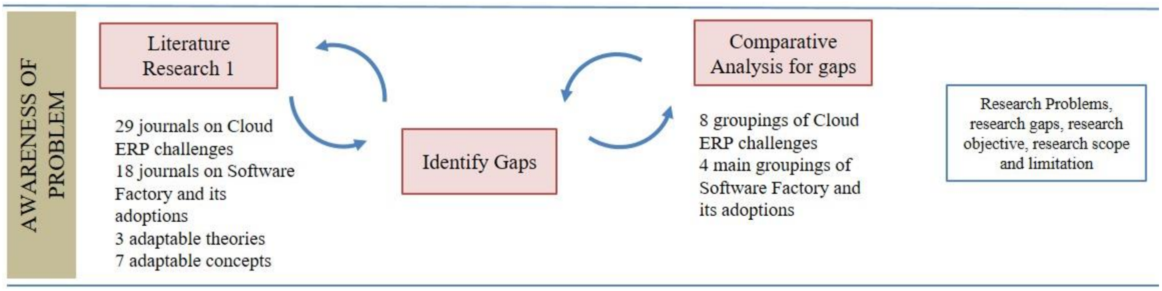



*Figure 3.2* Awareness of Problem Phase Activities

In this phase, problem identification was formulated through extensive and intensive literature reviews, which followed by comparative analysis of the chosen relevant models. Finally, preliminary investigation was performed to reconfirm the formulated research problem. Figure 3.2 explains the flow of the processes and the detail is elaborated in the subsection.

i. Literature Study and Content Analysis

Literature Review was done with the sources gathered from journals, proceedings, books, and whitepaper from professional website. In this study, the main aim of the content analysis was to gather the concept and theories in proposing a model of CEF.

ii. Comparative Analysis

Eighteen of the most relevant existing Software Factory Models and its adoption were identified for Comparative Analysis study. The relevant Cloud ERP existing models were reviewed and compared in order come out with model component variables. The results of comparative analysis as discussed in Chapter Two were compiled and utilized as the input in determining the main elements of the model of CEF.

### 3.5.2 Phase 2: Suggestion

In the second phase, the outcomes from the literature review, content analysis, and comparative study were studied and analyzed in order to derive the component variables of the CEF model. Activities involved in this phase are illustrated in Figure 3.3.

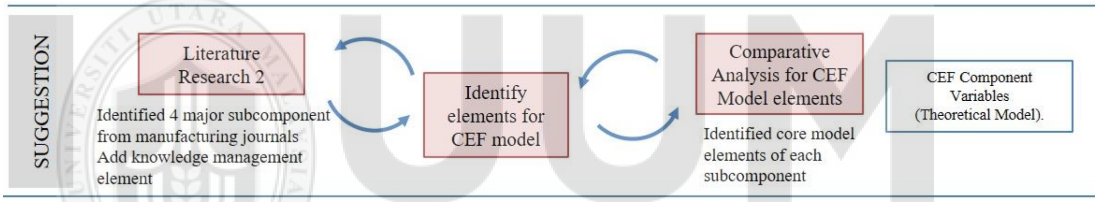



*Figure 3.3* Suggestion Phase Activities



The main purpose of this activity was to determine component variables of CEF, which was also research Specific Objective 1. It was then formulated to make up the intended model. At this stage, all components gathered that are related to Cloud ERP model were compiled and analyzed. The components were then reviewed in order to filter out the insignificant components, then the independent and dependent component variables were selected. The output of this stage is the Theoretical Model that can be used to formulate the intended model in the next phase.

### 3.5.3 Phase 3: Development

Development stage was the third main phase proceeding the suggestion phase. Model Construction was part of the development phase. The detailed steps involved were shown illustrated in Figure 3.4 and described in the subsequent sections.

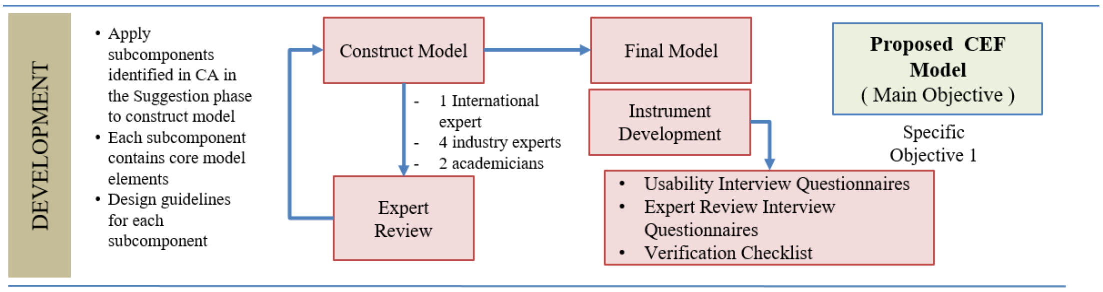



*Figure 3.4* Development Phase Activities

In this phase, the outcome which are CEF components, was used to develop the model as shown in Figure 3.4. The CEF Model was modularly developed. These submodels of CEF were then integrated to construct the overall CEF Model. The presentation concepts such as abstraction layers, workflow, and context diagram were applied in order to make up the abstraction of the model presentation. In this phase, the instrument for the face validity of the usability interview questionnaires and expert review interview questionnaires was created. The component of CEF, was the first artifact and output delivered during development phase. The main research artifact, The CEF Model, was produced at the end of development phase. However, the final model of the CEF was finalized after Expert Review interview.

#### 3.5.3.1 Model Construction

The model was constructed through several series of comparative analysis of related research articles. The first level of comparative analysis of research was conducted to define the actual model components for the CEF model. This was achieved by comparing multiple journals on manufacturing practices and standards. Once the actual model components were defined, another phase of comparative analysis was conducted in order to define the core model elements of the components in CEF model. Finally, the core model elements and the concepts and theories with the five core model components were mapped.

#### 3.5.3.2 Expert Review

The first model validation during the development phase to be carried out was the Expert Review. Expert teams were from academic user and industry experts. The objective of the expert review was to validate the CEF Model against the research objectives and to validate the survey questions. Once validated, the model was finalized as the final model. However, if the review team pointed out there was a need for model update, the model will be update with the input from the review team and then finalize as the CEF Model.



Expert review involves reviewing and validating the proposed model and it has been recognized as an important way to improve the quality in the developed application (Wiegers, 2002). It was suggested that having three to six experts participating in reviewing process is adequate to form a good review team (Shneiderman, 1992). Therefore, this research employed seven experts, which are made up of five reviewers from industry and two from academia. All the experts chosen were required to have more than ten years' experience in ERP development field.

The procedure of reviewing process was formed in the following technique: (i) prepare the review form by listing selected assessment attributes, (ii) conducting the review, and (iii) analyze the findings. In particular, face-to-face, emailing services, and telephone calls will be utilized as the communication mediums. There were two types of instruments used for the expert review, which were interview and questionnaires.

The following steps were taken for each expert reviewer before receiving the feedback from the expert with an average time spent of one and a half hour.

i. Explaining the research background, problem statements and research objectives and scope. (Ten minutes)
ii. Discussion about the Cloud ERP production and its problems. (Ten minutes)
iii. Explanation about Software Factory concept and its adaptations. (Ten minutes)
iv. Explanation of the proposed solution. (Ten minutes)
v. Present the model to the expert with simple elaboration. (Fifteen minutes)
vi. Interviewed the expert with the questionnaire. (Ten minutes)
vii. Request the expert to provide qualitative feedback. (Twenty minutes to one hour)

For the questionnaire section, a seven-point agreement Likert scale was utilized in order to measure their feedback.

**3.5.3.3 Instrument Development**

Two instruments were developed in order to evaluate the expert review survey questions and field usability case studies respectively. For all evaluations, the instruments were of Mixed Method approach with qualitative and quantitative evaluation.

#### 3.5.3.3.1 Instrument Development for Expert Review

In the Expert Review evaluation, mixed methods with both quantitative and qualitative evaluation were used in gathering constructive feedback from the experts. In order to assess the quality of the proposed model, Abu Bakar (2012) has described that there were nine model quality dimensions that should be considered for the measurement of the model. Table 3.2 shows the list of nine model quality dimensions that have been formulated for our expert review survey questions. Based on the same method and the quality dimensions, 9 Likert Scale questionnaires with 7 rating options covering all quality dimensions have been constructed to be the quantitative instrument, as per attached in Appendix A.

Table 3.2

*Nine Model Quality Dimensions for Expert Review Instrument*

| Model Quality Dimensions | Description | Research |
|---|---|---|
| Understandability | The model is easy to understand. (Understandability) | (Pipino, Lee, & Wang, 2002) |
| Clear Steps | The model provided a clear steps and procedures to follow. (Clear steps) | (B. Boehm, 1991; Sommerville, 2004) |
| Relevancy | The model is relevant to the software production environment of Cloud ERP. (Relevancy) | (Vassiliadis, Bouzeghoub, & Quix, 1999) |
| Flexibility | The model is able to support the needs for future Cloud ERP production environment. (Flexibility) | (Moody & Kortink, 2000) |
| Scalability | The model is able to be implemented based on the needs of an organization without having to add additional resources. (Scalability) | (Jarke, List, & Köller, 2000; Vassiliadis et al., 1999) |
| Accuracy | The data flow of the model is reliable and accurate. (Accuracy) | (Ravn & Høedholt, 2008; Reingruber & W. Gregory, 1994; Wang & Strong, 1996) |
| Timeless | The model is able to be applied in the future and in different context. (Timeless) | (Ballou & Pazer, 2003; Pipino et al., 2002; Redman, 1998) |
| Completeness | The model describes a complete Cloud ERP production lifecycle which can be applied by other Cloud ERP providers. (Completeness) | (Bovee, M., Srivastava, R., and Mak, 2001; Pipino et al., 2002) |
| Consistency | The model is able to act as a consistent information source to allow for decision making support. (Consistency) | (Batini, Cappiello, Francalanci, & Maurino, 2009; Jarke & Vassiliou, 1997) |

In addition, qualitative feedback is requested at the end of the interview session to gather constructive data that can be used to further improve the CEF model. The details of the instrument developed and the result were further discussed in Chapter Four (refer to section. 4.5). In the expert review questionnaire, several attributes have been encompassed, which include Name, Age, Position, Working Experience as well as comment section.

#### 3.5.3.3.2 Instrument Development for Model Validation

In the Model Validation, quantitative and qualitative validation was used for the data analysis summary from three different case studies. The instruments were adopted from Perceived Usefulness and Ease of Use (PUEU  Davis, 1989) utilized by IBM (Tullis & Albert, 2008) and the USE Questionnaire (USE Lund, 2008), which was used by SAPIENT and WEBLEI (Tullis & Albert, 2008). All of the questionnaires constructed from the instruments are in the form of a Likert Scale with seven rating options. The seven rating options was used based on the same rating utilized in the original PUEU questionnaire.

**A. Proposed Operational Definition**

The items in Table 3.3 lists the proposed operational definition of the selected dimension. In addition (i) Reduce Workload, (ii) Consistent Quality, (iii) Reduce Complexity, and (iv) Knowledge Retention are named as part of the dimensions to measure usability attached with the PUEU dimensions.



Table 3.3
*Proposed Operational Definition*

| **Dimensions** | **Operational Definition** |
|---|---|
| Perceived Ease of Use | • Perceived ease of use is the degree to which a model is perceived as easy to use. The simpler the model, the easier to use.<br>• The model is clear, understandable, easily to interpret and can be implemented easily. |
| Perceived Usefulness | • The proposed model is useful for understanding the development of Cloud ERP Factory. |
| Reduce Workload | • The model is able to reduce the overall workload compared to traditional ERP production practices. |
| Consistent Quality | • The model is useful in ensuring consistent quality in Cloud ERP software relative to previous methods. |
| Reduce Complexity | • The model is able to reduce the overall complexity of managing Cloud ERP production and deployment compared to previous practices. |

Table 3.3 continued

| | |
|---|---|
| Knowledge Retention | • The model can be used to manage the knowledge retention efforts in a knowledge-based organization compared to manual methods. |

**B. Drafted Items for First Draft Survey Instrument**

Table 3.4 shows the field usability survey question selection once the dimensions were determined. It shows that the survey questionnaire consists of twelve PUEU questions and with an additional of eight questions that focuses on the model's ability to address our research's problems.

Table 3.4

*Field Usability Survey Questionnaire Selection*

| **Dimension** | **Questionnaire** |
|---|---|
| Perceived Usefulness | 1. Using the system in my job would enable me to accomplish tasks more quickly.<br>2. Using the system would improve my job performance.<br>3. Using the system in my job would increase my productivity.<br>4. Using the system would enhance my effectiveness on the job.<br>5. Using the system would make it easier to do my job.<br>6. I would find the system useful in my job. |
| Perceived Ease of Use | 1. Learning to operate the system would be easy for me.<br>2. I would find it easy to get the system to do what I want it to do.<br>3. My interaction with the system would be clear and understandable.<br>4. I would find the system to be flexible to interact with.<br>5. It would be easy for me to become skillful at using the system.<br>6. I would find the system easy to use. |
| Reduce Complexity | 1. Managing Cloud ERP solution is easier than the previous method. [A]<br>2. The model has provided me with easier management tools for my work. [B] |
| Reduce Workload | 1. The model was able to reduce my daily workload of jobs. [C]<br>2. The model was able to improve the overall management of workload in the company. [D] |
| Consistent Quality | 1. The model was able to maintain the software quality being developed compare to previous method. [E]<br>2. The model provides better step by step procedures to ensure consistency. [F] |
| Knowledge Retention | 1. I am able to retrieve essential knowledge to complete my task. [G]<br>2. The model provides me with the tools for me to update the Knowledge repository. [H] |

Note:

[A] Uflacker & Busse (2007)
[B] Tracy (2016)
[C] Dillon, Wu, & Chang (2010)
[D] W. T. Tsai, Huang, & Shao (2011)
[E] Bryan (2012)
[F] Demarco & Lister (1985)
[G] Weber et al., (2007)
[H] Gulati & Srivastava (2014)

**C. Face Validity through Expert Review**

Ten experts were consulted for the expert review as suggested by Krejcie and Morgan (1970) to sufficiently verify that the questions were valid to be used in our field usability survey or better known as face validity. The method of validating the survey question by expert review was based on a similar approach utilized by Z. Abu Bakar (2015). It was also highlighted by N. Ghauri and Grønhaug (2005) that a researcher should consult experts before using the questions in an actual survey. The ten experts consisted of five industry experts and five academicians, with two of them holding a Bachelor's Degree and the other three holding a Master's Degree; whereas the academicians were all PhD holders. The questionnaires were restructured to fit the respondents' role respective to the research problems, which were (i) Reduce Workload, (ii) Reduce Complexity, (iii) Consistent Quality, and (iv) Knowledge Retention. More details are further described, such as the demography of the respondents of the case studies in Chapter Six (refer section 6.5).

The expert review on the face validity of the survey questions has yielded several refinements as listed in Table 3.5. In general, the experts agreed that the questions were able to address the usability of the model in solving the research problems. However, the focused questions were suggested to relate back with how work was carried out in the past, before the model was implemented and the reference to the CEF system instead of just system used in the survey. The finalized instrument after the refinement is attached in Appendix C.

Table 3.5

*Expert Comments on Survey Questions*

| **Before Refinement** | **After Refinement** |
|---|---|
| 1. Using the system in my job would enable me to accomplish tasks more quickly. | 1. Using the CEF system environment in my job would enable me to accomplish tasks more quickly. |
| 2. Using the system would improve my job performance. | 2. Using the CEF system environment would improve my job performance. |
| 3. Using the system in my job would increase my productivity. | 3. Using the CEF system environment in my job would increase my productivity. |
| 4. Using the system would enhance my effectiveness on the job. | 4. Using the CEF system environment would enhance my effectiveness on the job. |
| 5. Using the system would make it easier to do my job. | 5. Using the CEF system environment would make it easier to do my job. |
| 6. I would find the system useful in my job. | 6. I would find the CEF system environment useful in my job. |
| 1. Learning to operate the system would be easy for me. | 1. Learning to operate the CEF system environment would be easy for me. |
| 2. I would find it easy to get the system to do what I want it to do. | 2. I would find it easy to get the CEF system environment to do what I want it to do. |
| 3. My interaction with the system would be clear and understandable. | 3. My interaction with the CEF system environment would be clear and understandable. |
| 4. I would find the system to be flexible to interact with. | 4. I would find the CEF system environment to be flexible to interact with. |
| 5. It would be easy for me to become skillful at using the system. | 5. It would be easy for me to become skillful at using the CEF system environment. |
| 6. I would find the system easy to use. | 6. I would find the CEF system environment easy to use. |
| 1. Managing Cloud ERP solution is easier than the previous method. | 1. Managing Cloud ERP solution is easier compared to the previous method. |
| 2. The model has provided me with easier management tools for my work. | 2. The CEF system environment has provided me with easier management tools for my work compared to the previous method. |
| 3. The model was able to reduce my daily workload of jobs. | 3. The CEF system environment was able to reduce my daily workload compared to previous method. |
| 4. The model was able to improve the overall management of workload in the company. | 4. The CEF system environment was able to improve the overall management of workload in the company compared to the previous method. |

Table 3.5 continued

| | |
|---|---|
| 5. The model was able to maintain the software quality being developed compare to previous method. | 5. The CEF system environment was able to maintain the software quality being developed compared to previous method. |
| 6. The model provides better step by step procedures to ensure consistency. | 6. The CEF system environment was able provides better step by step procedures to ensure consistency compared to previous method. |
| 7. I am able to retrieve essential knowledge to complete my task. | 7. I am able to execute my task with the help of CEF knowledge management system compared to previous method. |
| 8. The model provides me with the tools for me to update the Knowledge repository. | 8. The CEF system environment was able to provide better knowledge retention compared to previous method. |

## D. Revised Items for Survey Instrument

Based on the comments made by the expert face to face sessions, the research has grouped the questions in the survey in order to correlate with the research problems, as shown in Table 3.6 as the Revised Items for the survey instrument.



Table 3.6

*Revised Items for Survey Instrument with Grouping*

| No | Perceived Usefulness | Perceived Usefulness | Ease of Use | Reduce Workload | Reduce Complexity | Inconsistent Quality | Knowledge Retention |
|---|---|---|---|---|---|---|---|
| 1 | Using the CEF system environment in my job would enable me to accomplish tasks more quickly. | √ | | | | | |
| 2 | Using the CEF system environment would improve my job performance. | √ | | | | | |
| 3 | Using the CEF system environment in my job would increase my productivity. | √ | | | | | |
| 4 | Using the CEF system environment would enhance my effectiveness on the job. | √ | | | | | |

Table 3.6 continued

| | | | | | | |
|---|---|---|---|---|---|---|
| 5 | Using the system environment would make it easier to do my job. | √ | | | | |
| 6 | I would find the CEF system environment useful in my job. | √ | | | | |
| | **Perceived Ease of Use** | | | | | |
| 7 | Learning to operate the CEF system environment would be easy for me. | | √ | | | |
| 8 | I would find it easy to get the CEF system environment to do what I want it to do. | | √ | | | |
| 9 | My interaction with the CEF system environment would be clear and understandable. | | √ | | | |
| 10 | I would find the CEF system environment to be flexible to interact with. | | √ | | | |
| 11 | It would be easy for me to become skillful at using the CEF system environment. | | √ | | | |
| 12 | I would find the CEF system environment easy to use. | | √ | | | |
| **No** | **Additional Questions** | | | | | |
| 1 | The CEF system environment was able to reduce my daily workload compared to previous method. | | | √ | | |
| 2 | The CEF system environment was able to improve the overall management of workload in the company compared to the previous method. | | | √ | | |
| 3 | The CEF system environment was able to maintain the software quality being developed compared to previous method. | | | | | √ |
| 4 | The CEF system environment was able provides better step by step procedures to ensure consistency compared to previous method. | | | | | √ |
| 5 | Managing Cloud ERP solution is easier compared to the previous method. | | | | √ | |
| 6 | The CEF system environment has. provided me with easier management tools for my work compared to the previous method. | | | | √ | |

Table 3.6 continued

| | | |
|---|---|---|
| 7 | I am able to execute my task with the help of CEF knowledge management system compared to previous method. | √ |
| 8 | The CEF system environment was able to provide better knowledge retention compared to previous method. | √ |

## E. Pilot Study for Reliability Testing

Reliability of a measure is an indication of consistency. The value of Cronbach's coefficient alpha was calculated and should indicate the value of alpha, to be accepted as reliable (Sekaran, 2013). A pilot study was conducted to ensure that the instrument measure the intended variables and with regards to its validity and reliability. In this study, Cronbach's Alpha was computed in order determine the questionnaire dimension internal consistency, in which, 72 respondents were involved. From the analysis, all dimensions were found to be highly reliable as depicted in Table 3.7. These results show that the instrument used were consistent and reliable, thus justifiable to be used for data collection in the field usability case studies.

Table 3.7
*Questionnaire Reliability Test*

| **Dimension** | **Cronbach's Alpha** | **N of Items** |
|---|---|---|
| Perceived Usefulness | 0.956 | 6 |
| Perceived Ease of Use | 0.918 | 6 |
| Reduce Workload | 0.937 | 2 |
| Reduce Complexity | 0.878 | 2 |
| Consistent Quality | 0.936 | 2 |
| Knowledge Retention | 0.911 | 2 |

### 3.5.4 Phase 4: Evaluation

The purpose of the evaluation phase of the CEF Model is to prove that the model is the right solution to tackle the research problem components. In software engineering domain, Boehm (1979) and IEEE standard (IEEE, 1998) stated that validation and verification are both test approach in system model evaluation, in which validation is a test as to whether the right product is built and verification is tested, whether the product has been built correctly according to specification. Both validation and verification methods were used across our evaluation phase. In Expert Review, experts were involved in validating and verifying the model. In Chapter Five, once the model has been developed into systems, verification is employed as a method to ensure that the product was developed correctly based on the specifications listed in the design guidelines. After the completion of the system development, the CEF model prototype environment is then validated using usability testing to ensure that the model is usable and able to tackle the research problem components.



The Expert Review was conducted as the first point of validation of the main artifact. The input gathered from the expert review were used to enhance the model and then presented as the main artifact and the completion of objective number two. Once the model was finalized, our concern was to verify technical feasibility in deploying the model into a Cloud ERP production environment. An existing commercial Cloud ERP company was selected to be transformed into CEF model prototype environment. The deployed CEF model environment was successfully verified with built-in checklist from the model. Once completed, the CEF prototype environment was used to further evaluate the CEF Model on the field usability aspect and its ability to address the research problem components using the developed questionnaire described the instruments development section. This method of field usability testing was highlighted by Delikostidis (2007) pointing out that field usability is an evaluation method, which is able to assess the usability of a model in a specific context of use. The result was then evaluated and analyzed to formulate the conclusion of this research. Figure 3.5 shows the detail flow of the activities in evaluation phase and the next subsections will elaborate further the activities.

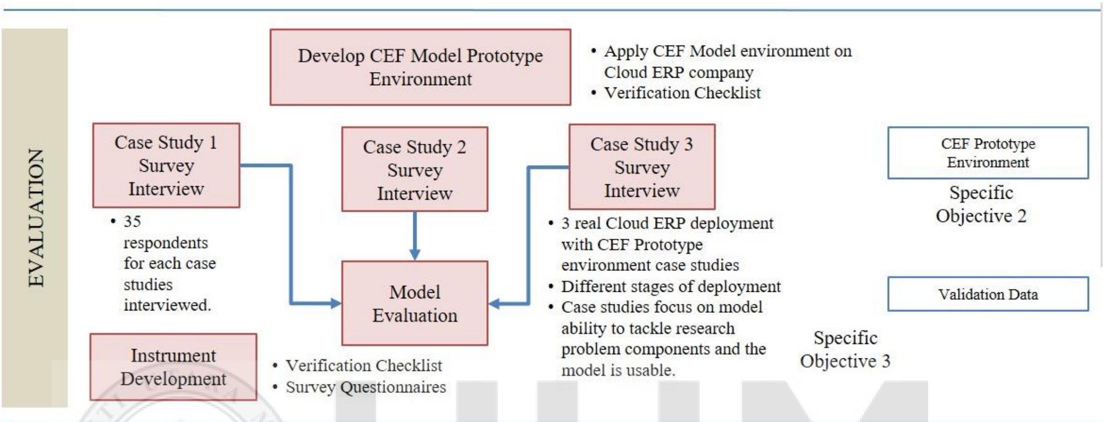



*Figure 3.5* Evaluation Phase Activities

### 3.5.4.1 Field Usability Testing

Usability testing can be considered one of the pivotal element in system design, especially in software development (Rubin, J., & Chisnell, 2008). It can be referred to as a method or tool that can be used to assess the effectiveness and efficiency of a system's capacity to meet users' requirements (Satam, Taslim, Adnan, & Manaf, 2017). Rubin, J., & Chisnell, (2008) points out that there are four types of usability testing, which are Exploratory, Assessment, Comparison, and Validation. The assessment usability test practically assesses the features developed during an implementation whereas validation usability test certifies the features following the benchmark or standards required. During the face validity review, the experts agreed that the PUEU is the best choice as an instrument (see section 3.5.3.3.2) to assess the usability of the system, the experts then added an additional eight questions as a form of validation usability. The eight additional questions focus on the model's ability to handle the research problem components.

Most of the usability testing points to user experience when dealing with user interfaces of systems or products (Isa, Lokman, Wahid, & Sulaiman, 2014). Our CEF Model however, consists of several systems and procedures that forms an environment, which require the testing to be conducted in a real-world environment. This is similar with the field testing conducted to test usability in bio-medical and on-line banking industry (Kantner, Sova, & Rosenbaum, 2003; Rosenbaum & Kantner, 2007).

In our research, the field usability conducted was focused on the assessment and validation usability testing to ensure that the prototype environment is able to be used in a real-world situation as most of the challenges described in the previous section is faced in the real-world Cloud ERP production environment. Apart from testing the ability of the model to handle our research problem components, the assessment usability testing is a method to test that the developed system is able to be used.



#### 3.5.4.2 Model Environment Prototyping

In order to verify the CEF Model's technical feasibility in terms of its ability to be deployed, the model prototype environment was developed using a selected Cloud ERP software production facility. This process was also a model feasibility verification process that was used to verify the CEF Model replication ability. Therefore, this study has developed a prototype environment that follows the model five component design guidelines from the finalized model. By using the model built in checklist, it can be determined whether the environment prototype built can fulfill all the checklist requirements. The checklist base verification approach was used to inspect and verify their research model by Maiani de Mello, Nogueira Teixeira, Schots, Lima Werner, and Travassos (2014). The same approach is utilized in verifying our model technical feasibility by using built-in verification checklist from the CEF model itself. The verification checklist development is further described in Chapter Four (refer to section 4.6). The prototype deployment team was set up to deploy the CEF model as a validation vehicle in the field usability case studies. Figure 3.5 shows the prototyping process and its linkage to other processes. The prototyping phase was the end of development phase and at this stage, the research is ready for the evaluation phase.

Once the model was finalized, a commercial Cloud ERP production company was selected to be the host of our CEF model environment prototyping activities. The model environment prototyping phase involves four phases of activities those are:



i. CEF modeling requirement study

   The CEF model was applied to a Cloud ERP production environment based on the guidelines provided.

ii. Model to System Development

   The affected component of the model is the developed into a system or tool.

iii. Model Checklist Verification

   Verification of the developed system by built-in verification checklist provided.

iv. CEF System training

   The completed CEF system training was provided to the selected company as means of assisting the personnel to actually use the developed system.

### 3.5.4.3 Model Validation using Field Usability Case Studies

Model Validation has utilized the prototype of the CEF Model environment developed in previous phase as the validation vehicle. Using this prototype environment, three real-world commercial ERP implementations were conducted in order to demonstrate the model's usability and its ability to tackle the research problem components. The instrument for this validation will be presented to respondents, in this case made up of all the actors in the three case studies. The actors in this context consisted of the personnel or department directly connected to the whole process as shown in Figure 3.6. Each party was presented with a set of questionnaires to validate the usability of the processes implemented in the model.

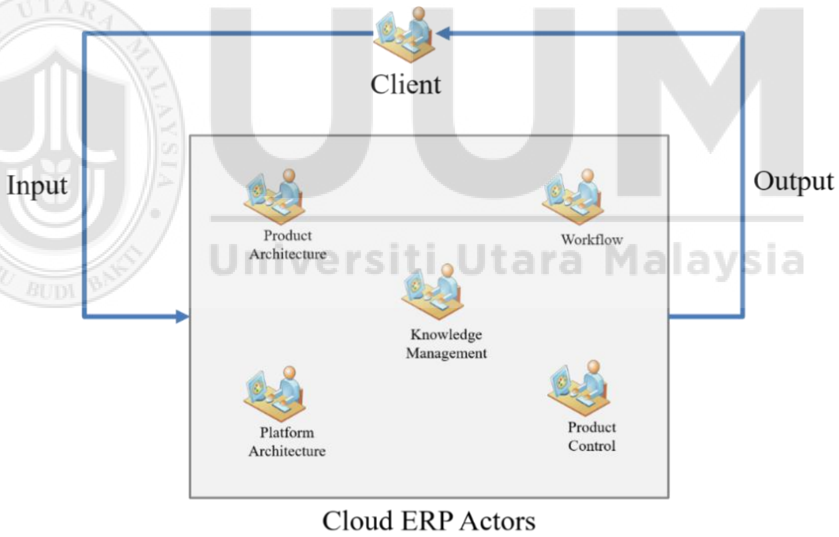



*Figure 3.6* Five Validation of Field Usability Respondent Groups based on the Actors Role within the CEF Model

The whole process of Cloud ERP life cycle was monitored and both qualitative and quantitative method were deployed. The three field usability case studies were conducted independently as well as concurrently. The study of the model usability had involved three deployments of real live Cloud ERP implementations. The field usability case studies also include survey questions, which specifically address the model's ability to address the research problem components.

#### 3.5.4.4 Unit of Analysis

The unit of analysis is commonly referred to as the key entity being studied during successive data analysis period (Sekaran, 2013; Trochim & Donnelly, 2006). In a research, unit of analysis could be of individuals, groups, and artifacts. Moreover, for different analyses in the same study, different units of analysis may have been identified. Below are the units of analysis identified throughout this research:

i. **Experts in Questionnaire face validity:** Ten experts from different background such as ERP, Cloud ERP, software engineering, and computer science.

ii. **Expert in expert review of proposed model:** Expert reviews (refer appendix A) were conducted to validate the proposed CEF Model. It involves 1 international expert, 4 local experts, and 2 experts from academia.

iii. **Respondents in Each Case Study Survey:** The respondents from the survey involved 35 people involved with Cloud ERP implementation and deployment. Total respondents were 35 x 3 = 105 respondents.

#### 3.5.4.5 Data Analysis

Validation of the field usability case studies was done in parallel with the case studies. Quantitative and Qualitative gathered data were segmented by the problems described in our Problem Statement section in Chapter One. Figure 3.7 depicts the data analysis flow model from each case study to the model usability result. Research instruments such as questionnaires and interview questions were formulated to test the usability of the CEF model.

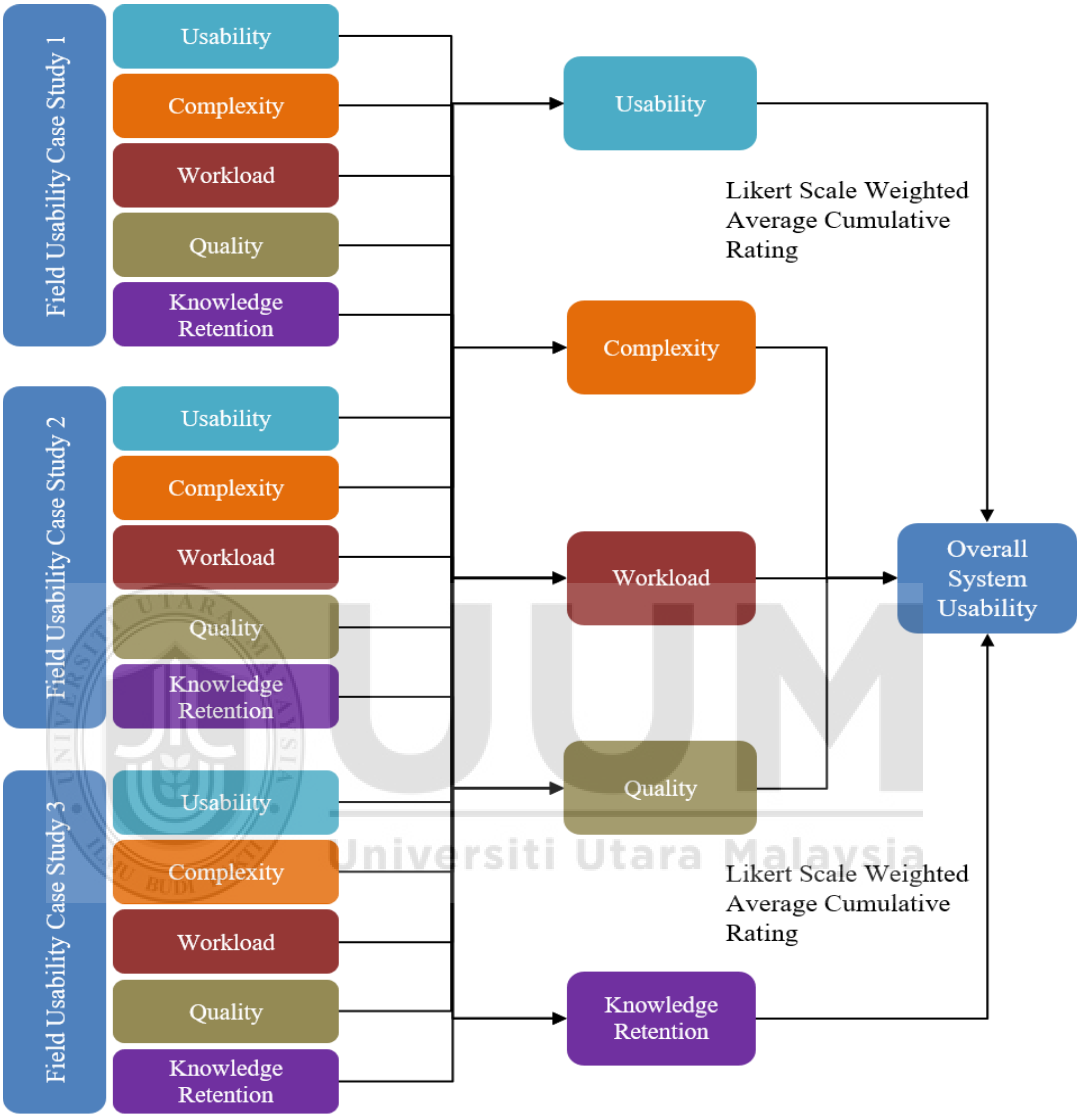



*Figure 3.7* Field Usability Validation Data Analysis Model

For this research, three parts of validation assessment was conducted, in order to confirm the proposed model. Those are (i) expert review, (ii) technical feasibility of deploying the prototype CEF Model environment, and (iii) field usability case studies to determine the usability of the model in real world environment.

The combination of three evaluation levels was to ensure that the final proposed model represents the model that was really usable to the commercial ERP industry as well as being able to tackle the research problem components. The field usability case study was carried out in the actual Cloud ERP implementation that has employed the elements depicted in the model in the development, deployment, and post maintenance of ERP. Each case study was carried out in a controlled manner, so that the independent elements were consistent for all of the three case studies. The usability of the elements and model was documented for analysis work.

As most of the validation data is in the form of Likert Scale format, data analysis was evaluated by weighted average for each case study independently, which was then compiled for final analysis to make up the conclusion to gauge the usability of the environment prototype, which reflects the CEF Model.



### 3.5.5 Phase 5: Conclusion

In the final phase as shown in Figure 3.8, all the findings gathered in each of the previous phases were concluded through revisiting and answering each of the research questions and research objectives. Finally, this study produces the full thesis and several publications as the contribution to the body of knowledge and theory.

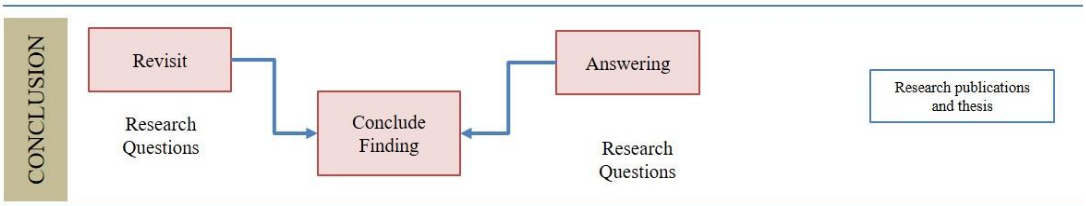



*Figure 3.8* Conclusion

### 3.6 Summary

As a conclusion, this chapter discusses the research approaches that were adapted in this study. Design Science Research methodology with mixed method approach has been selected to be utilized throughout this study, which includes five major phases, which are (i) awareness of problem, (ii) suggestion, (iii) development, (iv) evaluation, and (v) conclusion. Each of the phases was described in detail with the activities that was carried out throughout this study. The main phase diagrams and its subphases diagrams were included in the chapter to help in explaining the research activity linkages.

# CHAPTER FOUR
# MODEL DEVELOPMENT

## 4.1 Introduction

This chapter focuses on the model development phase for the CEF Model. This includes guidelines, components, specific activities, and deliverables. It explains the overall development and validation process of the CEF Model. The main focus in mind would be to provide a simplified blueprint for Cloud ERP providers on how to run a standardized Cloud ERP production by following the steps explained in the CEF Model. In doing so, several assumptions are spelled out as below to allow the full utilization of the CEF Model.

i. The intended audience should be familiar with ERP software development processes especially Cloud environment.
ii. Implementation of the model allows for standardization of production which relates to the need to have volume in terms of customers and software products. It is neither advisable nor feasible to implement this model if the current business strategy revolves around one customer with one type of product or project based.
iii. There should be a minimum number of human resources structure in place to ensure that the model is implemented properly.

## 4.2 Model Development Approach

In order to explain the CEF Model approach, reference will be made to the same concept of how a standard factory with manufacturing processes work as compared to a Cloud ERP Factory model.

The standard terminology for manufacturing processes such as Bill of Material (BOM), Product Process routing, and Process Work Centers will be used as a comparison to our terminology in Cloud ERP Factory model in order to elaborate the model. For any standard product to be manufactured there will be a documented BOM and Process Routing. This is similar to a cooking recipe with ingredients as BOM and cooking methods as the processes and routing. In our CEF Model, BOM can be referred to the consumed resources by development processes. For example, specific programmer's time is a required BOM. Work Center within the CEF Model is referred to as a department which provides specific services such as software coding, testing, and implementation. Process routing is then referred to as the process flow for a specified software application or product. The CEF Model aims to allow for a semi-automated environment, which provides programmers to complete the daily assigned jobs systematically instead of the normal "ad-hoc" type requests.

After obtaining the essential five components for the CEF Model based on the comparative analysis discussed in Chapter Two, a comparative analysis was made for each component based on related research journals to generate its core elements.

Table 4.1

*Product Line Core Elements*

| **Author/Core Elements** | **System Functional Architecture** | **System Module Components** | **Integration Model** |
|---|---|---|---|
| Software product lines essentials (Northrop, 2008) | √ | √ | √ |
| Multiple software product lines for service oriented architecture (Kamoun, Kacem, & Kacem, 2016) | √ | | √ |
| Modularity in the Context of Product Line Variability (Kang, 2013) | √ | √ | √ |
| Variability Management in Software Product Line Engineering (Pohl, Metzger, & Pohl, 2006) | √ | √ | √ |
| Introduction to software product lines (Donohoe, 2009) | √ | √ | |
| Variability Middleware for Multi-tenant SaaS Applications Categories and Subject Descriptors (Landuyt, Walraven, & Joosen, 2015) | √ | √ | |
| What do we know about software product management? - a systematic mapping study (Maglyas, Nikula, & Smolander, 2011) | √ | | √ |



Table 4.1 exemplifies that in Product Line Architecture components, three main core elements are identified as System Functional Architecture, System Module Components, and with Cloud ERP, an integration model is required.

Table 4.2

*Platform Core Elements*

| **Author/Core Elements** | **User Portal as a Platform** | **System Architecture Platform** | **Backup / Redundancy** |
|---|---|---|---|
| Cloud Environment Assignment: A Context-aware and Dynamic Software Product Lines-Based Approach (Benalil, Asri, & Kriouile, 2015) | √ | √ | |
| On the Dependability for Dynamic Software Product Lines A Comparative Systematic Mapping Study (Eleutério et al., 2016) | √ | √ | √ |
| Towards the Selection of Modeling Techniques for Dynamic Software Product Lines (de Jesus Souza, Magno Lua Santos & de Almeida, 2015) | √ | √ | |
| Dynamic Software Product Line: An Approach to Dynamic Binding (Banerjee & Kumari, 2016) | √ | | √ |
| Sharing with a difference: Realizing service-based SaaS applications with runtime sharing and variation in dynamic software product lines (Kumara, Han, Colman, Nguyen, & Kapuruge, 2013) | | √ | √ |

Table 4.2 illustrates that most research journals identify User Portal Platform, Developer Platform, and Backup as the major elements in platform research. To present the platform architecture in the model, the SOA Layer concept described by Emig, Langer, Krutz, and Link (2006) was employed, which have defined a standardized layering with an SOA.

Table 4.3

*Core Elements for Process Flow*

| Author/Core Elements | System Inputs | Work Centers | Jobs and Task | Work Order Routing | System Output |
|---|---|---|---|---|---|
| Monitoring processes through inventory and manufacturing lead time (Cuatrecasas-Arbós et al., 2015) | √ | √ | √ | √ | |
| Two-phase methodology for customized product design and manufacturing (Kamrani et al., 2012) | | | √ | √ | √ |
| Improving manufacturing process for biomedical products: a case study (Bimal Nepal et al., 2011) | √ | √ | √ | | |
| Dynamics of manufacturing productivity: lesson learnt from labor intensive industries (Shahidul, Shazali, & Syed Shazali, 2011) | | √ | √ | √ | |
| An examination of the relationship between manufacturing flexibility and firm performance (Camisón & Villar López, 2010) | √ | √ | | √ | √ |
| Autonomic smart manufacturing (Menascé et al., 2015) | √ | √ | √ | √ | √ |
| An approach to monitoring quality in manufacturing using supervised machine learning on product state data (Wuest et al., 2014) | | √ | √ | √ | |
| An integrated model for performance management of manufacturing units (Parthiban & Goh, 2011) | √ | | √ | √ | √ |



Referring to Table 4.3, it is evident that most research into the processes of manufacturing looks at five different elements. These elements are System Input, Work Centers, Jobs and Task, Routing templates, and System Output.

Table 4.4

*Core Elements for Product Control*

| **Author/Core Elements** | **System Identification** | **Version & Revision Control** | **Release Control Model** | **Software Updates** | **Anti-Software Infringement** |
|---|---|---|---|---|---|
| Dynamic configuration management of cloud-based applications (Schroeter et al., 2012) | √ | √ | √ | √ | |
| Version Models for Software Configuration Management (Conradi & Westfechtel, 1998) | | | √ | √ | √ |
| A variability aware configuration management and revision control platform (Linsbauer, Egyed, & Lopez-Herrejon, 2016) | √ | √ | √ | | |
| A software configuration management model for supporting component-based software development (Mei, Zhang, & Yang, 2001) | | √ | √ | √ | |
| UaaS: Software Update as a Service for the IaaS Cloud (K. Liu, Zou, & Jin, 2015) | √ | √ | √ | √ | √ |
| Tales of Software Updates: The process of updating software (K. Vaniea & Rashidi, 2016) | | | √ | √ | |
| Dynamic software update (Jhanwar & Yaryan, 2007) | | | √ | √ | |

Referring to Table 4.4, most research, concerning Software Configuration Management, has identified five major elements, which are System Identification, Version & Revision Control, Release Control Model, Software Updates, and Anti-Software Infringement.

Table 4.5

*Knowledge Management Core Elements*

| **Author/Core Elements** | **Knowledge Identification** | **Knowledge Retrieval & Dissemination** | **Dynamic Knowledge Logging** | **Knowledge Mapping** |
|---|---|---|---|---|
| Reusable active learning system for improving the knowledge retention and better knowledge management (Sridharan & Kinshuk, 2003) | √ | √ | | √ |
| Getting past knowledge management (J. Smith, 2006) | √ | √ | √ | √ |
| Knowledge Retention in Knowledge Management System: Review (Ghahfarokhi & Zakaria, 2009) | | √ | √ | |
| The Current Ecosystem of Learning Management Systems in Higher Education: Student, Faculty, and IT Perspectives (Dahlstrom, Brooks, & Bichsel, 2014) | √ | √ | | √ |
| Enhancement of Learning Management System with Adaptive Features (Ishak & Ahmad, 2016) | √ | √ | | |
| Learning Management System (LMS): The Missing Link and Great Enabler (Phillipo & Krongard, 2012) | | √ | √ | √ |
| Learning Management System success: Increasing Learning Management System usage in higher education in sub-Saharan Africa (Mtebe, 2015) | √ | √ | √ | |
| Sustainability Management within Selected Large-Scale Enterprises in Germany | | √ | √ | √ |
| Incremental knowledge acquisition and self-learning from text (De Silva & Alahakoon, 2010) | √ | √ | √ | |
| Knowledge Management Strategy and Structure in Service Sector (Gulati & Srivastava, 2014) | | √ | √ | √ |



Table 4.5 proves that Knowledge Management and Learning Management System focuses into the same common areas such as Knowledge Identification, Retrieval and Dissemination, Knowledge Logging, and Knowledge Mapping.

From all of the comparative analyses conducted, it appears that these are the major elements that are crucial to our CEF Model. These elements are then grouped together to address the five components that makes up the CEF Model as shown in Figure 4.1.

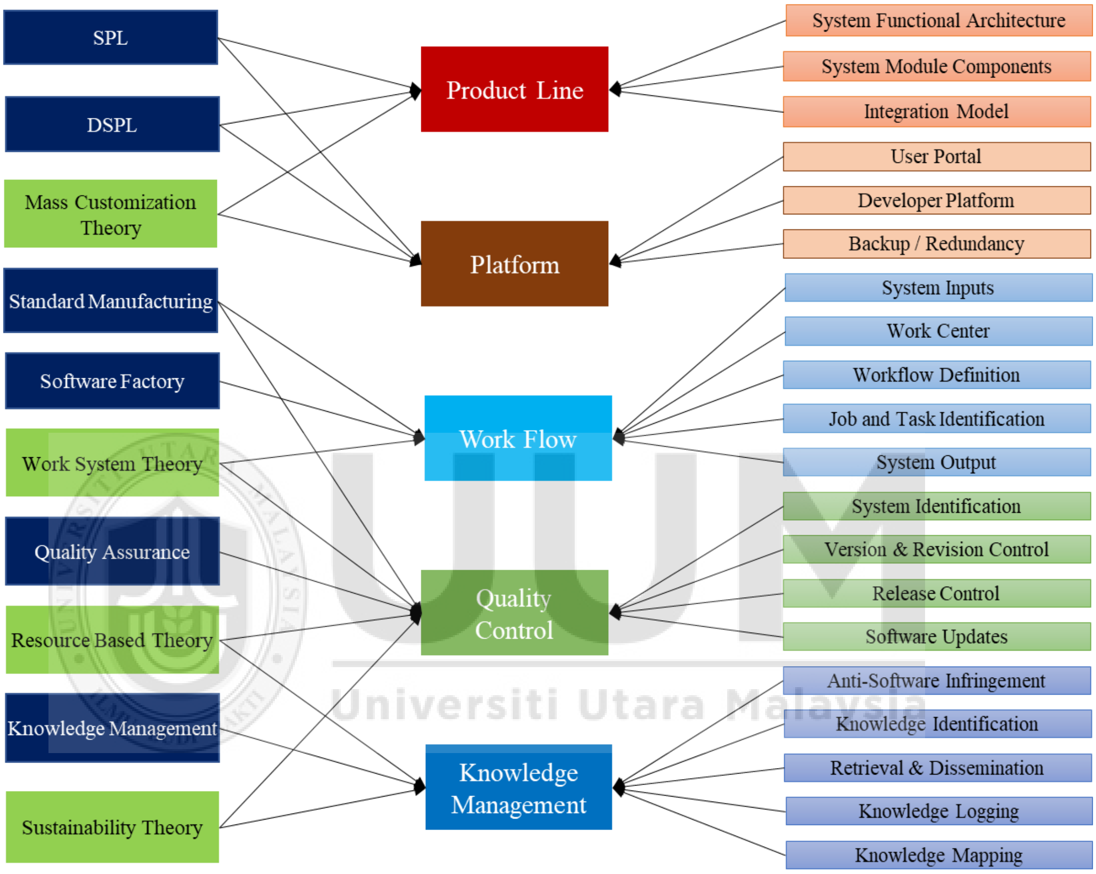



*Figure 4.1* Core Model Elements Mapping Identification

### 4.3 Cloud ERP Factory (CEF) Model Overview

The CEF Model should consist of five main components, which are CEF Product Line Architecture, CEF Platform Architecture, CEF Workflow Model, CEF Product Control, and Knowledge Management as shown in Figure 4.2. The CEF Product Line Architecture is mainly focused on the development and criteria of Cloud ERP products. CEF Platform Architecture details the requirements to develop a platform that is able to support the CEF Model. CEF Workflow, on the other hand, is the crucial engine for the entire CEF Model as it dictates the steps and flow of all the procedures within a Cloud ERP production environment. To manage the types of Cloud ERP products developed, the CEF Product Control looks into the methods to ensure the ability to support and deploy over the Cloud. The Knowledge Management, however, is the bind retaining all of the knowledge accumulated in a repository, thus, promoting a more self-sustaining ecosystem between employees and knowledge. These five components should be able to provide a comprehensive guide for Cloud ERP providers to create an environment of Cloud ERP production model.

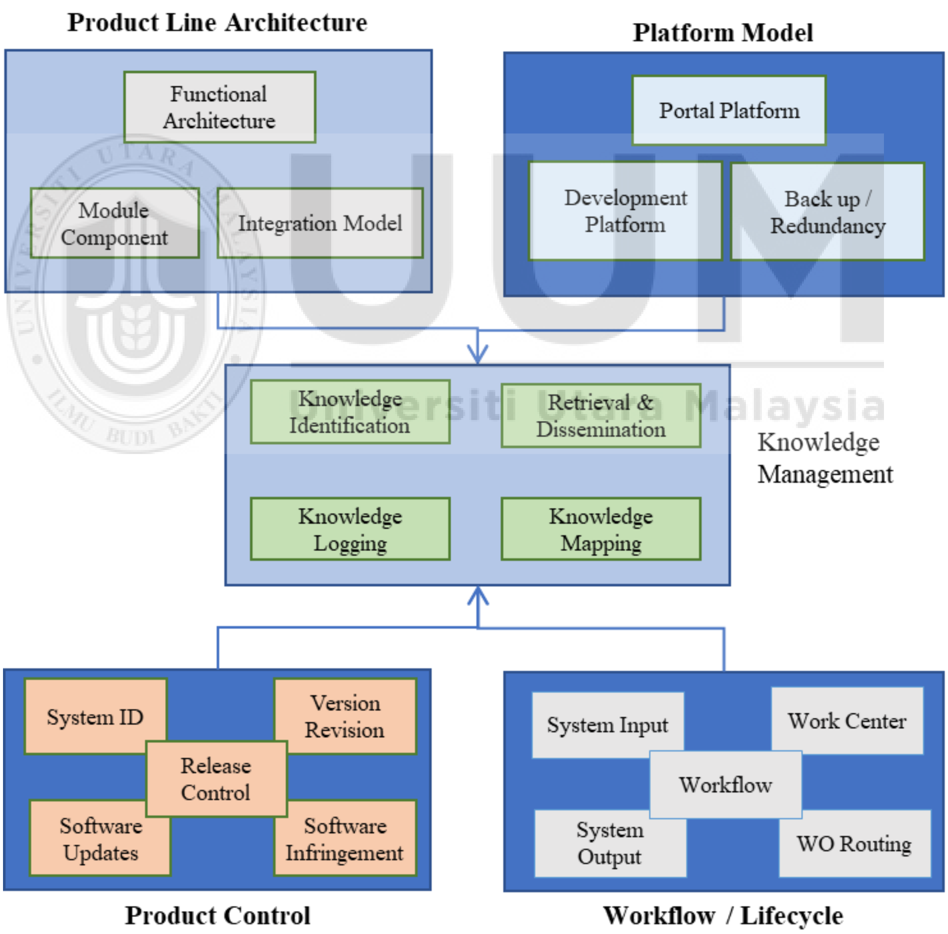



*Figure 4.2* Proposed Cloud ERP Factory (CEF) Model

### 4.4 CEF Model Subcomponents

In the below subsections, each of the subcomponents' model development for the CEF Model will be explained further. Each subcomponent includes core model elements and guidelines, which would ensure that the subcomponent is able to be modelled by other Cloud ERP providers.

#### 4.4.1 CEF Product Line Architecture

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 has to clearly define the product range and scope that will be built by the setup factory. Just like when producing a physical product in a traditional manufacturing factory, the product family to be manufactured must be clearly identified. By defining Cloud ERP product line, the manufacturing capability requirement of the factory can be made known. Figure 4.3 shows the Product Architecture component of the CEF Model that can be used to define the Cloud ERP Product that can be built by the factory.

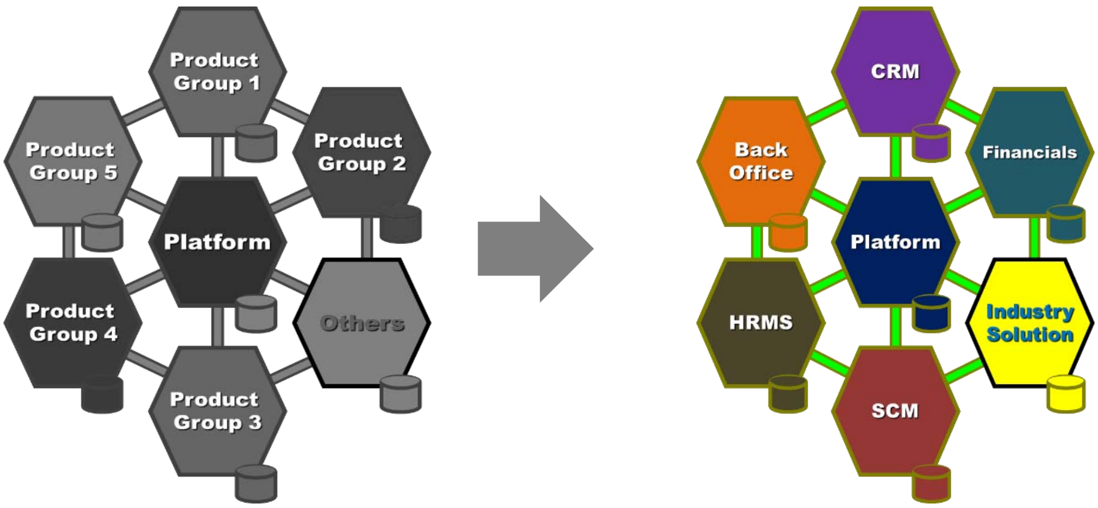



*Figure 4.3* Overall CEF Product Line Architecture

#### 4.4.1.1 Product Line Component Objectives

The following are the objectives of this component:

i. To promote reusability by utilizing pseudo-products.

ii. To promote product modularity and scalability.

iii. To promote clear product mapping group so that future products can be easily slotted into one of the existing groups.

#### 4.4.1.2 Product Line Model Elements

For CEF Product Line Architecture, the core model elements are System Functional Architecture, System Module Component, and Integration Model.

##### 4.4.1.2.1 System Functional Architecture

CEF Product Line should consist of a set of core ERP modules, which are mostly applicable to most enterprise system deployment and industry specific solution modules. A typical example set of a core CEF module includes Customer Relationship Management (CRM), Supply Chain Management (SCM), Financial Information System (FIS), Human Resources Management System (HRMS), and Back Office Solutions. The model should also take into consideration that any industry specific solution can be integrated into the core ERP module during the deployment phase as depicted in Figure 4.4.

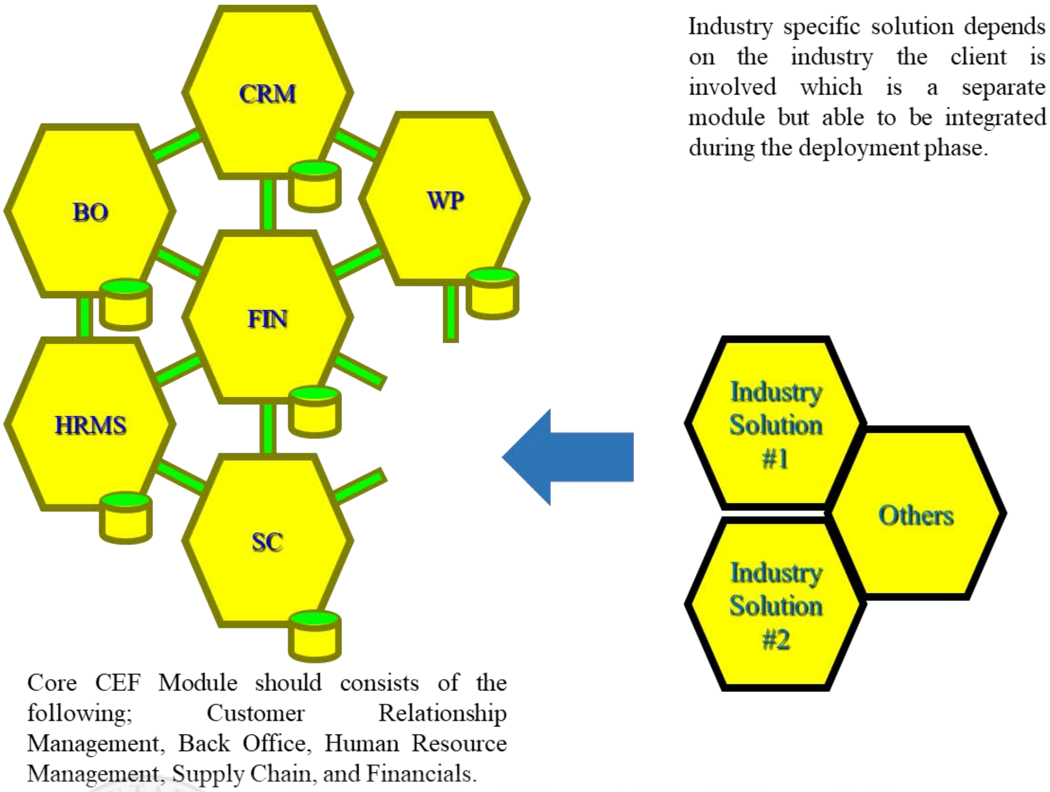



*Figure 4.4* Core ERP Modules with Industry Solution

#### 4.4.1.2.2 System Module Component

System module component defines the components that make up each system. The purpose is to standardize the component of every system module. Each system module should consist of an interface component, business services object component, and database component. These components serve independently and uniquely in terms of functionality from other modules. They represent the system and its functionality and physically they are source code or program scripts and data storage format. Figure 4.5 shows the components in layer presentation and it is actually part of the CEF Platform Architecture model, which will be explained in next subcomponent section.

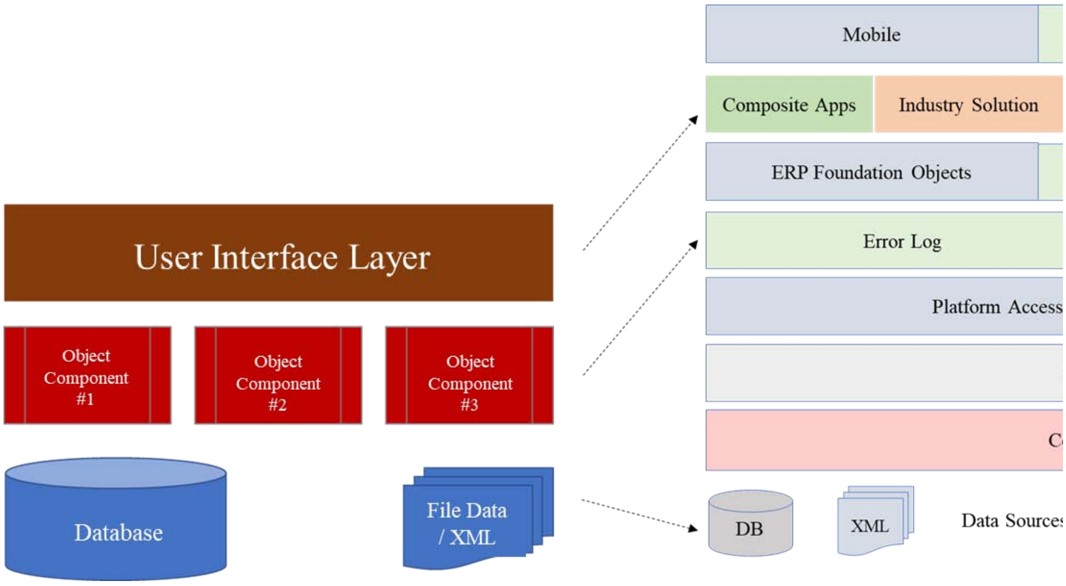



*Figure 4.5* System Module Component

#### 4.4.1.2.3 Integration Model

Each of the modules defined should then be able to be integrated with each other as well as with any third-party Cloud applications. The business functionality services protocol should be well defined for other internal modules to communicate with this system and Application Program Interface (API) should be defined for external applications through the platform communication channel. The main component for this purpose is the business object services. For illustrative purposes, the Figure 4.6 shows an example how the system functionality modules connectivity across each other. The connectivity represents data flow between the system modules.

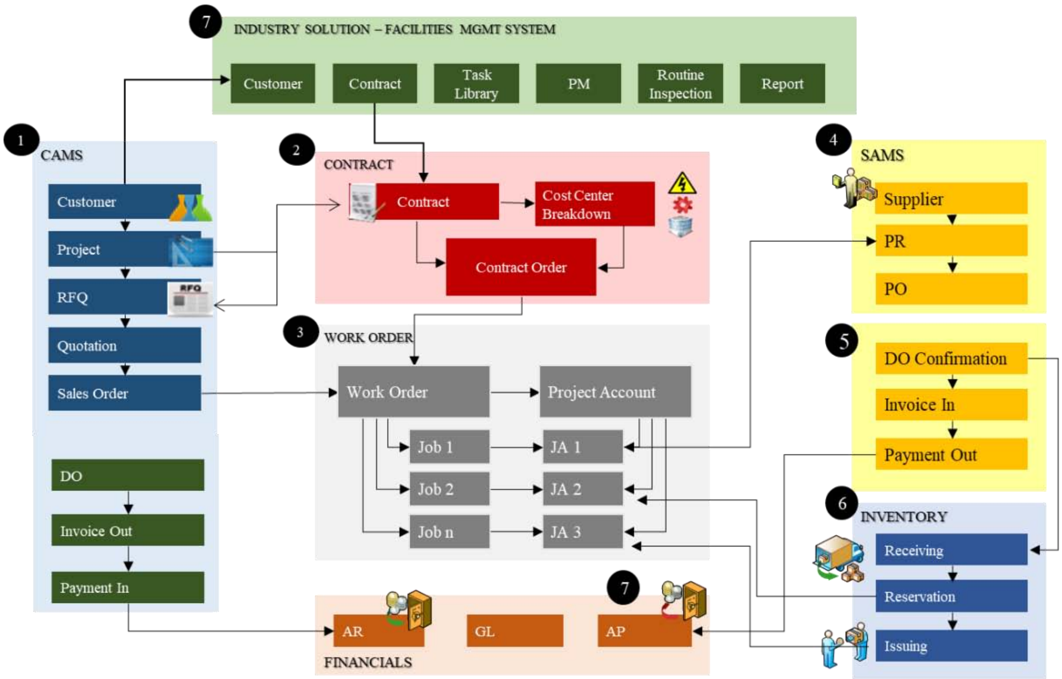



*Figure 4.6* System Functionality Module Connectivity

### 4.4.1.3 Product Line Design Guidelines

Table 4.6 shows the design guidelines for the CEF Product component.



Table 4.6

*CEF Product Line Design Guideline*

| No. | Guidelines | Core Model Elements |
|---|---|---|
| 1 | Define core product and their grouping to enhance reusability and modularity. The product line should be able to group current software modules as well as any future potential modules. | System Functional Architecture |
| 2 | Define pseudo product and their grouping to enhance reusability and modularity. | System Functional Architecture |
| 3 | Define industry solution product and their grouping. | System Functional Architecture |
| 4 | Define system module component that makes up a system module. | System Module Component |

Table 4.6 continued

| | | |
|---|---|---|
| 5 | Define foundation system component if applicable. | System Module Component |
| 6 | Define integration structure between all system modules and any legacy systems. Cloud ERP system modules should be loosely coupled in a way they can be subscribed and function independently but also be able to be integrated together. | Integration Model |
| 7 | Define integration between core ERP module and Industry Solutions. | Integration Model |
| 8 | Define product management model. All product information should be managed systematically by a system. | NA |
| 9 | Define audit method that can enforce the practice without fail. | NA |

#### 4.4.1.4 Product Line Summary

As a summary, CEF Product Line Architecture is an important element required before the deployment of the CEF Model can be initiated. This is the first step and consideration in the deployment of the CEF Model. The product architecture has to be firmed up before the CEF approach can be taken advantage of, which focuses on the ability to reduce the complexity of managing Cloud ERP solutions. The model offers only as a guide to develop a product family and the actual product group may have different module and system naming. Product determination is related closely to the Platform architecture, which will be discussed in the next section.

### 4.4.2 CEF Platform Architecture

The CEF Platform Architecture is a reusable component of Cloud ERP Product Architecture. The platform is defined as a common foundation of the Cloud ERP Product, which is based on the Serviced Oriented Architecture Layer model. For the case of Cloud ERP, three types of platform will be proposed, those are platform as user platform, system architecture platform, and lastly platform as the foundation component. Each type of platform carries a significant importance to the CEF Model as it ensures the type of product to be released.

#### 4.4.2.1 Platform Component Objectives

i. Promote Reusability using object oriented and service oriented approach.
ii. Promote Systematic Layer Model for Structured Programming.
iii. Promote Standard interface through single sign on use portal.

#### 4.4.2.2 Platform Core Model Elements

Platform Architecture core model elements deals with the different type of platforms in the overall architecture. The core model elements are User Portal Platform, Software Architecture Platform, and Backup/Redundancy.



##### 4.4.2.2.1 User Portal Platform

The CEF Model does not dictates the user portal platform features. However, the feature diagram shown in Figure 4.7 is recommended for management standardization and simplicity. The User Portal Platform is important element as it will provide user accessibility to the modules defined in the previous section. Users should be able to access the assigned modules as well as other features in the provided platform. User management, security management, and accessibility management are also able to be conducted in the User Portal Platform.

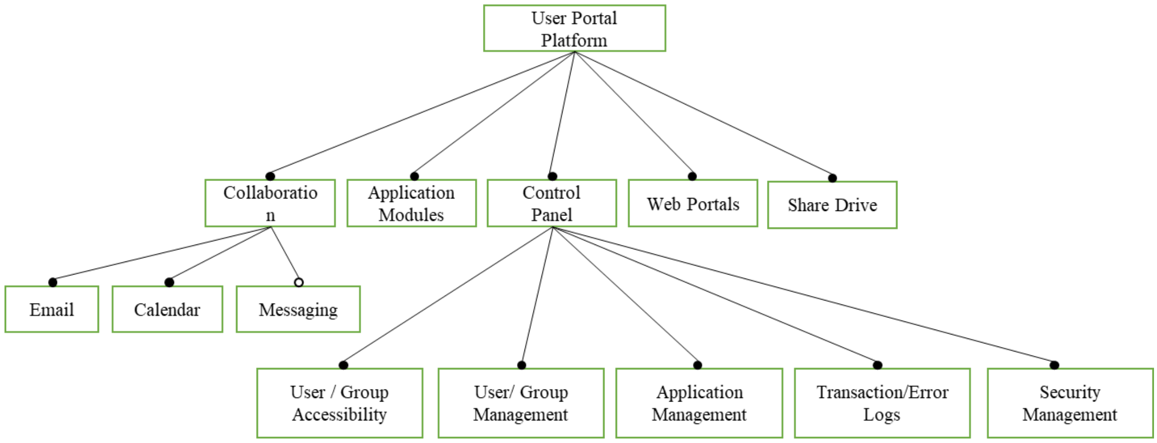



*Figure 4.7* User Portal Platform Feature Diagram

#### 4.4.2.2.2 System Architecture Platform

System Architecture Platform represents the architectural layer of programming services by functionality. In order for all Cloud ERP system modules to perform consistently, the software architecture platform is made to define the services layer by functionality as defined in Figure 4.8.



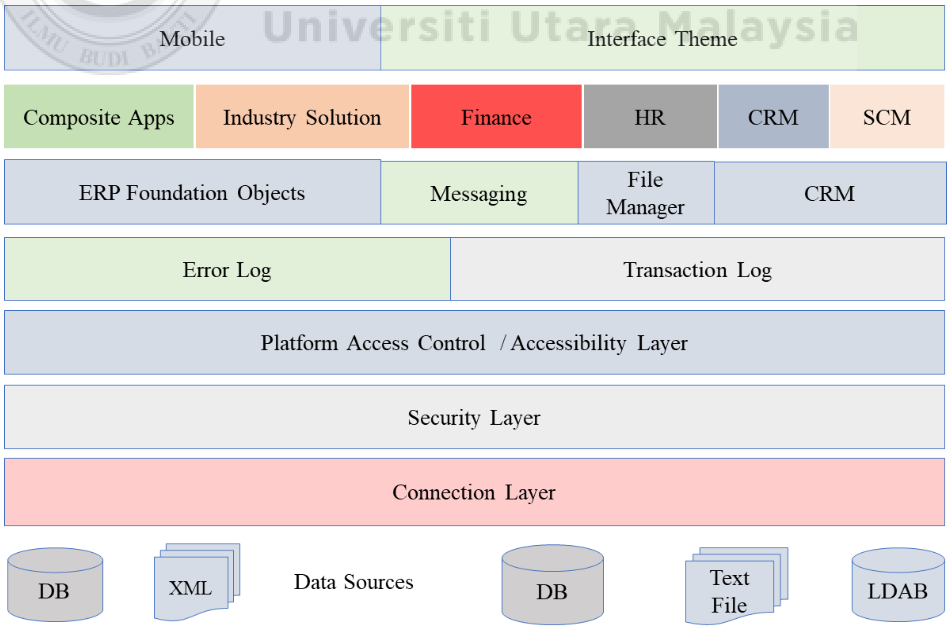



*Figure 4.8* Software Architecture Platform Example

From the top are the user interface layer, system business object layer, foundation and built-in services layer, transaction and error log layer, access control layer, security layer, data connectivity layer, and data source layer. Each layer has its own role, as defined in Table 4.7.

Table 4.7
*Software Architecture Platform Layers*

| **Layer** | **Functionality Description** |
|---|---|
| User Interface Layer | System user interface using browser, mobile interface. |
| System Business Object Layer | System object connecting user interface layer to the database. Each system object should provide services using transaction code for each event. |
| Foundation and Built-In Services Layer | Common services defined by platform accessible to all system and built-in services. |
| Transaction and Error Log Layer | Records all successful transaction events and failed transactions into error logs. |
| Access Control Layer | User accessibility services to manage user control. |
| Security Layer | User defined security measures in terms of connectivity protocol. |
| Data Connectivity Layer | Manage data source accessibility, scripts and passwords connecting system to the data source layer by creating connectivity ID for programmer's reference. |
| Data Source Layer | Manage database, text file, XML files connectivity by giving specific ID to each data source. |

#### 4.4.2.2.3 Backup / Redundancy

As Cloud ERP manage critical data for any organization, our CEF Model recommends a built-in data redundancy model to be integrated within the Cloud ERP platform. This can be achieved by having multiple instance of system and data distributed across multiple location. The system should provide real time replication between the primary and secondary system. In the event of disaster, the fail over process can be done by making the secondary system as the production system. The primary system will be the publisher and the secondary system will be the subscriber as shown in Figure 4.9.

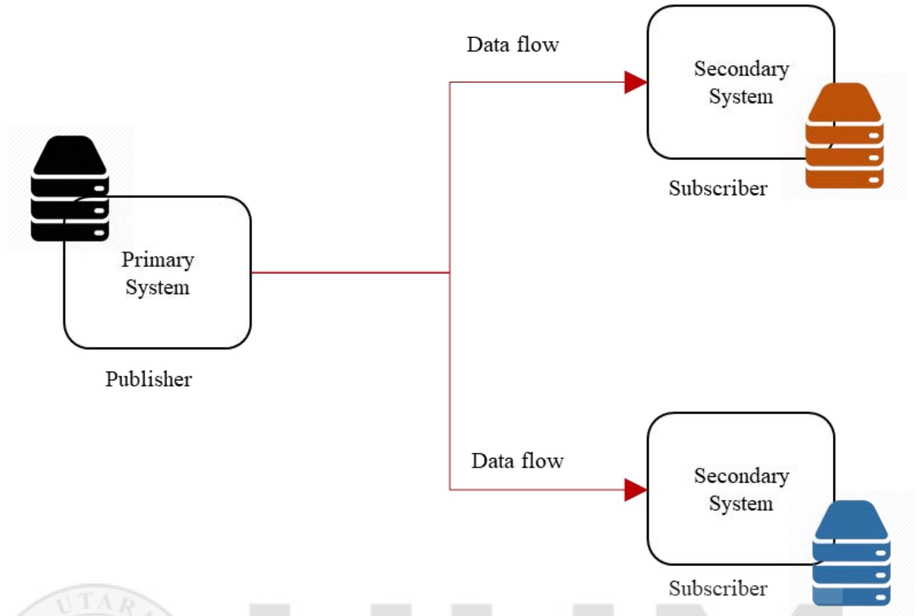



*Figure 4.9* Example of Cloud ERP Redundancy Architecture

#### 4.4.2.3 Platform Design Guidelines

Table 4.8 shows the design guidelines for the CEF Product component.

Table 4.8

*CEF Platform Design Guidelines*

| No. | Guidelines | Model Element |
|---|---|---|
| 1 | User portal platform should be able to host all Cloud ERP modules. | User Portal Platform |
| 2 | Developer platform should clearly separate all layers for scalability. | System Architecture Platform |
| 3 | System security and redundancy should be part of the overall system security built in features. | Backup / Redundancy |
| 4 | Each platform's functionality should work as a service object within the platform's layer. | System Architecture Platform |
| 5 | All system modules should define their services with transaction codes. | System Architecture Platform |

Table 4.8 continued

| | | |
|---|---|---|
| 6 | All transaction should be captured in the transactions log. | System Architecture Platform |
| 7 | All transaction services should define possible error logs. | System Architecture Platform |
| 8 | All system modules should provide debug features. | System Architecture Platform |
| 9 | Platform should provide multi-tenancy control for all system modules. | User Portal Platform |
| 10 | All transaction must go through security layer object. | System Architecture Platform |
| 11 | All transaction must go through data connectivity layer object. | System Architecture Platform |
| 12 | Define audit methods. | NA |

#### 4.4.2.4 Platform Summary

In summary, the Platform Architecture should be able to provide a structured method for the products developed by the factory. The implementation for the CEF Platform model can greatly improve the problems with security concerns within the Cloud Computing deployment method. Each element contributes to the ability of the platform to handle all the products as well as the security element needed for the platform to be secured.

### 4.4.3 CEF Workflow

Workflow model for the Cloud ERP Factory basically simulates a standard manufacturing process flow as well as the core adaptation to Software Factory approach. Work center 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 CEF engine, as each task would be defined with specific checklist of tasks to maximize the available resources within the company to leverage on resource-based theory. Figure 4.10 shows an example of a Workflow management within a Cloud ERP production environment.

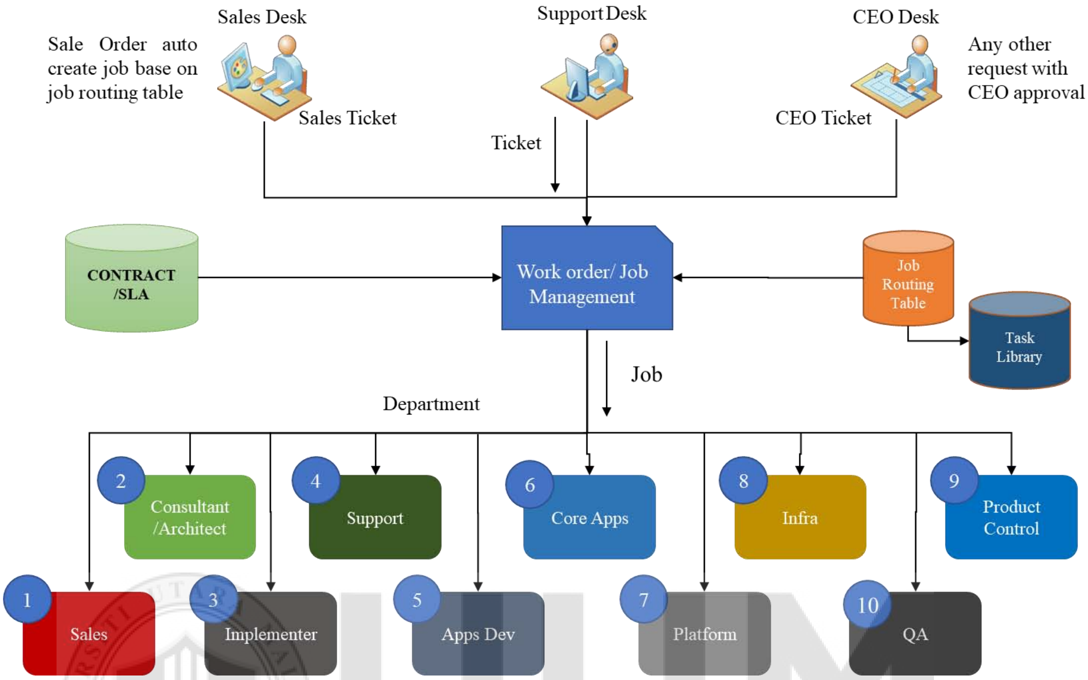



*Figure 4.10* Workflow Management Architecture



### 4.4.3.1 Workflow Component Objectives

The following are the objectives:

i. To promote standardized operating procedures to handle each task.

ii. To promote segregation of processes for more efficient execution.

iii. To allow better estimation of time frame to complete specific tasks.

### 4.4.3.2 Workflow Core Model Elements

In the CEF Workflow, there are five core model elements, which are System Input, Work Center, Job and Task Library, Work Order Routing, and System Output. Each of the core model elements plays an integral part in ensuring the Workflow component is properly defined.

#### 4.4.3.2.1 System Input

For the CEF Workflow Model, system input is a very important core model element as it identifies the type of input accepted by the Cloud ERP Factory Production environment as shown in Table 4.9. Each system input carries its own importance and is tracked by a request number. The request number is used throughout the whole process as a point of reference until it is solved or completed. The Figure 4.11 shows an example of a system input flow chart in a Cloud ERP Production floor.

Table 4.9

*System Input Description*

| Sales Desk | Support Desk | Other Input Request |
|---|---|---|
| New sales request/New sales order request | After-sales support contract | Internal request and R&D activities |



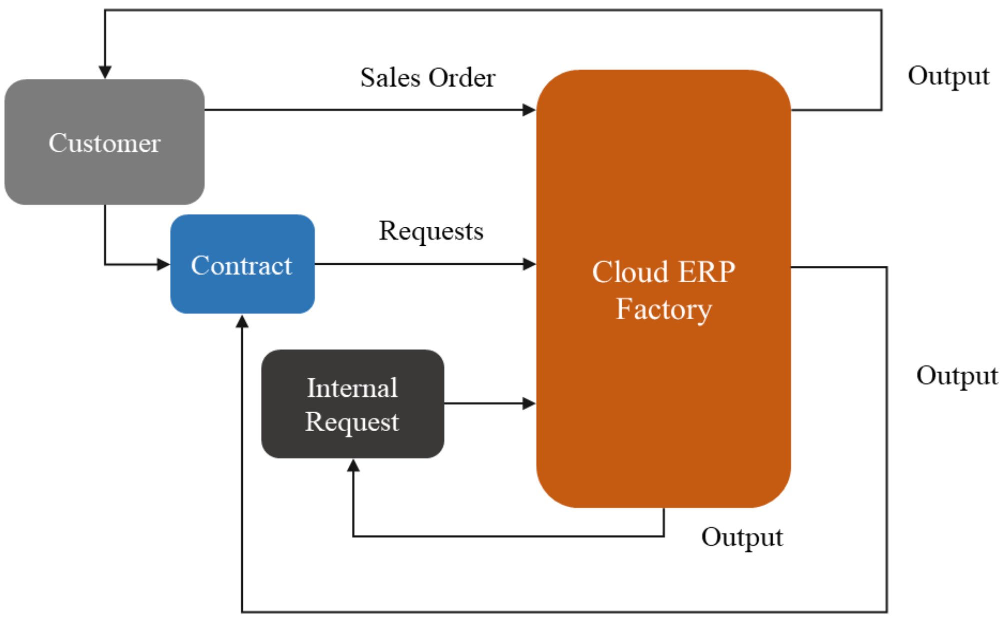



*Figure 4.11* System Input Workflow

#### 4.4.3.2.2 Work Center

Each work center represents specific task with specific skills required to execute the task. In a CEF Model, work center is primarily a department, which provides a specific set of services, which is defined in the task library. Figure 4.12 illustrates a sample of Work Centers in a typical Cloud ERP Factory production. Each department provides different set of tasks and services according to the recipe decided for a specified product. The definition of each department can be varied according to the organization.

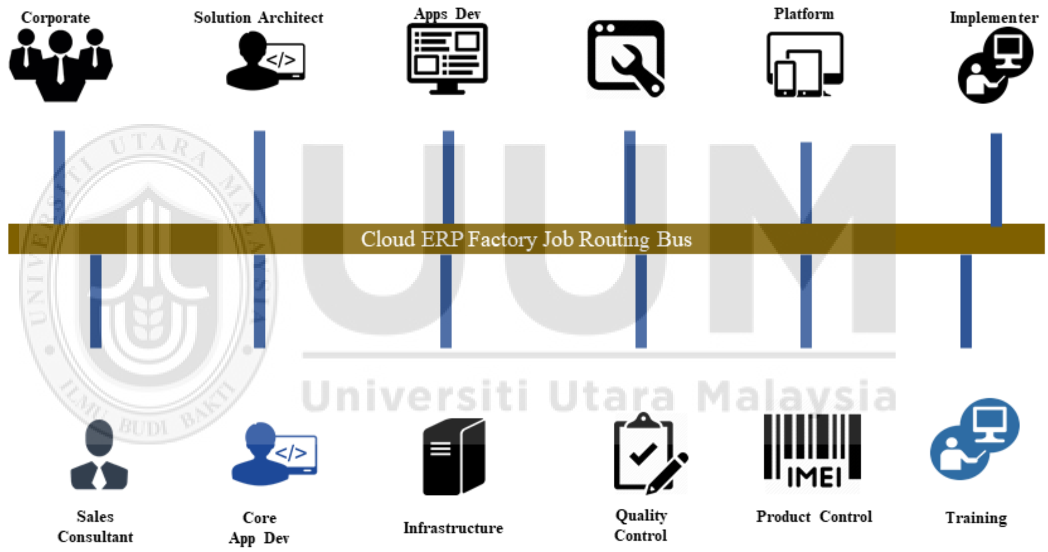



*Figure 4.12* Typical Work Center in CEF

#### 4.4.3.2.3 Job and Task Library

Every single job in the CEF Model must be clearly defined with the specific tasks required to be executed. A job may have many tasks in order to complete that specific job as shown in Figure 4.13. All of this information would then be stored in a Task Library.

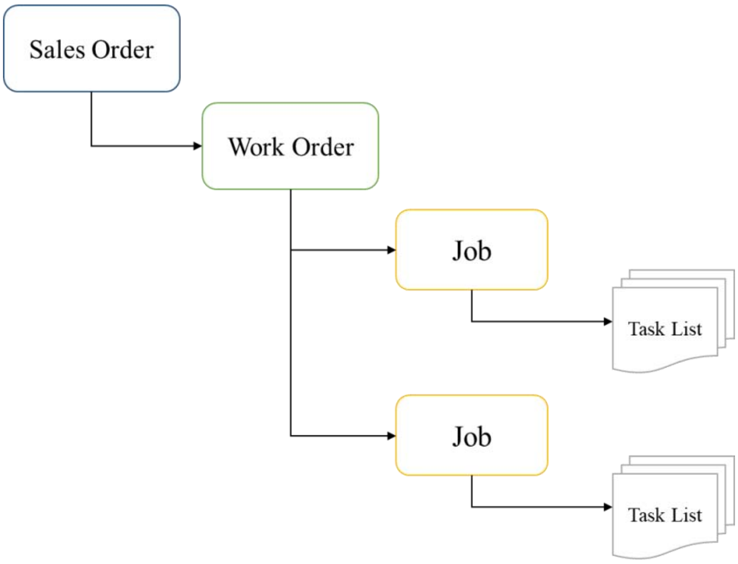
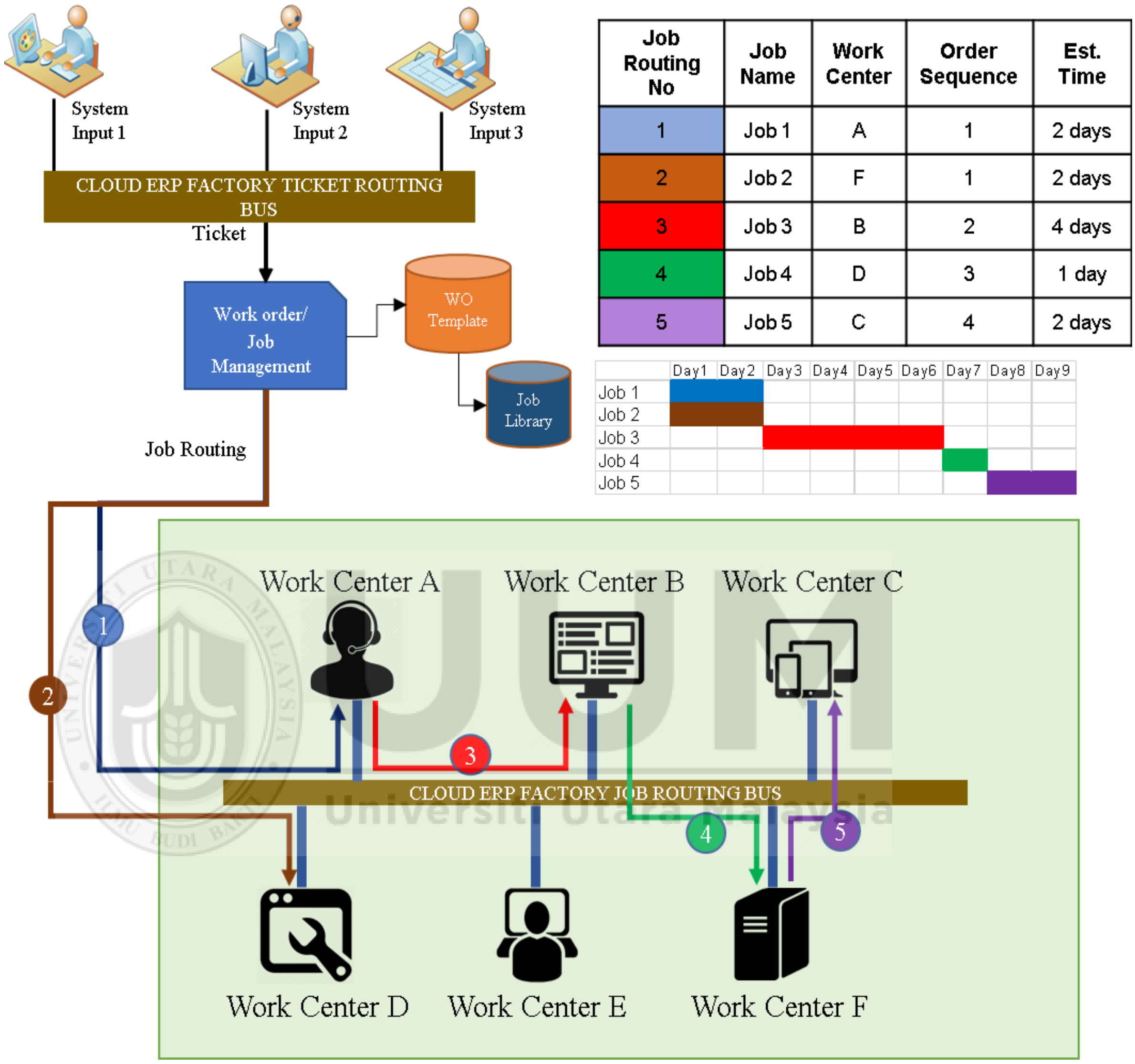



*Figure 4.13* Job and Task Definition in Job Library

#### 4.4.3.2.4 Work Order Routing / WO Template

As with conventional manufacturing processes, the CEF Job Workflow is defined by Work Order routing. In manufacturing convention, product process routing is a process flow involved to build a particular product with BOM as material required. In our CEF Model, the term Work Order routing refers to processes required to complete a particular Work Order. A routing template is used to pre-define the processes involved and their routing or sequence in completing a particular Work Order. Basically, a routing template consists of a list of processes and their sequence to be executed. When any routing template is used, the system will assign the required processes to the respective Work Centers or departments and each process will be referred to as a job.

Based on Figure 4.14, an example of how CEF Workflow model is shown. The ticket generated from one of the system input is converted into a Work Order, which contains a Work Order template complete with its defined job routing table. Each job is then distributed to the assigned Work Center according to the order sequence. Once every single job is completed, the ticket will be updated and closed.




| Job Routing No | Job Name | Work Center | Order Sequence | Est. Time |
|---|---|---|---|---|
| 1 | Job 1 | A | 1 | 2 days |
| 2 | Job 2 | F | 1 | 2 days |
| 3 | Job 3 | B | 2 | 4 days |
| 4 | Job 4 | D | 3 | 1 day |
| 5 | Job 5 | C | 4 | 2 days |

*Figure 4.14* Sample Work Order Routing Template

CEF Workflow can be configured to be either fully automated or semi-automated.

i. Semi-automated – In this mode, from the Work Order template, which contains list of jobs, each job will be routed or assigned to the particular Work Center. The Work Center administrator or Head of Department has the option to delegate to job to the member attached to the Work Center. In this mode, the CEF time block is recommended to be set up in day block.

ii. Fully automated – In order to enable this mode, it is required to set up the CEF time block with more accuracy. It is basically the most refined time scale in a 24 hours' period. Typically, 30 minutes can be assigned as the time block, which means that the scheduling of jobs for a particular Work Center will be in the 30-minute scale. Here, the estimated time to complete the job will be used by the system to slot the particular job to the available workers with the right competency who are attached to the Work Center. In summary, in fully automated mode, jobs will be assigned directly to the staff based on the availability and skill level. A few other parameters, such as time between job and working hour patterns can be configured in order to achieve optimum productivity level.



Figure 4.15 illustrate a typical Work Order Routing flow in a Cloud ERP Production environment that would support the CEF Model.

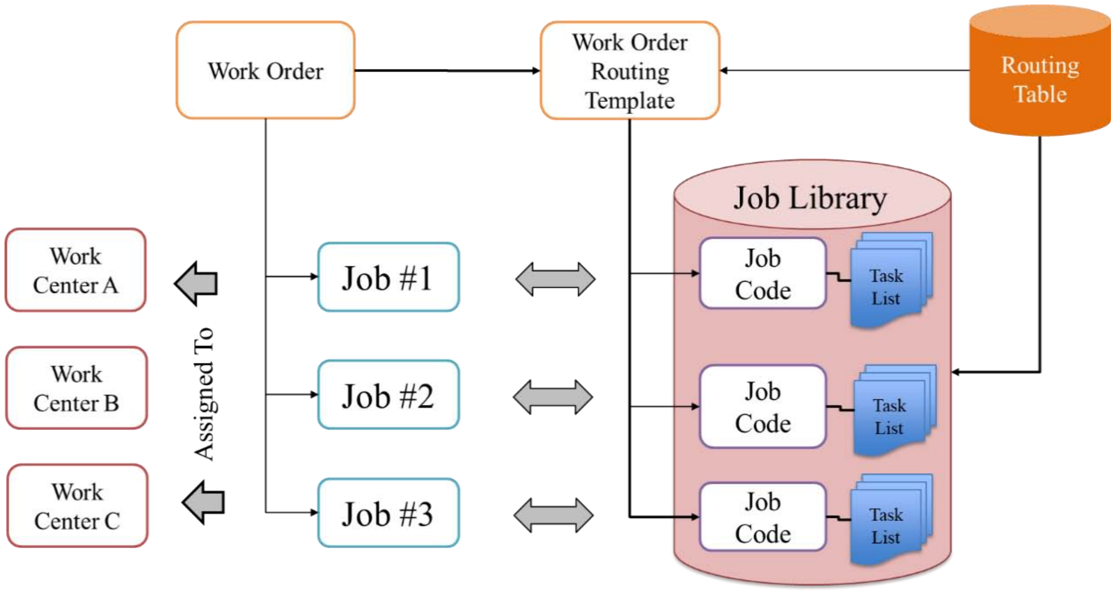



*Figure 4.15* Work Order Routing Template Example

#### 4.4.3.2.5 System Output - CEF Job Dashboard

CEF Job Dashboard can be considered as the planning output of the system and serves as a planner in the manufacturing plant. In a manufacturing plant, the production planner will plan for the optimum capacity to achieve maximum productivity. Here, the system can be configured to automatically take the system input requests and translates them into multiple jobs displayed in the CEF Job Dashboard of the respective assignees.

Figure 4.16 explains how the employee is assigned with several jobs. Using a CEF user dashboard, the employee is able to refer to a specific set of job checklists on how the task should be executed, which is cross referenced with the LMS system.

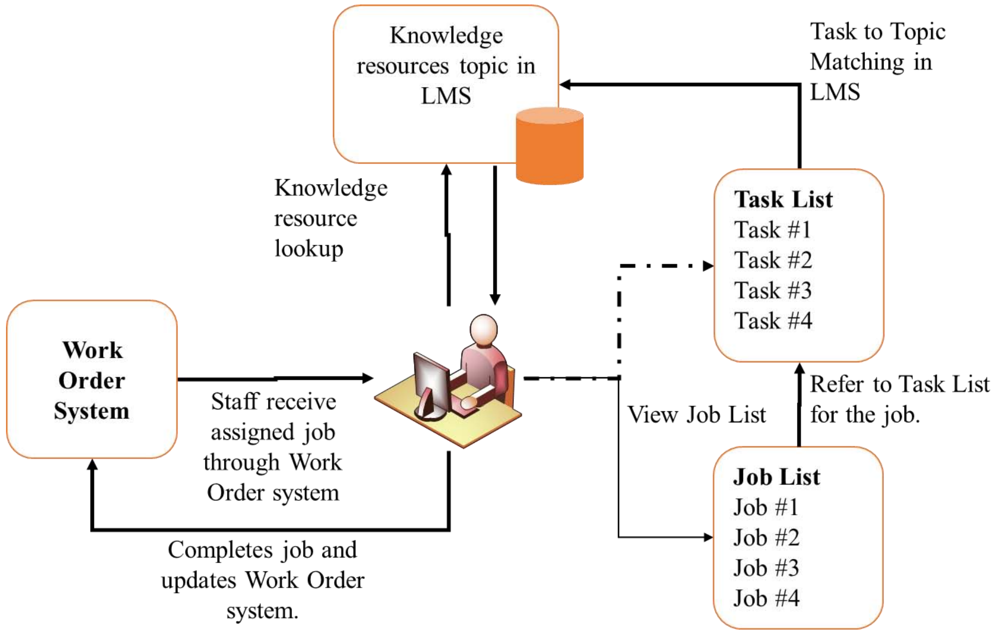



*Figure 4.16* Job Activity Update using CEF User Job Dashboard

### 4.4.3.3 Workflow Design Guidelines

Table 4.10 shows the design guidelines for the CEF Product component.

Table 4.10

*CEF Workflow Design Guidelines*

| No. | Guidelines Description | Model Element |
|---|---|---|
| 1 | Define Work Centers to provide specialized services within the factory process. It can also be referred to as Departments. | Work Center |
| 2 | Define system inputs into the CEF Workflow model. | System Input |
| 3 | Define work order, job, and task library into a library of services. | Job and Task Library |
| 4 | Define time block to represent the minimum time scale for the job allocation slot. | Job and Task Library |
| 5 | Define competency skills required for each job. | Job and Task Library |
| 6 | Define the task list for all specified jobs. | Job and Task Library |
| 7 | Define Work Order template to represent specific routing for all common input types. | Work Order Routing Template |
| 8 | Define priority level for each Work Order template. | Work Order Routing Template |
| 9 | Create a system as per the CEF Workflow requirement. | NA |

#### 4.4.3.4 Workflow Summary

In summary, the Workflow model could be one of the most crucial components as it dictates the overall process to be completed within a designated Work Center. The elements within the Workflow Model provides the necessary tools to achieve a semi-automated environment in a Cloud ERP factory production. The automated environment can definitely promote the reduction of workload as well as maintaining a quality standard as each job is defined with precise process flows and maximizing the capacity of the resources available. The five core model elements for CEF Workflow model define the way to configure and operate the operational workflow of the Cloud ERP Factory. The model only defines general specifications guidelines. Obviously, the CEF Workflow model have to be translated into a system that acts like a manufacturing planning system in a conventional factory. The development of the CEF Workflow system will be demonstrated in Chapter Five.

### 4.4.4 CEF Product Control



Product Control is another significant component of the CEF Model. It covers the Cloud ERP product management record and its versioning control. Generally, it is a quality control mechanism of the Software production. CEF Model Figure 4.17 shows the overview diagram of Product Control workflow recommended to be used in the CEF Model.

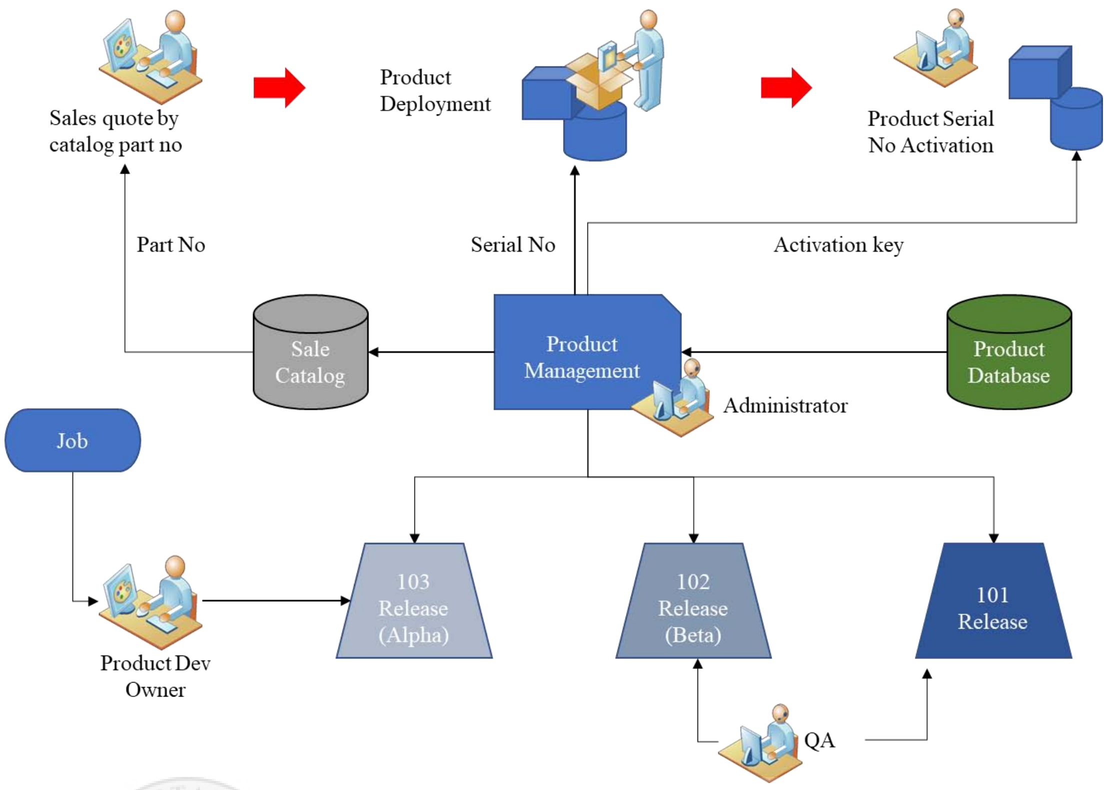



*Figure 4.17* Product Control Workflow Sample

#### 4.4.4.1 Product Control Component Objectives

i. To provide software product quality control within the Cloud ERP Factory.

ii. To manage versioning control over the software products.

iii. To provide anti-software infringement using serial number and activation control.

#### 4.4.4.2 Product Control Core Model Elements

Product Control Model consists of five core model elements. The elements are System Identification, Version Control, Release Control Model, Software Update, and Anti-Software Infringement Control.

#### 4.4.4.3 System Identification

Every product should have its own unique identification number. In a traditional manufacturing practice, this is referred to as a manufacturing part number. In the context of CEF Model, Part number represents the different type of product in a product group family. The part number convention should be defined clearly as not to create any ambiguity between the product family. Part number convention for a software product might differ from a normal manufacturing part number as it requires a more detailed elaboration of its part number.

Figure 4.18 illustrates how a software product part number convention can be decided. Each segment denotes the specific information that is able to identify the product.

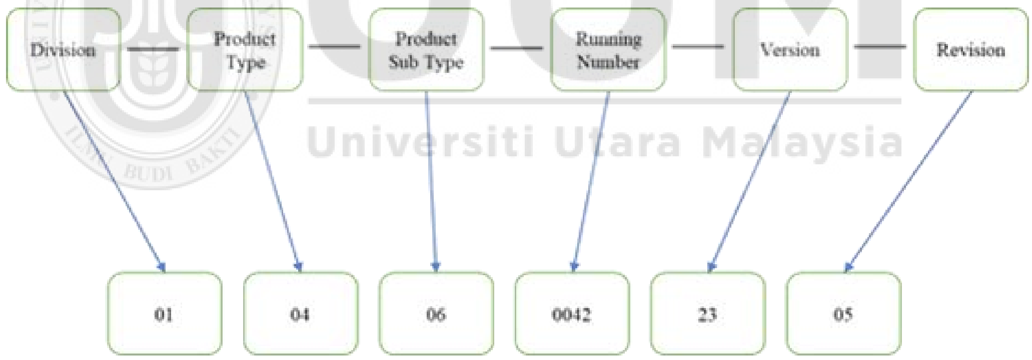



*Figure 4.18* Part Number Convention Sample

Figure 4.19 shows how a new part number is issued when a new system has been requested by a customer. Before the development of the system, a part number is generated through the Product Management module.

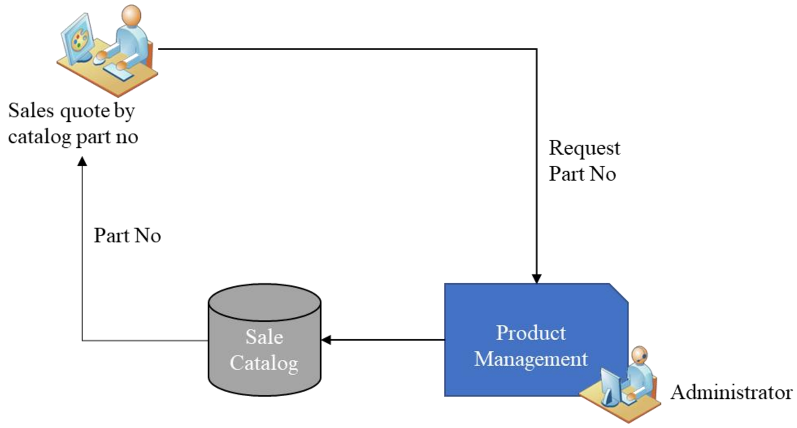



*Figure 4.19* Part Number Request Model

#### 4.4.4.3.1 Version and Revision Control

Version and Revision represent a quality control mechanism over the Cloud ERP software release updates. In our CEF Model, revision is referred to as incremental changes update to a particular version of a module, while version is a variation of software from the original standard version. For example, if a standard module is released with version 1.0, a variation of non-standard or customized version will be referred to as version 1.1. On the other hand, revision is referred to incremental progressive improvement or updates for a particular version of a software module. This example is shown in Table 4.11.



Table 4.11

*Version and Revision Sample*

| Part Number | Version | Revision | Description |
|---|---|---|---|
| 5001 | 00 | 00 | Standard Version Revision 00 |
| 5001 | 00 | 01 | Standard Version Revision 01 |
| 5001 | 00 | 02 | Standard Version Revision 02 |
| 5001 | 01 | 00 | Variation of Standard Version Revision 00 |
| 5001 | 01 | 01 | Customized Version Revision 01 |

#### 4.4.4.3.2 Release Control Model

Another part of quality measure within CEF Product Control is Release Control procedure. As CEF Model should be able to work in public Cloud ERP as well as private and hybrid Cloud. Based on that, the assumption is made that the server can reside with the Cloud ERP provider or at the customer's site. However, the production should be interconnected with the product management server. Software product is assumed to be release into product management server before it can be deployed into the production server regardless of location. Therefore, this section tackles the release procedures of the software to the central product management server. Figure 4.20 is an example of a typical new version software release procedures:

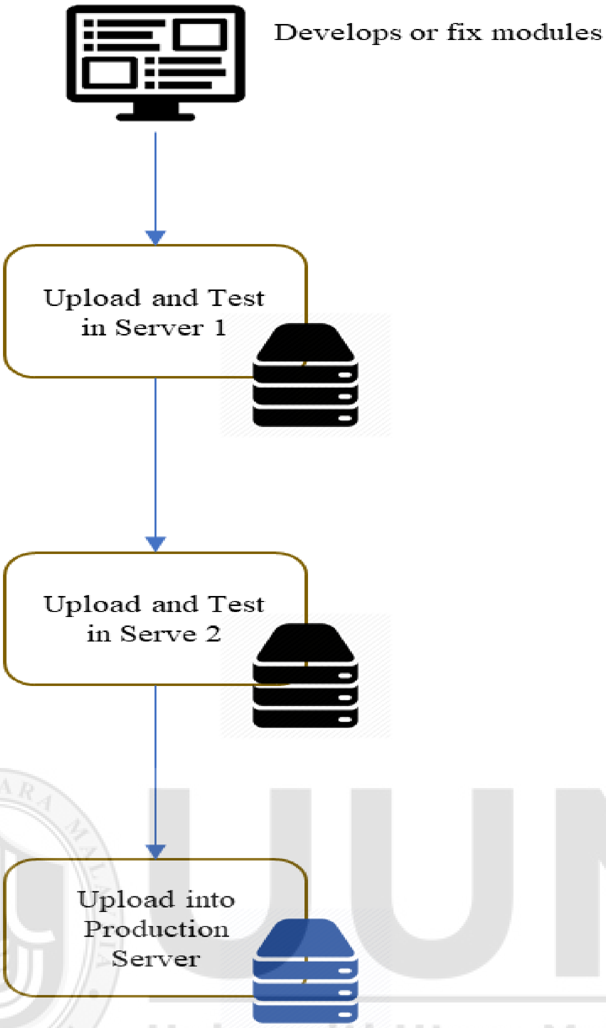





*Figure 4.20* New Software Release Process Flow

**4.4.4.3.3 Software Updates**

Traditional software update requires the provider to personally oversee the process to ensure that the required updates are applied. In Cloud ERP production, the methodology of distributing software update differs by means of allowing clients to dictate what and when the product should be updated. Two methods of online marketplace where clients are able to access and download the necessary updates to be installed are suggested, as shown in Figure 4.21 and 4.22.

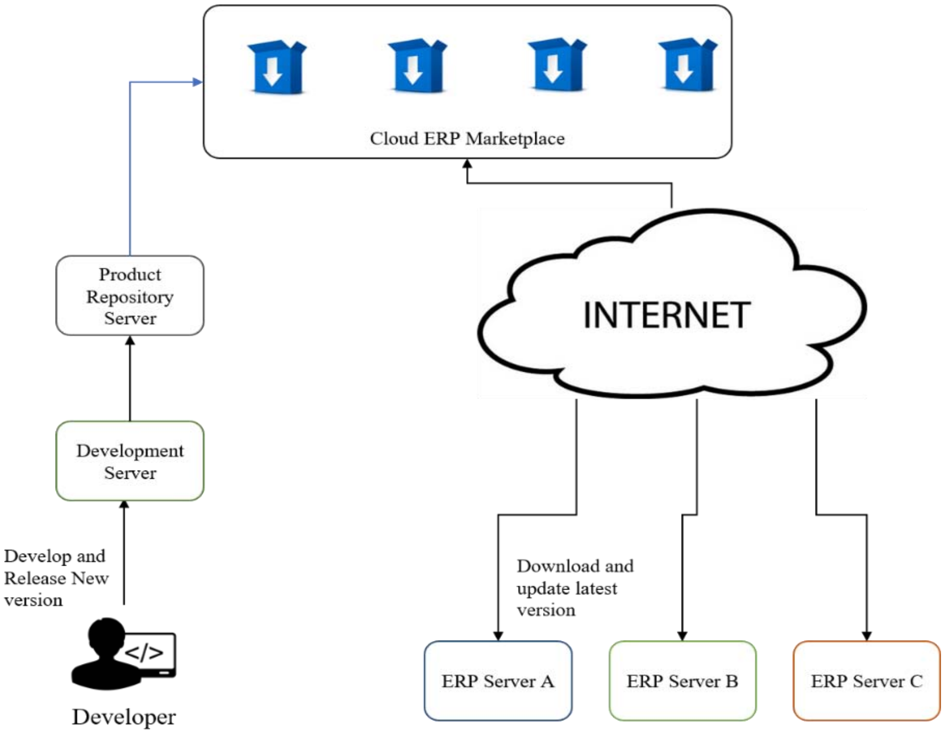



*Figure 4.21* Software Update Method A



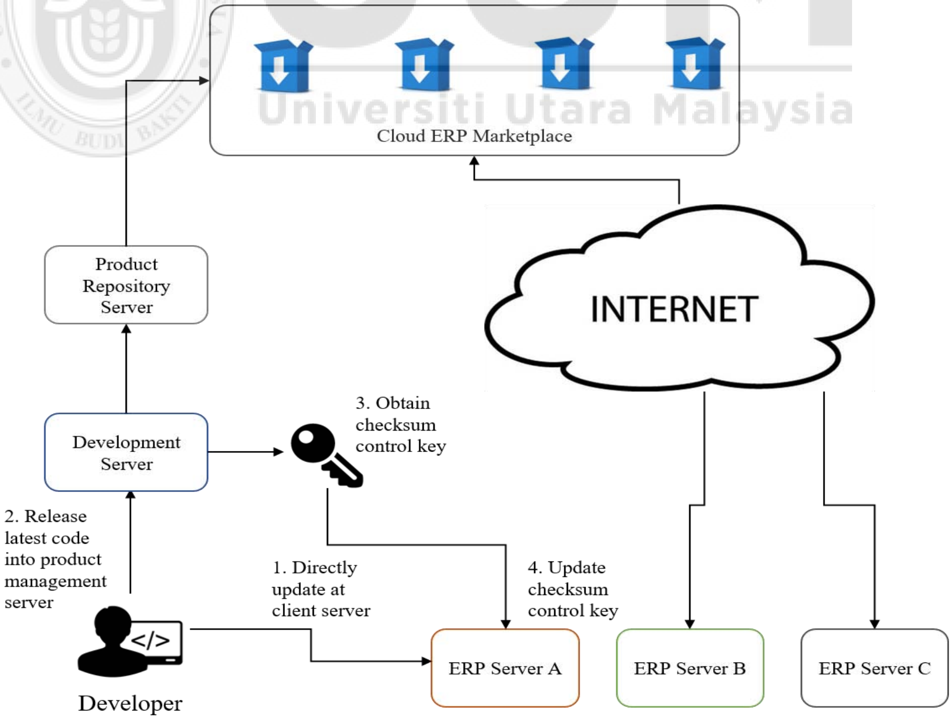



*Figure 4.22* Software Update Method B

#### 4.4.4.3.4 Anti-Software Infringement Control

In order to avoid software illegally being deployed, a serial number is generated for every software that has been issued. The serial number acts a license key for the software to be legally operated. However, after the software has been installed to the intended Cloud server, the system will prompt for an activation key, which will be authenticated through the product management system. The activation key as being commonly deployed in the industry, make use of a combination of the serial number and the server hardware ID to generate the activation key with a specific algorithm or formula. The process can be done manually or directly linked over the Cloud. Figure 4.23 shows the overall process flow to ensure that software is protected from being illegally deployed.

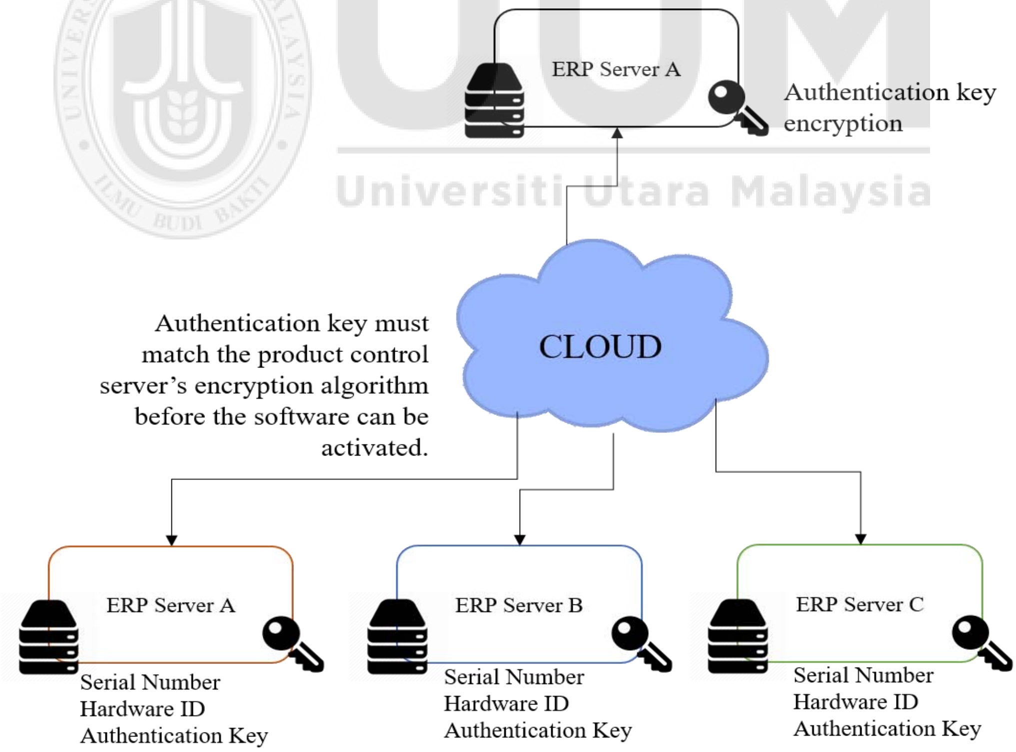





*Figure 4.23* Anti-Software Infringement Method

While a part number differentiates the type of products in the CEF Model, a serial number is a series of running number that helps to keep track of the number of the same product being deployed as per shown in Figure 4.24. The procedures of obtaining a serial number is similar to the generation of part number explain previously.

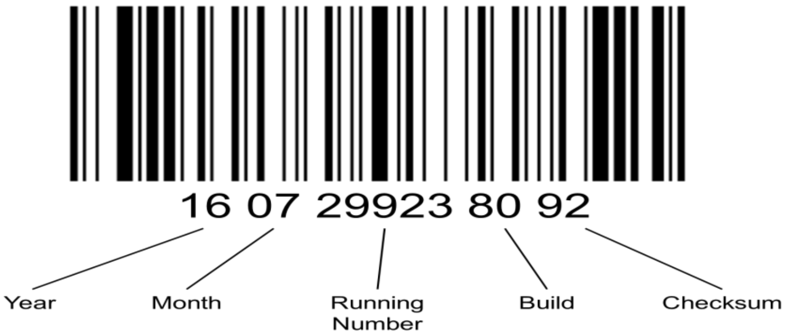



*Figure 4.24* Serial Number Convention Example

#### 4.4.4.4 Product Control Design Guidelines

Table 4.12 shows the design guidelines for the CEF Product Control component.

Table 4.12

*CEF Product Control Design Guidelines*

| No. | Guidelines | Model Element |
|---|---|---|
| 1 | All product registered in the system have its own unique part number. | System Identification |
| 2 | The product should go through several levels of different quality control to ensure the product quality. | Version and Revision Control |
| 3 | Documentation for each registered product is a must. | Release Control Model |
| 4 | Each product developed will go through different release phases which are Alpha Release and Beta Release stage. | Release Control Model |
| 5 | A centralized mechanism is required to hold all information of the product released and the status of the product. | System Identification |
| 6 | Define part number convention. | System Identification |

Table 4.12 continued

| | | |
|---|---|---|
| 7 | Define serial number convention. | System Identification |
| 8 | Define version and revision within the part number convention. | Version and Revision Control |
| 9 | Define a mechanism of inconsistent update proof. | Release Control Model |
| 10 | Define software update check-out control protocol. | Software Updates |
| 13 | Define software release procedures. | Release Control Model |
| 14 | Define software testing methodology. | Release Control Model |
| 15 | Define quality assurance mechanism. | Release Control Model |
| 15 | Create a system as per the CEF Workflow requirement. | NA |

#### 4.4.4.5 Product Control Summary

In summary, the CEF Product control model provides the quality assurance mechanism over the overall CEF process flow. Each model core elements strengthens the ability of the CEF Model to keep track of the type of products being developed and deployed. This indefinitely enhances the CEF Model objective to solve the issue of inconsistent software quality being developed.

### 4.4.5 Knowledge Management

The last but most important component of CEF Model is the knowledge management model. This component oversees the knowledge gathering, dissemination, and retention aspect of CEF Model as shown in Figure 4.25. This model is heavily related to the concept of sustainability of a business entity.

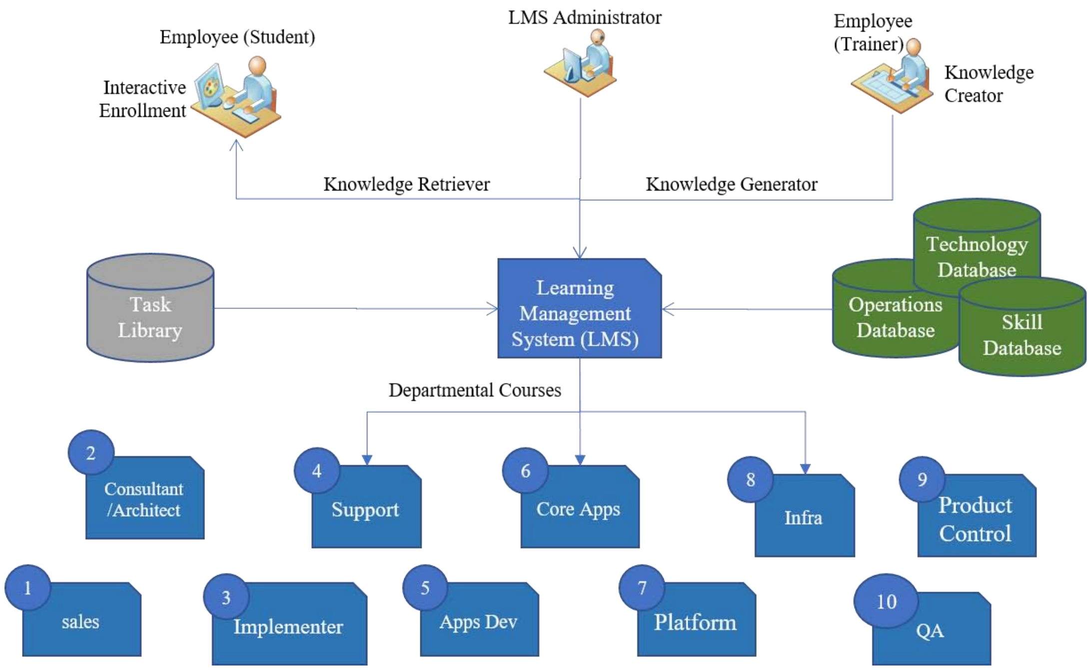



*Figure 4.25* Knowledge Management Example



#### 4.4.5.1 Knowledge Management Component Objectives

i. To provide model of knowledge repository within the CEF.

ii. To provide dynamic knowledge retrieval and dissemination.

iii. To provide dynamic knowledge update or gathering.

#### 4.4.5.2 Knowledge Management Core Model Elements

The core model elements for the Knowledge Management subcomponent are identified as Knowledge Identification and Segmentation, Knowledge Retrieval and Dissemination, Dynamic Knowledge Logging, and Knowledge to Job/Task Mapping. The combination of every single element would solidify the Knowledge Management Model so that every single information that obtained or learned remains within the system for further reference.

#### 4.4.5.2.1 Knowledge Identification and Segmentation

Knowledge identification and segmentation is required as to establish a base of the types of knowledge which will be stored in the repository. For the CEF Model, three types of knowledge have been identified, which are Technology Knowledge, Operational Knowledge, and Skill Knowledge as shown in Figure 4.26.

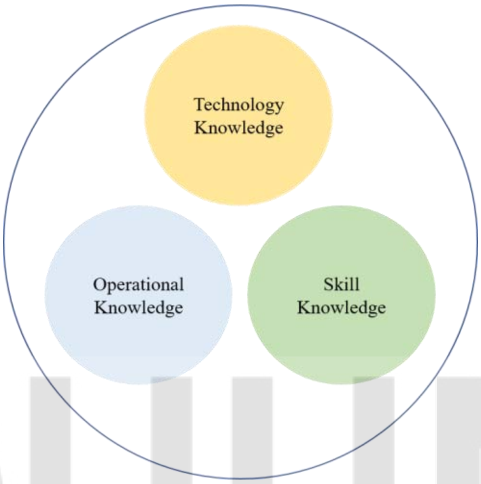



*Figure 4.26* Type of Knowledges



i. Product, Project, and Business Process Knowledge

To understand how to properly retain the knowledge obtained in the organization, each type of knowledge needs to be identified and categorized accordingly. Every product is required to have its own information portfolio, providing reference to both the current and future employees. This is would be very helpful with new hires to be acquainted with the products developed in the Cloud ERP production.

ii. Technology Knowledge

Technological knowledge in a very knowledge dependent environment is one of the most important factor to consider as retaining the technological know-how translates into business leverage. In the scope of CEF Model, the technological knowledge should always be considered top priority, as Cloud computing advancements are being made daily; thus, in order to capitalize on the situation, it should be acquired and retained as quickly as possible.

iii. Skill Knowledge

In every single type of working environment, a skill set or competency can either be learned or acquired through training. The type of skills needs to be identified clearly depending on the type of products developed as well as the services provided by the organization.

#### 4.4.5.2.2 Knowledge Retrieval and Dissemination

As more and more knowledge is being catalogued and stored, the CEF Model would need to take into consideration the method of how to retrieve and disseminate the knowledge to the employees. This core model element looks into the actual process flow, connectivity between knowledge repository, and how to retrieve the information for employee's usage as shown in Figure 4.27.



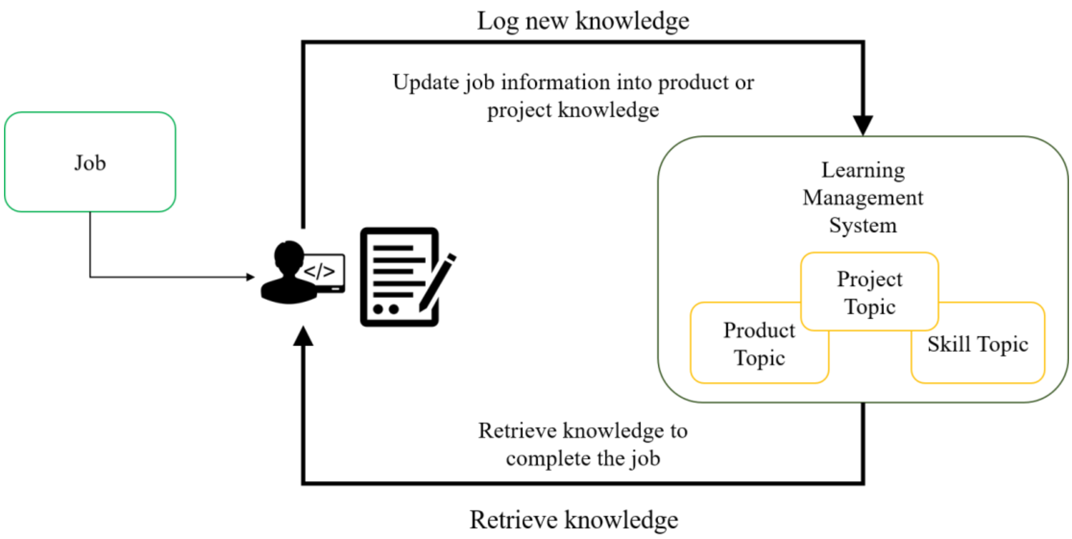



*Figure 4.27* Knowledge Retrieval and Dissemination

#### 4.4.5.2.3 Dynamic Knowledge Logging

Every single type of knowledge amassed during a software development lifecycle or a particular project implementation should be logged dynamically. This activity allows for the population of knowledge pertaining to a specific product, a project and company's standard operating procedures. Every information, such as training materials, client's user requirements, presentation slides, and actual application codes should be logged and archived. The objective in the long run would be to empower any new employee with the information needed to be able to support the assigned project or product.

#### 4.4.5.2.4 Knowledge to Task/Job Mapping

This core model element focuses on the mapping of each product, service, project, and even company standard operating procedures to a specific course. This in turn would allow for each course to be broken down into multiple topics, which contains all of the associated materials with it. As Figure 4.28 shows, each course can be connected to several topics and each topic would then have its own specific elements.

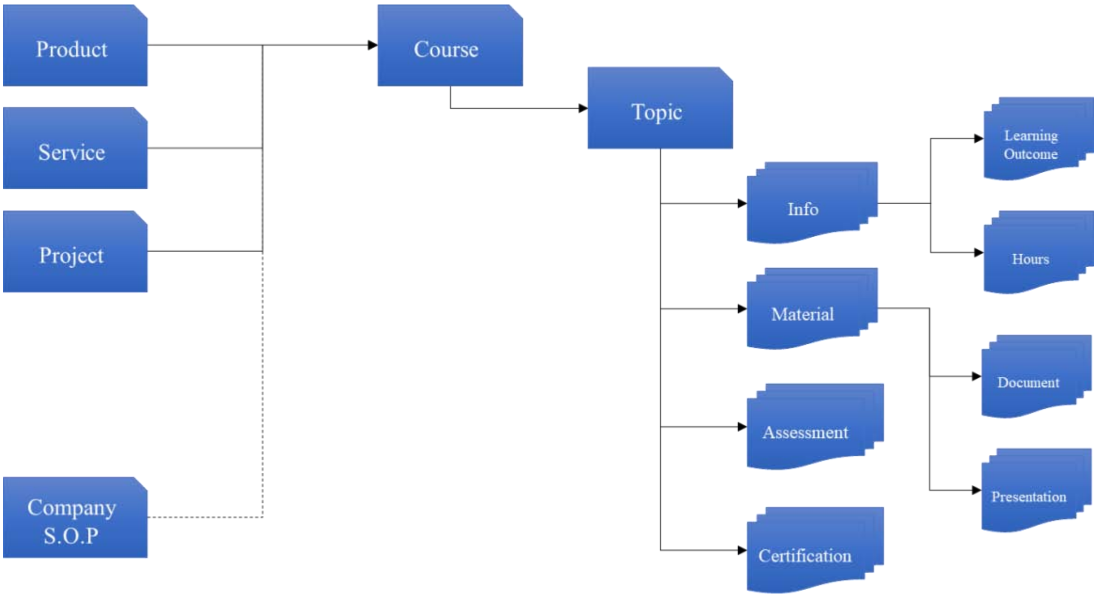



*Figure 4.28* Mapping of Product to Tasks

### 4.4.5.3 Knowledge Management Design Guidelines

Table 4.13 shows the design guidelines for the CEF Product Control component.

Table 4.13

*Knowledge Management Design Guidelines*

| No. | Guidelines | Model Element |
|---|---|---|
| 1 | Create courses with topics relevant to the jobs/tasks. Maps courses and topics to the predefined jobs and tasks. | Knowledge to Task/Job Mapping |
| 2 | Define skill tasks, knowledge tasks and operational tasks. | Knowledge Identification and Segmentation |
| 3 | Define all effected users in the model. | Knowledge Identification and Segmentation |
| 4 | Define method of knowledge accessibility within the assigned job parameters. | Knowledge Retrieval and Dissemination |
| 5 | Define the new knowledge logging procedures. | Dynamic Knowledge Logging |
| 6 | Create a system as per the Knowledge Management model requirement. | NA |

#### 4.4.5.4 Knowledge Retention Management Summary

In summary, knowledge management is very important for any knowledge-based organization, as it holds the key for business sustainability and knowledge retention management. By implementing this subcomponent, the CEF Model is able to cultivate a culture of peer-to-peer knowledge sharing and the retention of the knowledge in the case of staff turnover.

## 4.5 Model Validation by Expert Review

This model validation by expert review was gathered with formulated questionnaires to address the proposed CEF Model from selected domain experts. Each model is validated by expert reviews consisting of industry experts selected from various practitioners and corporate users with good subject domain knowledge. Five practitioners from cloud based ERP developer companies, corporate users with Cloud ERP expertise, and two experts from the academia were involved with the review. The rationale behind the selection of the expert reviewers is explained as:

i. More than ten years of experience in software development industry or academia and has professional certification in computer science or software engineering.

ii. Experienced in managing sizeable software development projects (Applicable to industry experts).

iii. Understands the challenges faced by software production environment, especially in Cloud ERP.

iv. Has direct or indirect experience in Cloud ERP project.

Table 4.14

*List of Reviewers Selected for Expert Review*

| No | Gender | Position | Field of Expertise | Experience (Year) | Affiliations |
|---|---|---|---|---|---|
| 1 | Male | Academician | Manufacturing System | 25 | UniKL |
| 2 | Male | Software Support Manager | Computer Science | 22 | Dell Corporation, USA. |
| 3 | Male | Chief Executive Officer | Software Engineering | 22 | Datamover, Software Development company |
| 4 | Male | IT Manager | Computer Engineering | 22 | PT Siam-Indo Gypsum Industry |
| 5 | Male | IT Manager | Computer Science | 20 | Government Agency |
| 6 | Male | IT Manager | Software Engineering | 20 | Barakah Offshore Berhad |
| 7 | Male | Academician | Software Engineering | 30 | UUM |



As shown in Table 4.14, the experts came from a diversified yet similar understanding in the software production environment. This similarity was crucial to ensure that the experts were able to establish review and provide quality constructive feedback on the proposed model.

Expert One is a PhD holder in manufacturing industry research. Familiar with the overall manufacturing processes, the expert is experienced in managing software development team for manufacturing system, which provides us with a good insight due to the manufacturing basis to our Software Factory approach. Expert Two has more than 25 years of experience in managing software system support for a multinational corporation based in the United States. Expert two has been working in a system environment using logging cases for customer requests and disseminate to create technical support jobs.

Expert three and Expert Four both have ten years of experience as a Chief Information Officer for an MNC and ten years of experience in managing a software production company. Expert Three is well versed in business intelligence and shop floor manufacturing system, making him the right candidate to understand and evaluate the model; whereas, Expert four handles multiple teams of software developers and outsourced vendors.

Our Expert five has been managing teams of software developers and outsourced software vendors in a government agency for nearly a decade and experienced in coding and database design. Expert Six is an IT manager for a large national oil and gas corporation that manages the overall software deployment for the organization. This is particularly useful in providing insights in terms of customer's expectations dealing with Cloud ERP deployment. As for the final expert, it is prudent to invite another software engineering expert, thus, Expert Seven is also a professor at an established institution of higher learning in Malaysia.

Due to the geographical limitation and time constraint, it is almost impossible to bring all the experts in a single review session. Therefore, the expert reviews were conducted separately at different time and location for each expert. In order to ensure consistency in the review environment, the process of conducting the review session has been standardized. All the reviewers were interviewed either face to face or through a video conference. The expert review questions were based on the nine quality model dimension questions which are Understandability, Clear Steps, Relevancy, Flexibility, Scalability, Accuracy, Timeless, Completeness, and Consistency (Batini et al., 2009). The expert review form is as per attached in Appendix A. The following Table 4.15 shows the results of the expert review that has validated the proposed CEF Model with a Likert Scale that is scaled with 1: Strongly Disagree, 2: Disagree, 3: Slightly Disagree, 4: Neutral, 5: Slightly Agree, 6: Agree, and 7: Strongly Agree.

Table 4.15

*Expert Review Results*

| No | Questions | Respondents Likert scale (1-7) n= 7 |
|---|---|---|
| 1 | The CEF Model is easy to understand. (Understandability) | 6.3 |
| 2 | The CEF Model provided a clear steps and procedures to follow. (Clear steps) | 6.2 |
| 3 | The CEF Model is relevant to the software production environment of Cloud ERP. (Relevancy) | 6.0 |
| 4 | The CEF Model is able to support the needs for future Cloud ERP production environment. (Flexibility) | 6.4 |
| 5 | The CEF Model is able to be implemented based on the needs of an organization without having to add additional resources. (Scalability) | 6.6 |
| 6 | The data flow of the CEF Model is reliable and accurate. (Accuracy) | 6.3 |
| 7 | The CEF Model describes a complete Cloud ERP production lifecycle which can be applied by other Cloud ERP providers. (Completeness) | 6.4 |
| 8 | The CEF Model is able to act as a consistent information source to allow for decision making support. (Consistency) | 6.1 |
| 9 | The CEF Model is able to be applied in the future and in different context. (Timeless) | 6.2 |

### 4.5.1 Expert Review Feedback

From the expert review validation of the CEF Model, most are in agreement that the CEF Model is able to be implemented as shown in Table 4.16. However, it is suggested that a performance-monitoring element should be included in the model, a detailed checklist for each subcomponent so that the model would be able to provide a more elaborate way to be replicated by other Cloud ERP providers. The checklist suggested by the expert reviewers were related to the type of verification checklist based on the findings by Maiani de Mello et al (2014).

Table 4.16

*Further comments from the experts on the proposed model*

| **Experts** | **Comments** |
|---|---|
| Expert 1 | • The model provides a good insight into process dimension mimicking manufacturing processes but the overall model scope is very big and can be enhanced over time.<br>• Should consider if the sequence for the subcomponent is required.<br>• Is there a self-checking steps to ensure that the output is viable? |
| Expert 2 | • The model looks interesting but requires explanation to be understood.<br>• Is it possible to provide a narrative explanation so that the model is self-explanatory?<br>• How to tune the priority of the jobs as suggested in the Workflow component? |
| Expert 3 | • The model is a good attempt to simulate Software Factory as a first attempt.<br>• It is a challenging endeavour to prove the model is workable.<br>• Should simplify the model to make it easily deployed. |
| Expert 4 | • The model is simple and readable.<br>• Yet to prove that the model is easily deployed in the real environment.<br>• How much time would it require to deploy this model?<br>• Should consider simplifying the design guidelines. |
| Expert 5 | • Understanding the model requires background knowledge in Software Engineering.<br>• Good attempt in solving the current real-world problem in software development arena.<br>• Suggest to include monitoring element in the Workflow component. |
| Expert 6 | • Model is sound and presents logic in creating a processed based software production lifecycle.<br>• Model provides insight as to create a process driven software production.<br>• The tool engine for the model is still a challenge to be developed.<br>• Model should include deployment steps. |

Table 4.16 continued

| | |
|---|---|
| Expert 7 | • Model is a logical representation of how to manage the overall software production.<br>• Possible to have a mechanism determining the completeness of model deployment.<br>• Design guidelines should be flexible.<br>• Combine the model with steps to how to apply it in real environment |

### 4.5.2 Refinement of the CEF Model

The feedback provided by the expert reviews was compiled and each of the constructive suggestion was taken into consideration to its usability of the CEF model.

Overall, most of the experts were in agreement that the model was a good attempt to solve the issues presented. Several experts pointed out that the model components were not in a sequence, which might lead to confusion as to how the model would be implemented. Expert Two and Expert Five both commented on the fact that although the CEF Workflow component is the engine of the CEF Model, a monitoring element is recommended to oversee and fine tune any job priority as proposed in the component. Expert Four and Expert Seven had the same idea with regards to the simplification of the design guidelines that were initially proposed. This, according to the experts, was to allow for more flexible model repeatability by other Cloud ERP production environments. After gathering the qualitative feedback provided through the expert reviews, the CEF Model was refined as listed out in Table 4.17. The refinement is also depicted in the next section.

Table 4.17

*Modification List on the Proposed Model*

| No | Before Refinement | After Refinement |
|---|---|---|
| 1 | All model components were not in sequence and were labelled in functional scope. | Subcomponents were numbered to denote the sequence of implementing the model and not required to have functional scope labeling. |
| 2 | Model component with required system development has no checklist as internal verification. | Defined and provided checklist as internal verification. |
| 3 | CEF Workflow component has no monitoring element. | Defined and included a performance monitoring element for the CEF Workflow component. |
| 4 | Design guidelines for some of the model components were not necessary. | Removed the unnecessary design guideline items for some of the model components. |
| 5 | Model does not provide steps on how to deploy it in Cloud ERP production environment | Added deployment steps to the finalized CEF model. |



## 4.6 Finalized CEF Model

The CEF Model is enhanced and improved into a finalized model based on the expert reviews and feedback primarily for the inclusion of a detailed checklist for each subcomponent based on Figure 4.29 and proposed a performance monitoring element to be in place. After which, Figure 4.30 shows the changes made on the previous proposed model for the more improved and finalized Cloud ERP Factory Model, which is then illustrated in Figure 4.31.

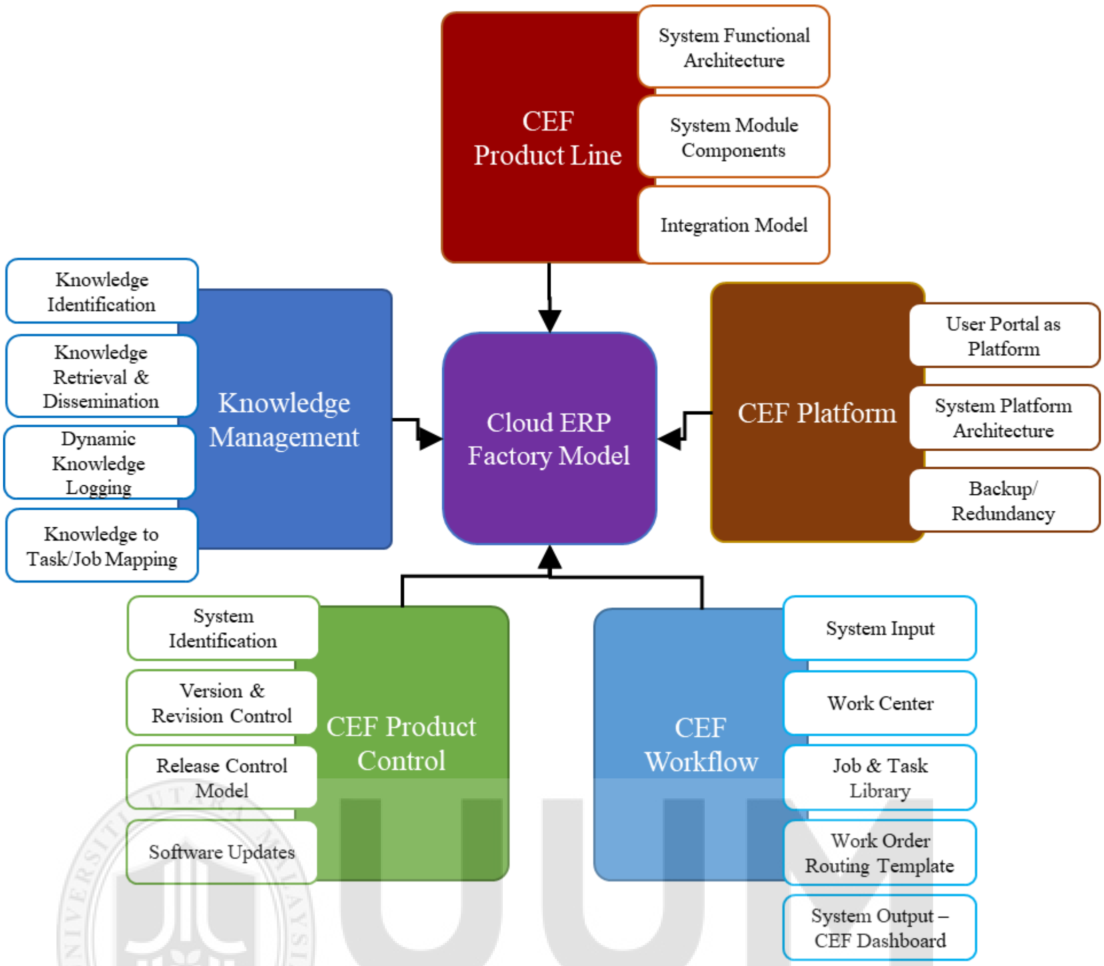



*Figure 4.29* Proposed CEF Model

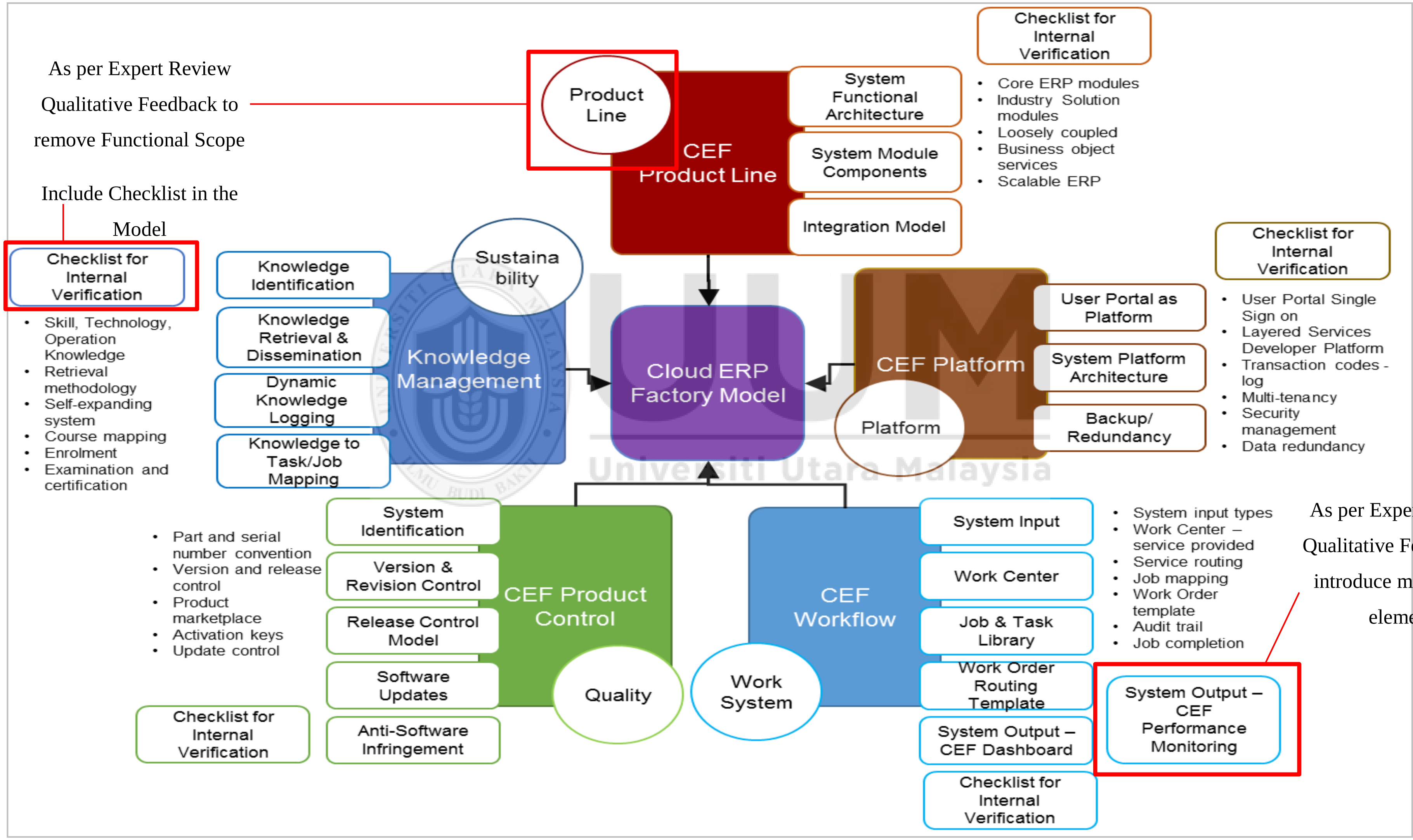



*Figure 4.30* Proposed CEF Model with Expert Review Feedback

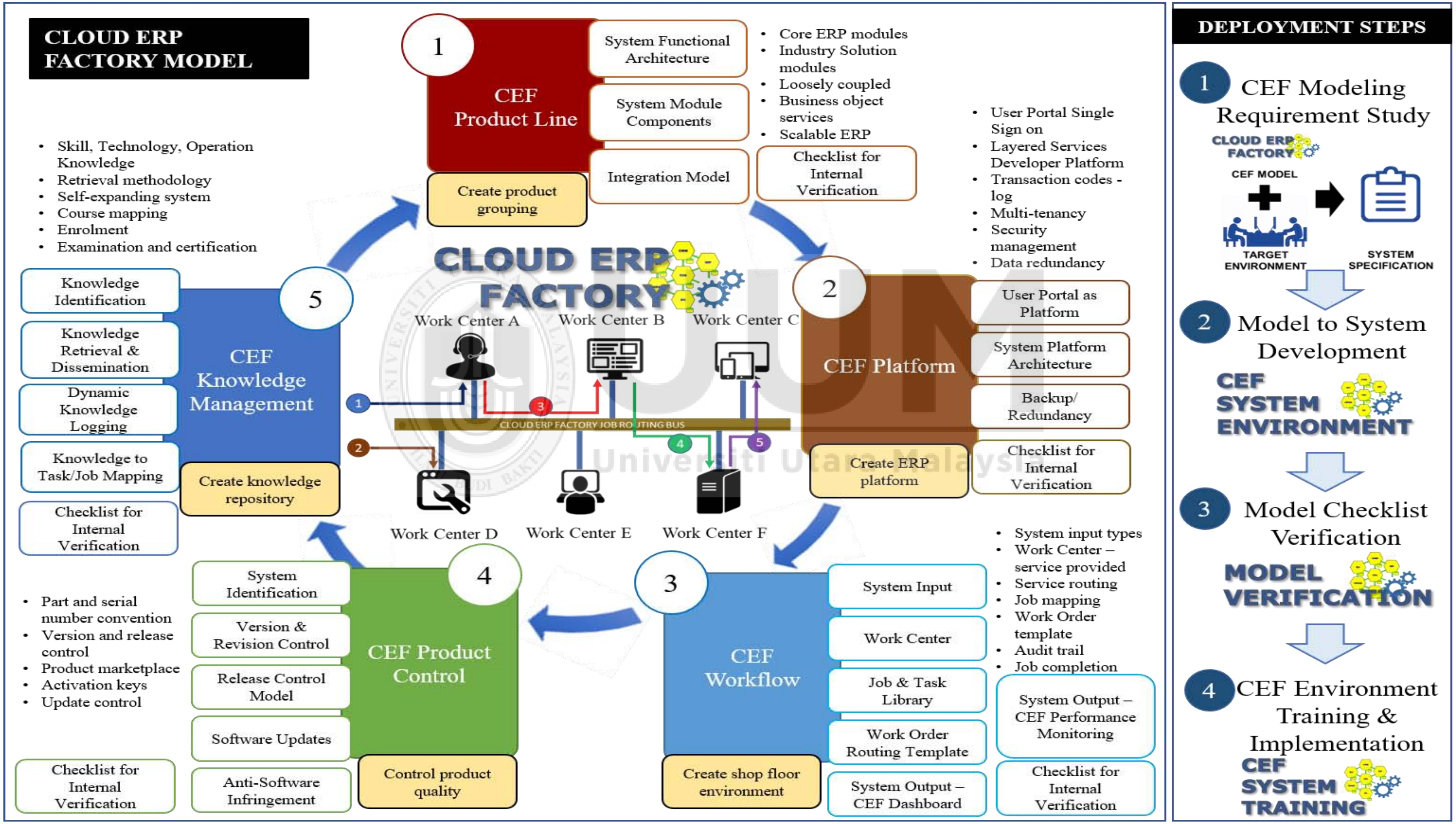



*Figure 4.31* Finalized Cloud ERP Factory Model

Each of the subcomponent checklists are focused on the ability of the subcomponent to provide Cloud ERP providers with detailed step by step method in implementing the CEF Model. Figure 4.32 until Figure 4.36 shows each subcomponent's core model elements with its corresponding checklist in Table 4.18 until Table 4.22.

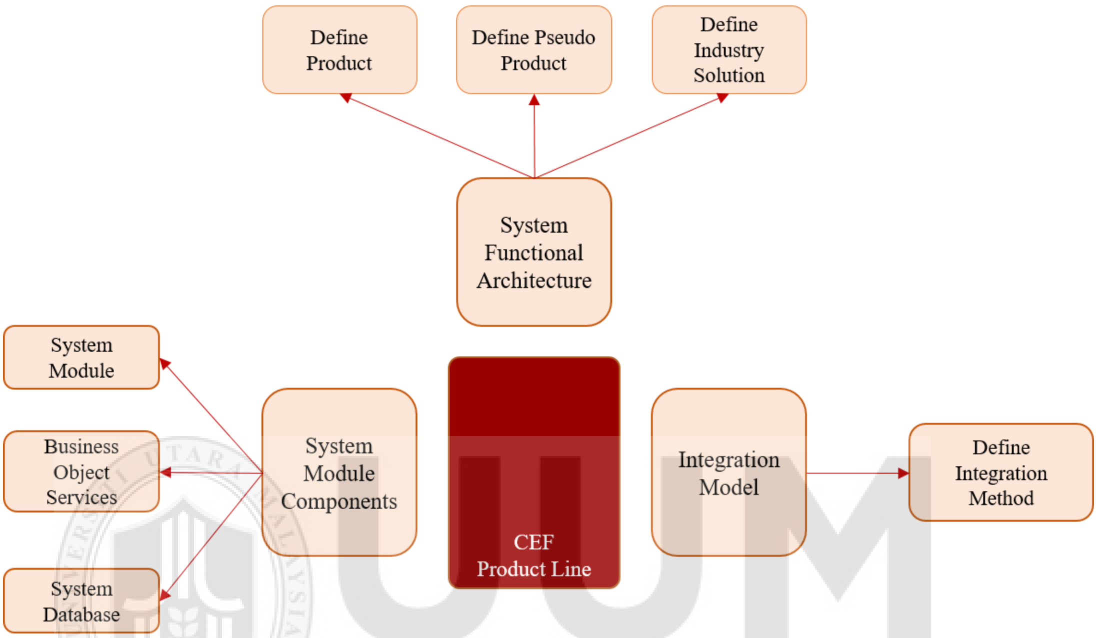



*Figure 4.32* CEF Subcomponent - CEF Product Line Architecture

Table 4.18

*CEF Product Line Architecture Checklist*

| **No.** | **Checklist Description** | **Tick Box** |
|---|---|---|
| 1 | The product line grouping should cover intended product family and its sub product as well as future product can be fit into of the product family. | ☐ |
| 2 | Important and common ERP modules should be defined and grouped as Core Module. | ☐ |
| 3 | Model can be deployed to customer individually or in integrated mode (loosely coupled). | ☐ |
| 4 | Core system, industry solutions and legacy can be integrated with standard interface protocols. | ☐ |

Table 4.18 continued

| | | |
|---|---|---|
| 5 | Future upgrades for one particular product should not affect the other system through backwards compatibility. | ☐ |
| 6 | Each product has its own configuration parameter. | ☐ |
| 7 | All product should fit in a common platform which has built-in product components. | ☐ |
| 8 | All product information is systematically through a system. | ☐ |
| 9 | Audit process and enforcement model are available. | ☐ |

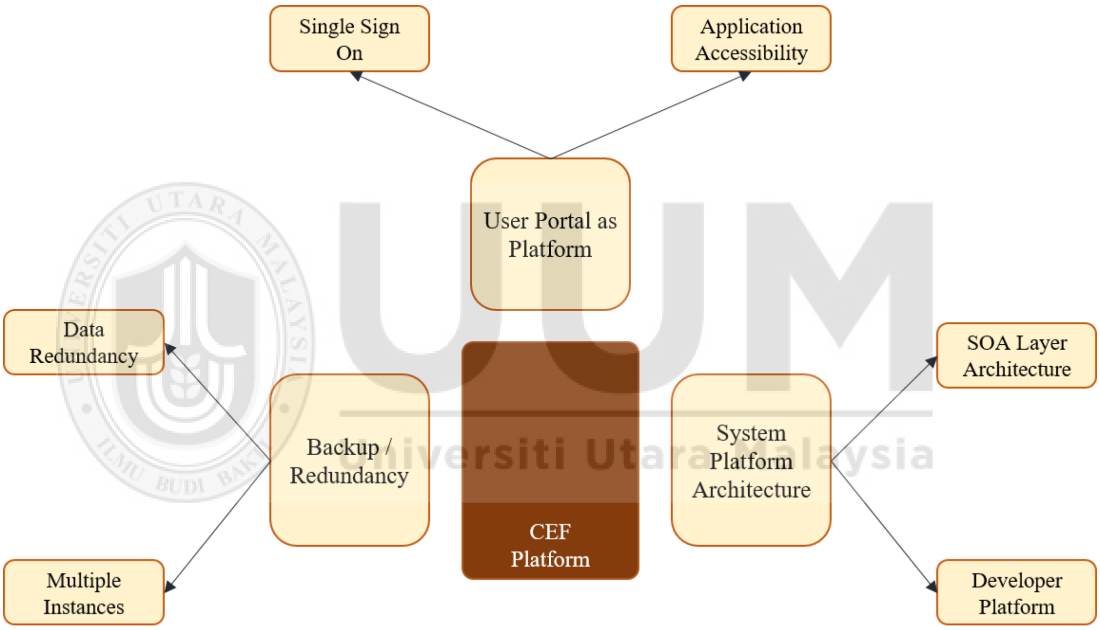



*Figure 4.33* CEF Model Subcomponent - CEF Platform Architecture

Table 4.19

*CEF Platform Architecture Checklist*

| No. | Checklist Description | Tick Box |
|---|---|---|
| 1 | The product line grouping should cover intended product family and its sub product as well as future product can be fit into of the product family. | ☐ |

Table 4.19 continued

| | | |
|---|---|---|
| 2 | It should be able to integrate with existing legacy product through API. | ☐ |
| 3 | Model can be deployed to customer individually or in integrated mode (loosely coupled). | ☐ |
| 4 | Future upgrades for one particular product should not affect the other system through backwards compatibility. | ☐ |
| 5 | Each product has its own configuration parameter. | ☐ |
| 6 | All product should fit in a common platform which has built-in product components. | ☐ |

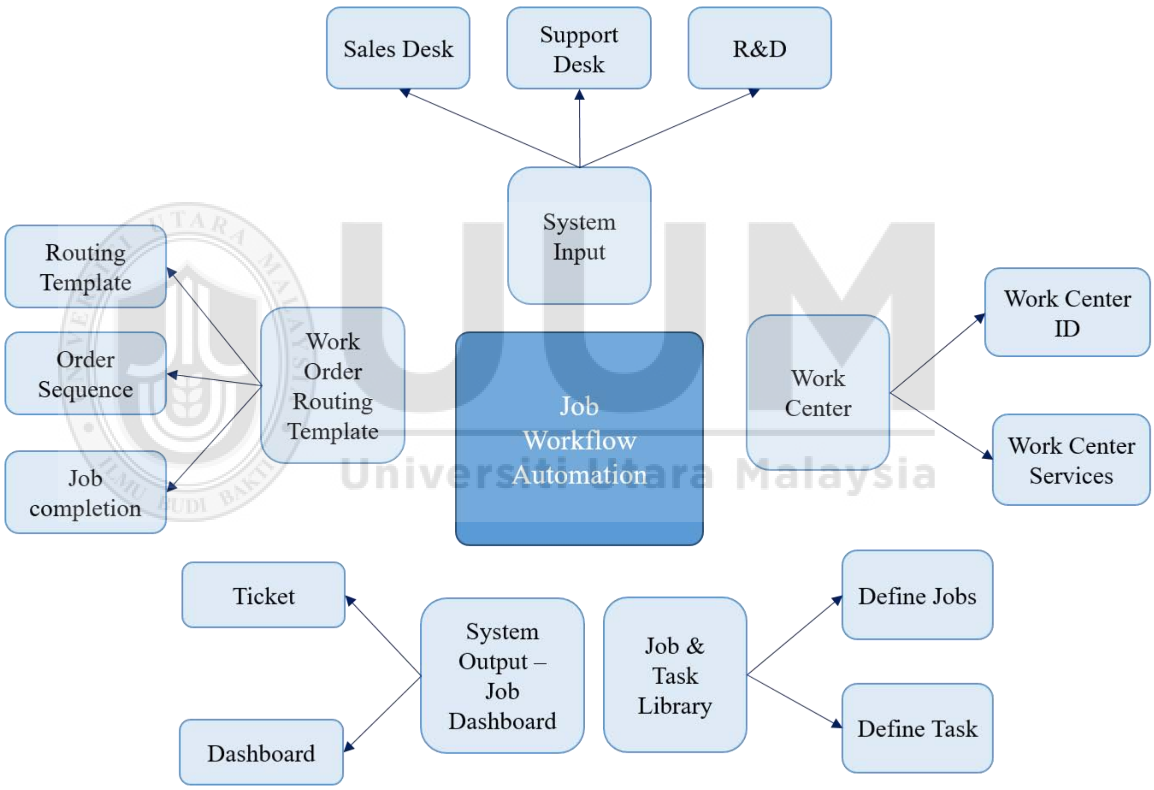



*Figure 4.34* CEF Model Component - CEF Workflow

Table 4.20

*CEF Workflow Model Checklist*

| No. | Checklist Description | Tick Box |
|---|---|---|
| 1 | The jobs can be across the department depending on the responsibility the task performed. | ☐ |
| 2 | Each department need to have their own department specific job complete with checklists. | ☐ |
| 3 | Each job can be used for other department's Work Template. | ☐ |
| 4 | Upon confirmation of a Work Order, work order template will be used to distribute the jobs assigned to the designated staff within the system. | ☐ |
| 5 | Each task should carry its own time frame to allow for tracking of progress and at which task. | ☐ |

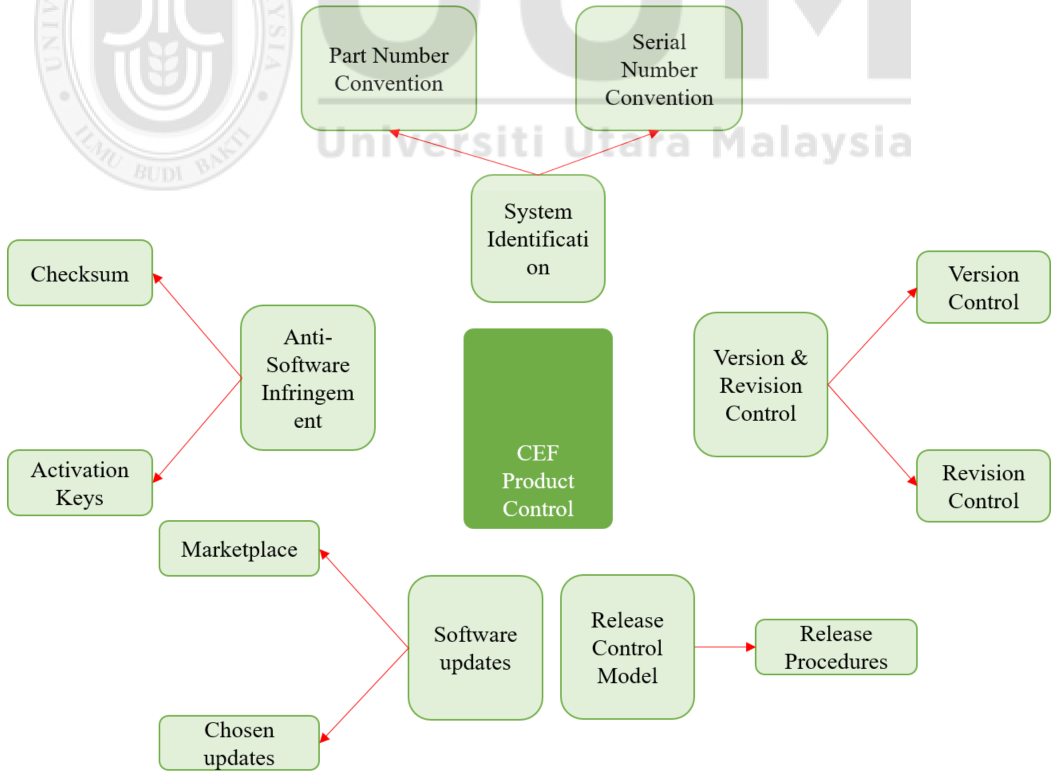



*Figure 4.35* CEF Model Subcomponent - CEF Product Control

Table 4.21

*CEF Product Control Checklist*

| No. | Checklist | Tick Box |
|---|---|---|
| 1 | Part number generation must be done with the event of new product released. (Concurrent only with a sale of a new product). | ☐ |
| 2 | Product database should contain all documentation, which includes product specification, release notes, system algorithm and actual coding files. | ☐ |
| 3 | Each product development owner is responsible for the release of the developed product. | ☐ |
| 4 | Serial number activation is required to allow for keeping tabs on the number of products being sold and/or deployed to clients. | ☐ |

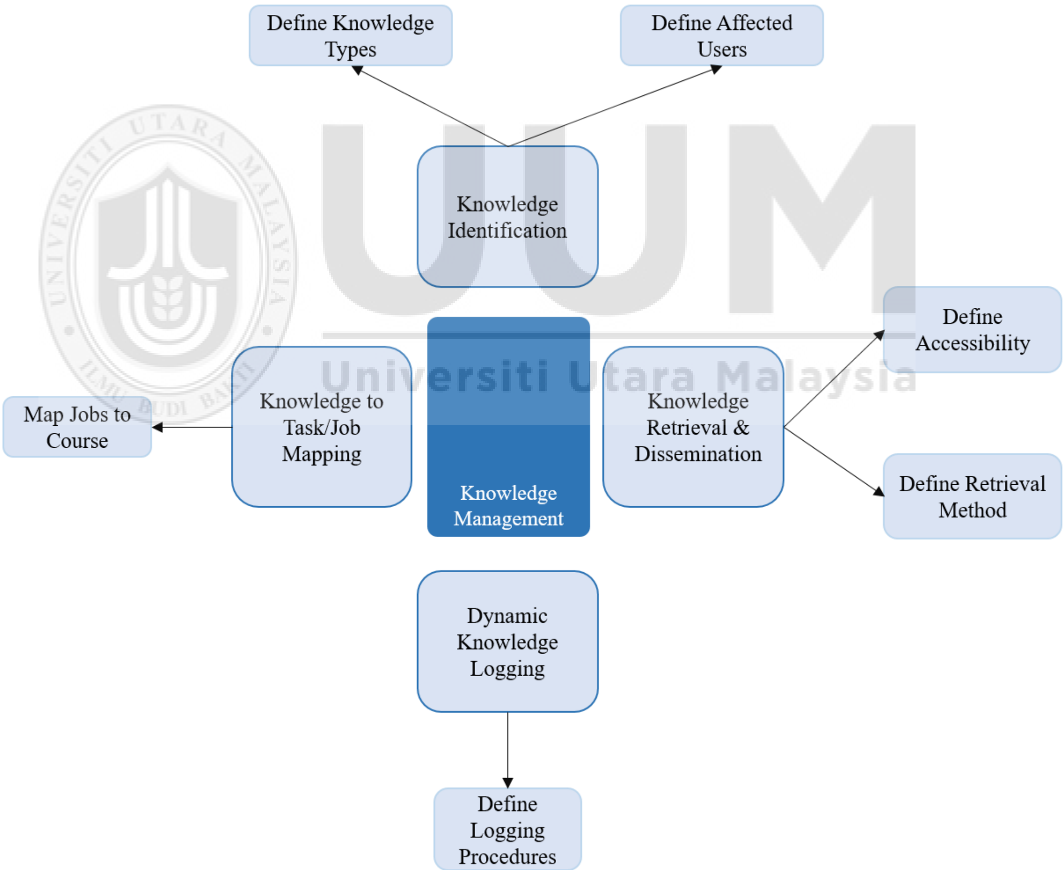



*Figure 4.36* CEF Model Subcomponent - Knowledge Management

Table 4.22

*Knowledge Management Model Checklist*

| No. | Checklist | Tick Box |
|---|---|---|
| 1 | The knowledge creator has full power over the content of the knowledge provided. | ☐ |
| 2 | The updates of contents, training materials, guides, and related documentation are conducted by the administrator, through the main knowledge management engine. | ☐ |
| 3 | Able to ensure knowledge retrieval and logging through procedures in the system | ☐ |
| 4 | The system ensures that all knowledge can be referred to from a single focal point i.e. dashboard. | ☐ |

CEF Planning and Performance Monitoring Dashboard is another output of the CEF Workflow component. This planning and performance monitoring dashboard was developed as a planning tool to be managed by a planner to oversee the overall jobs progress in the Cloud ERP production environment. The planner is able to monitor the progress of the request tickets and jobs created as well as manually changing the priority of the job based on selected criteria. The selected criteria can be defined either based on the value of the project, customer work load, and due date provided.

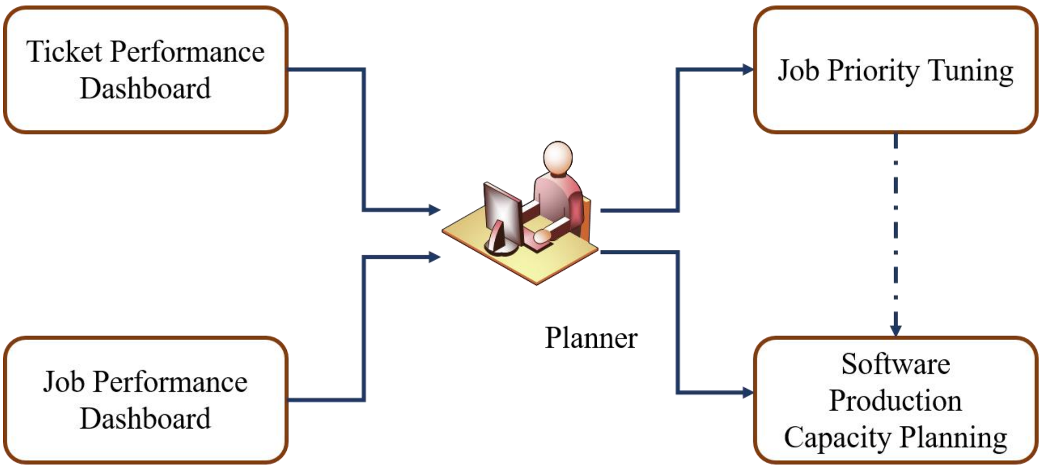



*Figure 4.37* CEF Planning and Performance Monitoring Dashboard Overview

In Figure 4.37, the CEF Planner is able to view four different types of monitoring information. Each monitoring information viewed provide the CEF planner with the necessary tool to make the decision to either increase or decrease the job's priority level. Through this monitoring module, the planner is able to monitor the overall capacity of the production environment, as well as the resources required to complete the job. The CEF model capacity planning is very much similar to standard product manufacturing model. However, unlike a standard manufacturing model, inventory or stock level is not required in the capacity calculation. This is due to the fact that software products do not depend on raw materials to be manufactured. The capacity planning depends heavily on the manpower resources that operates the Work Center. Thus, the bottleneck for the production depends on any of the heavy-loaded Work Center for a particular software product work order routing template. The manpower resources for any particular Work Center in the CEF model production can be compared to the machine's capacity of standard product manufacturing. During the production run, job performance monitoring can be used as feedback performance data for the capacity planning module to optimize production efficiency. Figure 4.38 shows an example of capacity planning dashboard that can be used to manage the resources' workload by Work Center. The information can be used to feedback the job performance monitoring dashboard so that the software production is able to achieve an optimized capacity.

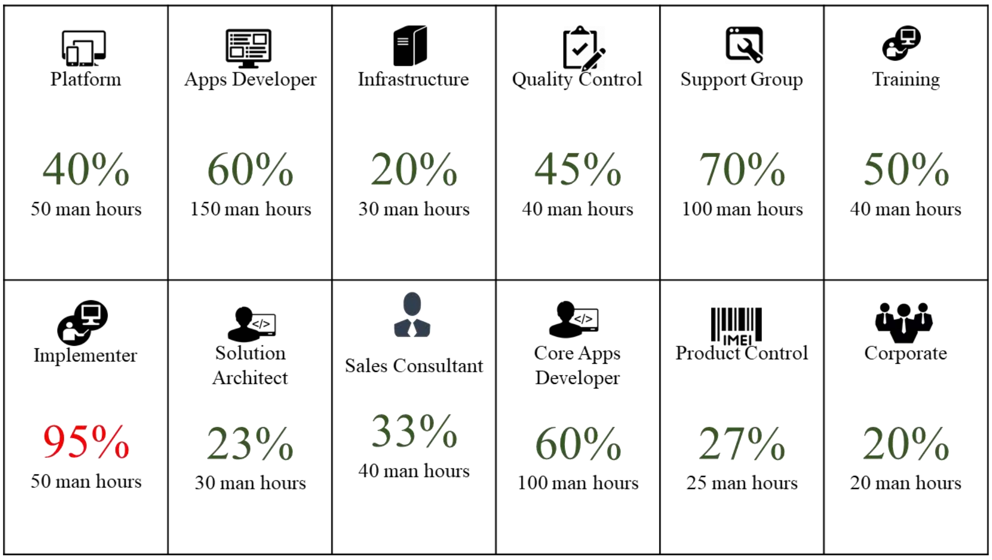



*Figure 4.38* Visualization of Capacity Planning Dashboard

The detail of the production capacity calculation will not be discussed here as it is similar to the normal product manufacturing standard. This module is vital to the whole CEF Workflow model as it provides the overall view of the jobs created and distributed to the Work Centers. In general, by applying the CEF Workflow component, a platform has already been setup to manage the workload in a CEF production environment, which in turn, can be utilized to significantly reduce the workload as compared to traditional project-based approaches.



## 4.7 Applying the Cloud ERP Factory Model

Once all of the model components were identified complete with its own design guideline and checklist, the CEF Model can be applied through a series of steps referring to Figure 4.39. The typical expected output of the CEF Model is illustrated in Figure 4.40. The figure shows the relationship between the model components, which shows the CEF Product Line and CEF Platform being managed by CEF Product Control. CEF Workflow component can be translated into Work Order management system, which manages the process of scheduling and assignment of job to the Work Centers. CEF Product Control can be realizes into a system that manage the product specification and its release control. CEF Knowledge Management on the other hand, provides real time knowledge repository using Learning Management System.

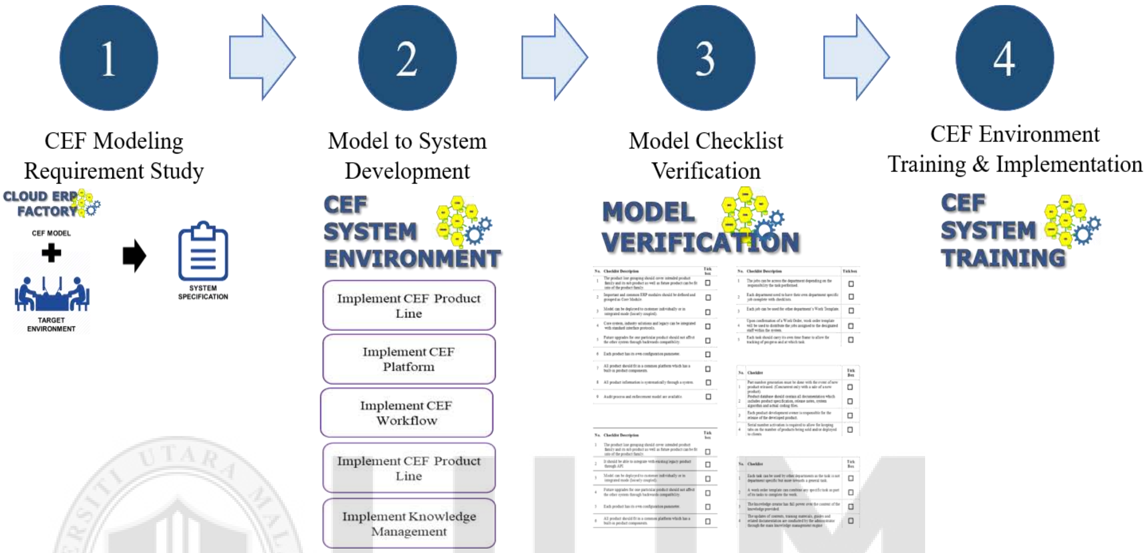



*Figure 4.39* Step by Step of Applying CEF Model

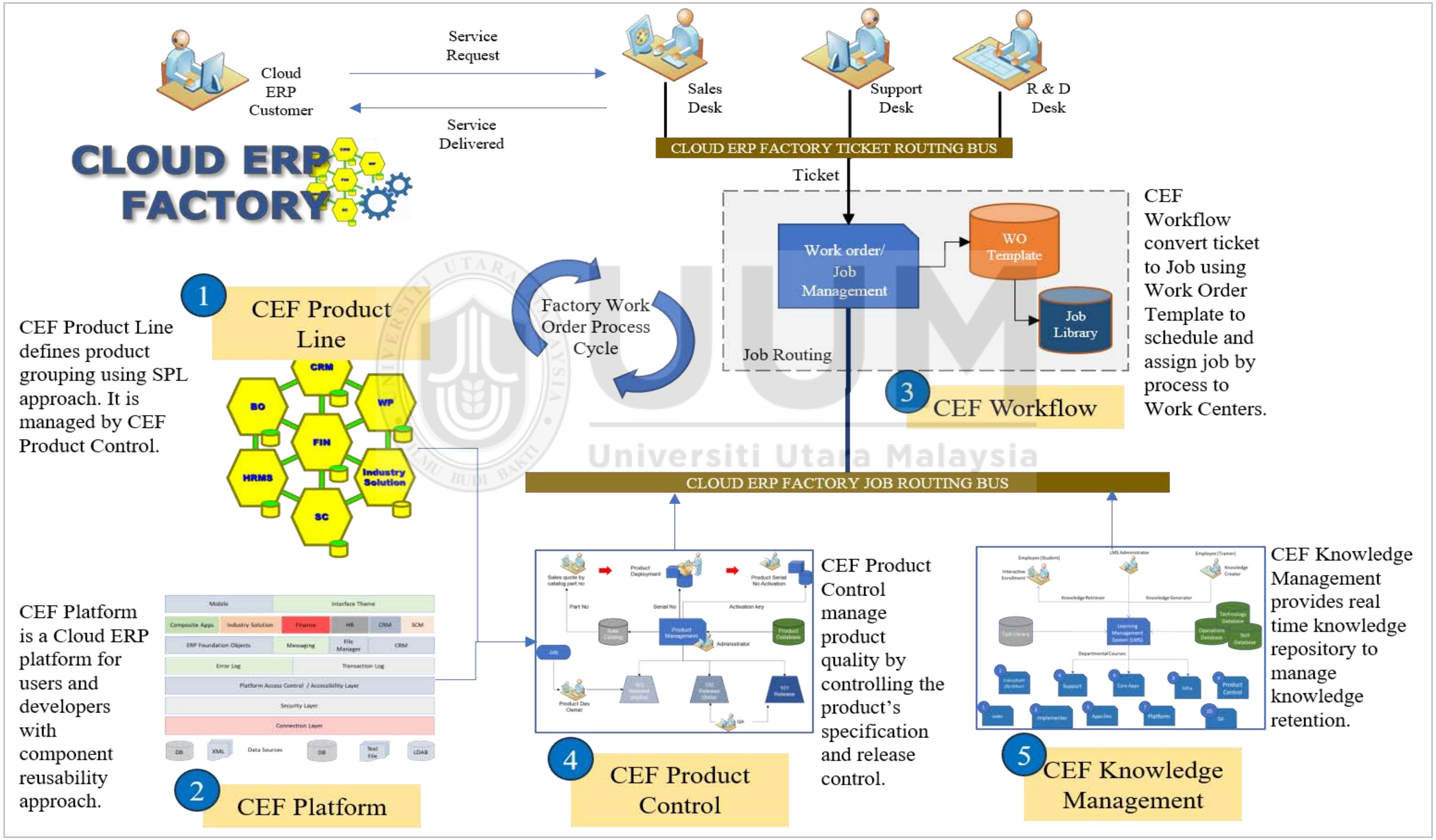



*Figure 4.40* CEF Model Typical Expected System Output

## 4.8 Summary

In summary, each subcomponent was derived from several factors based on the findings discussed in our Literature Review. These subcomponents were then compared with similar research to identify its core model element. Five subcomponents have been identified to make up the CEF Model, those are CEF Product Line, Platform, Workflow, Product Control, and Knowledge Management. The model was then subjected to expert reviews feedback for enhancement. The feedbacks of the expert reviews were adapted into the final model which was presented with the guideline of its deployment.

# CHAPTER FIVE
# MODEL VERIFICATION

## 5.1 Introduction

This chapter focuses on the activities to provide a technical feasibility of the CEF Model application finalized in Chapter Four by developing a CEF environment. The purpose is to verify that the model is technically feasible to be deployed into a Cloud ERP production environment. The development of a CEF environment was designed to capture the essence of the CEF Model including its subcomponents. This will be done by adopting the model to an existing commercial Cloud ERP company as each subcomponent is used to demonstrate the usability of the CEF Model and presents a model verification and result analysis of the exercise. The objective of this phase is to verify the finalized model by adopting it into the existing commercial Cloud ERP production environment.



The model to system development for the CEF System was completed with the conjunction of a Ministry of Education’s grant. This grant has enabled us to employ two full-time research assistant (RA) to assist in the process of identifying and executing the research. The system development of the CEF System was in part a collaboration with a Cloud ERP production company together with our research personnel in consulting and CEF software development role.

The model verification process started with the implementation of the finalized CEF Model into an existing Cloud ERP production company. Each of the CEF model component were implemented by taking the existing practices and matching them with the guidelines provided.

The overall model verification was done in five stages corresponding with the five major components of the CEF Model.

1. CEF Product Line – examined existing products developed by the company and verify the ability of the products to be considered as Cloud ERP solution.

2. CEF Platform – verified that the platforms used by the company were in line with the guidelines of the CEF Model.

3. CEF Workflow – verified the jobs done by the whole organization and determined the necessary workflow to complete them. A system was developed to ensure that each job follows certain procedures and protocols.

4. CEF Product Control – determined the steps taken during product release was in accordance with the guidelines. A system was required to be developed to ensure the products were released based on a strict protocol to ensure the quality of the released products.

5. CEF Knowledge Management – base study of the organization ability to log knowledge dynamically was conducted. A knowledge repository and dissemination system were developed to allow employees to access and log knowledge based on the jobs performed.

## 5.2 Environment Selection Criteria

The CEF Model environment depends on the company selected to be implemented. The selection of the environment can be classified to several factors. The first selection criteria state that it needs to be a software production company that develops ERP or Cloud ERP solutions. The second criteria as stated in Chapter Four, the company suitable to be implemented with a CEF Model should also be in a business that has volume, servicing more than 100 customers. Thirdly, the company has to be accessible to allow us to conduct a model implementation as the CEF Model requires for business process change within the company. The fourth criteria involve the size of the company whereby the CEF Model implementation have to take into account the scope of the existing departments to ensure that the CEF Model is a feasible attempt.

Based on the criteria provided above, Company A was selected, which has been in business for the past 20 years in ERP software production. Referring to Table 5.1 is the criteria table for the companies that were approached.

Table 5.1
*CEF Environment Selection Criteria Result*

| **Environment Selection Criteria** | **Company A** |
|---|---|
| Develops ERP or Cloud ERP solutions | ERP and Cloud ERP |
| Services upwards of 100 customers/clients | 400 clients |
| Allows access to research | Yes |
| Number of employees | >30 |

The selected company has been in the ERP solution business and has faced the same issues regarding the complexities of managing Cloud ERP deployments, increase of workload as more clients are adapting to Cloud solutions, inconsistency of application quality by the developers, as well as not able to retain the knowledge within the organization if and when an employee resigns. One of the main challenges of this phase is to make the organization understand the CEF Model and being able to transform the current practice into the CEF Model environment. This also validates the understandability of the CEF Model by the relevant audience.

### 5.3 Prototype Model Deployment Team

A team consisting of researcher and the selected company's personnel was formed to handle the CEF model deployment activities and finally verify model prototype environment completion using model verification checklist. The team's demography is listed in Table 5.2, which highlights the fact most of the team members are highly experienced in their line of work, having more than 15 years of working experience.

Table 5.2
*Demographic of Model Deployment Team*

| **Position** | **Qualification** | **Experience (Years)** |
|---|---|---|
| Researcher | MSc. Computer Science | 25 |
| Researcher | PhD Software Engineering | 25 |
| Researcher | PhD Computer Science | 25 |
| System Analyst | MSc. Computer Science | 25 |
| System Architect | MSc. Computer Science | 15 |
| Software Developer | BSc. Computer Software | 10 |
| Research Assistant | BSc. Business Administration | 15 |

### 5.4 Overall Environment Requirement Studies

In order to come up with a system specification based on the CEF Model for the selected environment, the existing system environment must be considered as well as how to make it adapt to the CEF Model. As shown in Figure 5.1, the process of fitting the model into the existing environment finally generates the system specification which then can be translated into few software systems simulating the CEF Model.

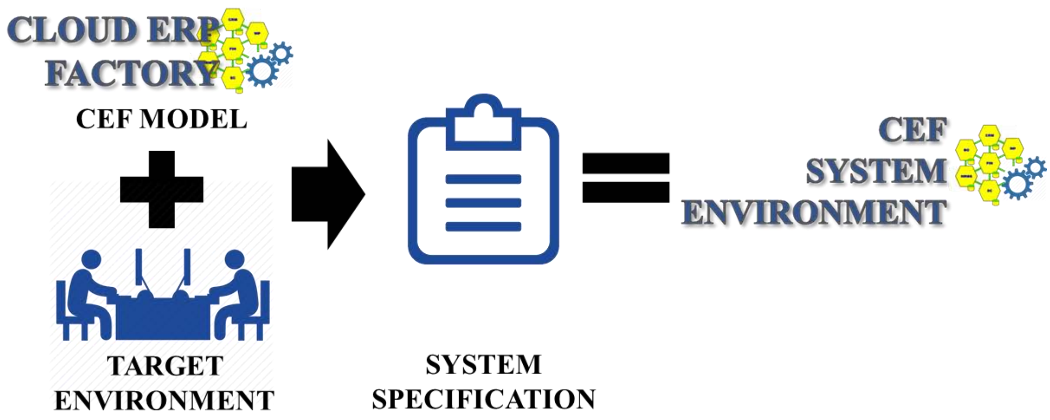



*Figure 5.1* Modeling the CEF environment

The first phase in our model prototyping validation activity was to conduct and verify the overall existing environment requirement studies. This has allowed us to take our CEF Model and combine it with any of the company's existing practices to demonstrate our CEF Model environment effectively. Each environment requirement studies are separated based on the subcomponent of the CEF Model described in Chapter Four. Table 5.3 describes the summary of the overall requirement studies done for each subcomponent.



Table 5.3

*Summary of Overall Requirement Studies*

| **CEF Model Subcomponents** | **Previous Settings** | **Changes Required** |
|---|---|---|
| Software Product Line | • Products are heavily customized<br>• Cater for enterprise typed client<br>• Existing products already based on modularity | • Redefined product group and pseudo product |
| Platform Architecture | • Current platform very robust<br>• Includes security, user management and application modules | • Adapts the current platform<br>• The necessary element to fit into the CEF Model is present<br>• No major changes required |
| Workflow Management | • Manual support flow, no jobs created based on helpdesk tickets<br>• Helpdesk tickets are closed not according to SLA | • Create a semi-automated system for helpdesk/dashboard to distribute the helpdesk requests into jobs. |

Table 5.3 continued

| | | |
|---|---|---|
| | | • Job is assigned to the specific work centers |
| Product Control | • Existing product management control exist but lack versioning control and software update management | • Define version control protocol for Cloud deployment and updates<br>• Define part number convention and serial number<br>• Define activation key usage |
| Knowledge Management | • All knowledge, skill and material reside with specific assigned developers<br>• No central knowledge repository<br>• File management in platform is not properly set up | • Define knowledge type to be used as reference<br>• Define knowledge retrieval mechanism<br>• Define dynamic logging procedures of knowledge discovered<br>• Define overall structure of LMS system |

### 5.4.1 CEF Product Line

The CEF Product Line model dictates the product definition and the details of the defined products. The guidelines outlined in Chapter Four are used to explain the steps needed to implement the CEF Product Line model.



i. Define core product and their grouping to enhance reusability and modularity. The product line should be able to group current software modules as well as any future potential modules.

   The company has existing core ERP products, which are already being deployed on the Cloud. The company's core ERP products included Customer Account Management System (CAMS), Supplier Account Management System (SAMS), Human Resource Management System (HRMS), Inventory Management System, Work Order Management System, and Financial Information System (FIS). Each core ERP product was able to provide sufficient evidence that the product group is able to conform to the CEF Model requirements, thus no major changes have been made.

ii. Define pseudo product and their grouping to enhance reusability and modularity.

Due to the fact that the core ERP products for the company already existed, the pseudo product and their grouping has already been defined. Each of the core ERP module pseudo products are explained below as no changes were required to be conducted.

Customer Relationship Management (CRM) module should consist of several important aspects to handle different customer relationship activities. An example of a typical CRM module includes Customer Information System, Sales and Invoicing, Project Management, Marketing, and Order Fulfillment as shown in Figure 5.2.


*Figure 5.2* Customer Relationship Pseudo Product

The Supply Chain Management (SCM) modules deal with the various transactions, in terms of purchasing sourcing of materials and adds more value to the organization in order to increase productivity, speed of response, and provides proactive trigger mechanisms. SCM should consist of modules that handle Supplier Information System, Purchasing, Invoice and Payment, Inventory, and Budgeting as shown in Figure 5.3.

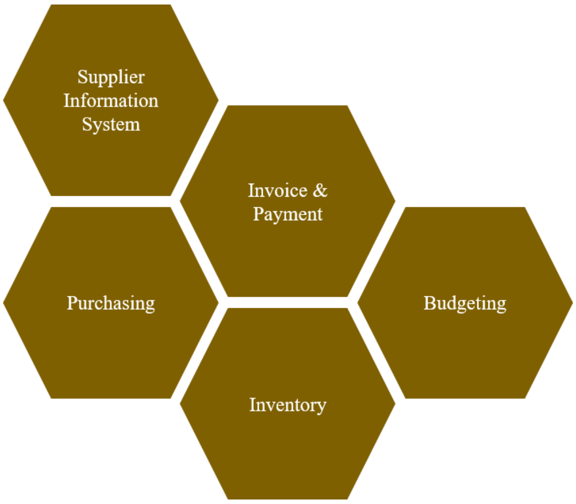



*Figure 5.3* Supply Chain Module Example

Human Resources Management System (HRMS) is relatively self-explanatory, as it deals with the human development factor in an organization. HRMS modules should then include Personnel Administration, Attendance/Leave, Payroll, Recruitment, and Training as shown in Figure 5.4.

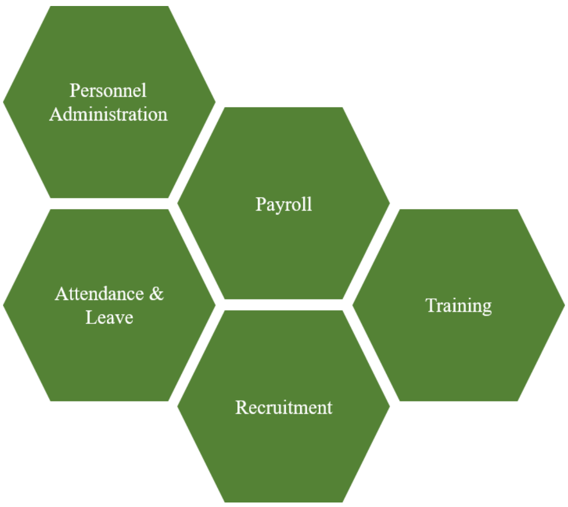



*Figure 5.4* HRMS Sample Modules

Financials module should consist of a general ledger, accounts receivable, accounts payable, analysis reporting, and treasury to audit as shown in Figure 5.5.

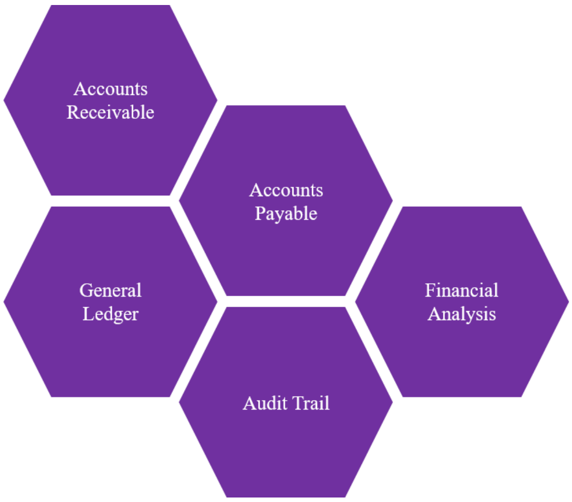



*Figure 5.5* Financials Modules Example



Back Office Solution modules can be comprised of site activities, document control, security, and communication channels as it is integral to the success of implementation, delegation, and communication as shown in Figure 5.6.

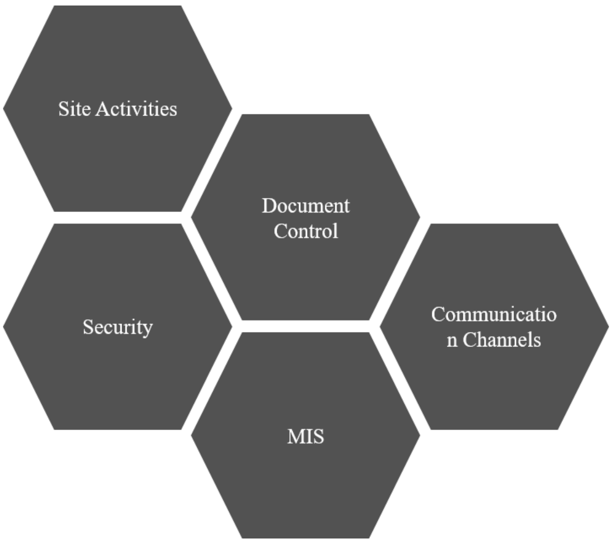



*Figure 5.6* Back Office Module Sample

iii. Define industry solution products and their grouping.

In the next step, it was defined that the Industry Specific Solutions can vary from different business entities. Although the core ERP modules remain the same, the business operations module is specific to the type of industry the organization belongs to. Depending on the business structure, the Industry Specific module should be designed with the intention to work coherently with the other business core modules. Figure 5.7 shows a sample of some of the industry solution products.

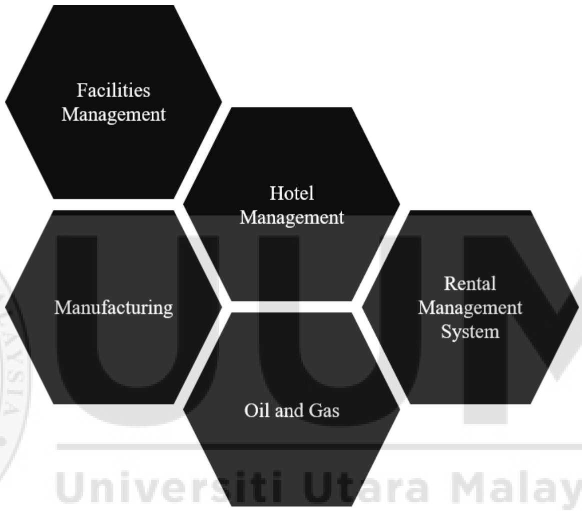



*Figure 5.7* Industry Specific Solution Example

iv. Define system module component that makes up a system module.

The existing core ERP products were completed in terms of having its system module components. Each system contained the main system module, its business object services and system database. Was shown adequate evidence that the system module components exist for the existing products.

v. Define foundation system component if applicable.

The foundation system component was already defined within the core ERP products.

vi. Define integration between core ERP module and Industry Solutions.

For every single new ERP implementation, the company has shown that the core ERP modules were not developed from scratch, thus promoting reusability, and are able to be integrated with the Industry Solution modules.

vii. Define product management model. All product information should be managed systematically by a system.

It was discovered that the company has its own product management system. The system is able to register new products or services, but lacked the ability to control the version of products being deployed. The current research redefined and added a version control procedure within the existing product management system.

viii. Define audit method that can enforce the practice without fail.

Evidence has been shown through documentation that all of the company's existing products were developed according to the CEF Model's practice.



### 5.4.2 CEF Platform Architecture

The CEF Platform Architecture model dictates the platform definition detailed platform characteristic to be used. The guidelines outlined in Chapter Four were applied to explain the steps needed to implement the CEF Platform Architecture model.

i. User portal platform should be able to host all Cloud ERP modules.

The company has its own platform used to deploy their current existing products. The user portal was verified to allow access all of the hosted products within the platform. The hosted products or application modules was listed based on the user's accessibility clearance. If a user does not use a particular application, the application is not displayed in the platform.

ii. Developer platform should clearly separate all layers for scalability.

The organization has complied to the platform requirement with the system architecture platform that was suggested by the CEF model. The separation of the layers with its clear functionality can be illustrated by Figure 5.8. By having the separation layers, the developer's role and functionality can be separated by platform group and application group. In this prototyped CEF environment, the platform group is a group that develops and enhance the platform and tools which are used by the application development team. The job of managing the layers between one and six is handled by the platform group. The core application group is responsible for the development and maintenance business process application that belongs to layer seven. The final layer that deals with interface and theme is handled by the mobile team.



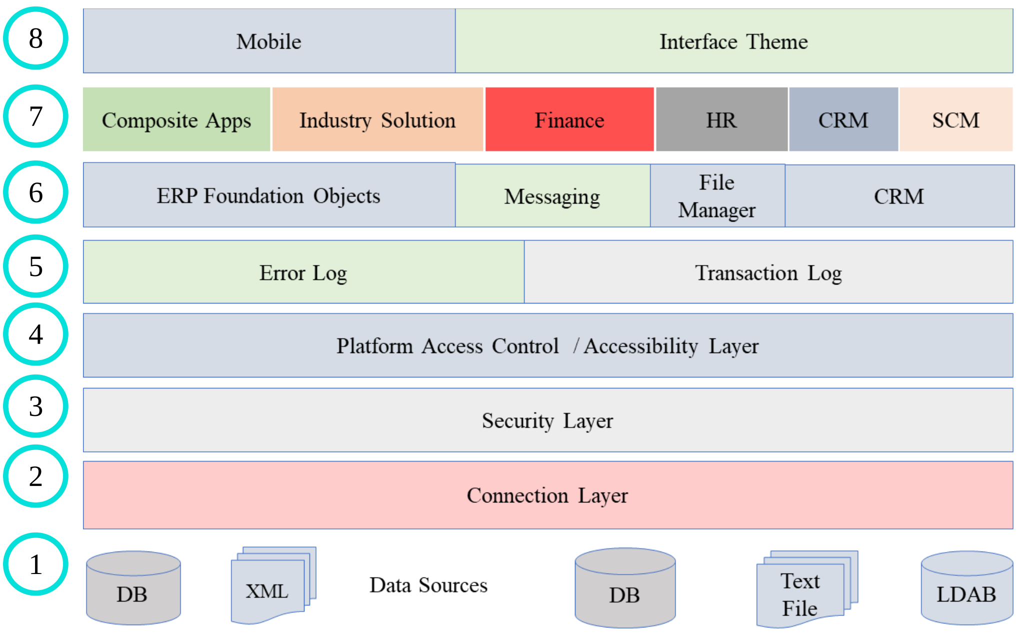



*Figure 5.8* Platform Layer Functionality Separation

iii. System security and redundancy should be part of the overall system security built-in features.

The platform group which also handles the Security Layer as described previously employed multi-instances of the system as a form of system redundancy feature.

iv. Each platform's functionality should work as a service object within the platform's layer.

The setup for this CEF environment has demonstrated that each platform's functionality communicated with function's services programming.

v. All system modules should define their services with transaction codes.

The system modules hosted in the platform has been defined with specific transaction codes for their own services. In Table 5.4, a sample of transaction codes used by some of the system modules is displayed.



Table 5.4
*Sample Transaction Codes*

| **System Modules** | **Sample Transaction Codes** |
|---|---|
| CAMS | Create Customer, Edit Customer |
| SAMS | Create Supplier, Edit Supplier |
| HRMS | Create Personnel, Apply Leave, Approve Leave |
| Inventory | Create Part Number, Edit Part Number, Approve Part Number |
| Financials | Create Account<br>ID, Create New Transaction |

vi. All transaction should be captured in the transactions log.

These transaction codes were defined so that each data transaction was logged in the database. This would allow for data lookup for report purposes as well. Table 5.5 shows some samples of transaction log types used in the platform.

Table 5.5
*Sample Transaction Log*

| Transaction Log Name | Transaction Description |
|---|---|
| LM02 | Location registration |
| LM97 | Failed Login |
| LG21 | Log In |
| LG22 | Log Out |
| LG23 | Login Session Expired |
| LG33 | Change Document No |

vii. All transaction services should define possible error logs.

Each data transaction was defined with possible error logs for future reference in the event of any errors occur while using the system. Table 5.6 shown is a sample of possible error logs that were captured.



Table 5.6
*Sample of Error Logs*

| System Module | Sample Error Logs |
|---|---|
| CAMS | Unable to create new customer |
| SAMS | Unable to select supplier |

viii. All system modules should provide debug features.

To help the developers in fixing any possible bugs quickly, all system modules was defined to provide debug features. The debug features reside inside the system modules.

ix. Platform should provide multi-tenancy control for all system modules.

To implement multi-tenancy control, the platform has a feature to create multi-company to support multi-tenancy control.

x. Define audit methods.

Evidence has been shown through documentation that all of the company's existing platforms were developed according to the CEF Model's practice.

### 5.4.3 CEF Workflow

As the engine of CEF, a Job Management System was created complete with a Dashboard view. Then, each business process for each department has been defined to better suit a CEF environment. Each department have listed their own specific task list and been entered into the Job Management System. The Job Management system acts as a task list library to be used in the Work Order routing workflow in the Job Management System.



i. Define system inputs into the CEF Workflow model.

The definition of system inputs for the CEF Model implementation has been defined as Sales Desk, Support Desk, and Other Input Request. Sales Desk refers to any request requested during sales activity. This may include the request to create a new version of an application module for a specific customer, a new installation request for platform, and core ERP products. Support Desk requests are mainly requests during and after a Cloud ERP product or project has been implemented for the client. The Other Input Request was defined as any request for the purpose of research and development. Each system input is designated by its own ticketing request ID number, which is used throughout the whole workflow implementation as a reference point.

ii. Define Work Centers.

Next, Work Centers were defined, which provided specialized services within the factory process. It can also be referred to as Departments. For this company, twelve Work Centers were defined as shown in Table 5.7. Each Work Center is specialized in their roles within the CEF Model.

Table 5.7
*Work Center Identification*

| No. | Name | Role | WCID |
|---|---|---|---|
| 1 | Platform | Maintain and upgrade the platform environment |  |
| 2 | Apps Developer | Maintain, configure, customize, improve |  |
| 3 | Infrastructure | Build, maintain, and manage the Server and network Infra. |  |
| 4 | Quality Control | Enforce and audit QA procedure |  |
| 5 | Support Group | Maintain and support post implementation job |  |
| 6 | Implementer | Plan and deploy new implementation order |  |
| 7 | Solution Architect | Business Consultation and Product and Solution Architect |  |
| 8 | Sales Consultant | Manage CRM scope, quotation, and invoicing |  |
| 9 | Core Apps Developer | Maintain and Upgrade standard product, develop new products |  |
| 10 | Product Control | Manage product management policy and server |  |
| 11 | Training | Manage Product training |  |
| 12 | Corporate | Finance, HR, and CEO office role |  |

iii. Define work order, job, and task library into a library of services.

Each of the Work Centers defined have their own type of jobs. Each job defined will have its own set of task list. This information is then stored in a central database, which the research has defined as Job Library. Table 5.8 is a sample of jobs and task for each Work Center.

Table 5.8

*Sample of Work Center Jobs*

| No | Work Center | Sample Jobs |
|---|---|---|
| 1 | Sales Consultant | a) Presale Meeting<br>b) Register Customer and Project in CAMS |
| 2 | Solution Architect | a) Verify new application features |
| 3 | Core Apps Developer | a) Release new module development<br>b) Register new application modules |
| 4 | Application Developer | a) Customize client application<br>b) Support client request for bug fix |
| 5 | Support Team | a) Prepare technical request document<br>b) Verify apps functionality |
| 6 | Quality Assurance | a) System testing and buyoff<br>b) Prepare customer satisfaction survey |
| 7 | Corporate | a) Create/update LMS record<br>b) Run monthly payroll |
| 8 | Training | a) Prepare training materials for application training<br>b) Prepare and set training agenda |
| 9 | Implementer | a) Test and buyoff system using UAT<br>b) Conduct User Requirement Study |
| 10 | Platform | a) Update Platform features<br>b) Create new platform interface |
| 11 | Infrastructure | a) Set Up system server<br>b) Upgrade Networking Hardware |
| 12 | Product Control | a) Release activation key<br>b) Provide part number and serial number |

iv. Define competency skills required for each job.

The competency skills required have been defined for the job defined for each Work Center. In Table 5.9, is a list of sample competency skills for crucial jobs.

Table 5.9

*Sample Competency Skills*

| No | Job Name | Sample Competency Skills |
|---|---|---|
| 1 | Set Up system server | Server Set up knowledge |
| 2 | Support client request for bug fix | ASP, PHP, Java, Database programming knowledge<br>Application troubleshooting knowledge |
| 3 | Verify new application features | Database Design<br>Application Design<br>Software Engineering |



v. Define the task list for all specified jobs.

It has been clarified that each job defined in the Job Library contained its own Task List. In Table 5.10 are a sample of Jobs complete with their own task list.

Table 5.10

*Sample of Task List for Jobs*

| No | Job Name | Sample Task List |
|---|---|---|
| 1 | Documentation and Training | a) Prepare training material<br>b) Set training session<br>c) Setup system / module for training<br>d) Train users |
| 2 | Conduct User Requirement Study (URS) | a) Prepare meeting for URS buyoff<br>b) Create test data<br>c) Test system functionalities<br>d) Test system usability<br>e) Release test report<br>f) Make decision no -no go |

Table 5.10 continued

| | | |
|---|---|---|
| 3 | Kickoff Meeting | a) Schedule Meeting<br>b) Meeting attendance report<br>c) Introduce team members<br>d) Brief project overview to clients<br>e) Review project schedule<br>f) Overview requirement study<br>g) Review project expectation<br>h) Generate and publish meeting report<br>i) Set baseline Gantt Chart<br>j) Set next meeting date |

vi. Define Work Order template to represent specific routing for all common input types.

The Work Order template has been defined to contain several jobs across Work Centers. These jobs were provided with their own order sequence. In Figure 5.9, it was defined that each Work Order has its own routing table, which flows through the CEF Job Routing Bus.

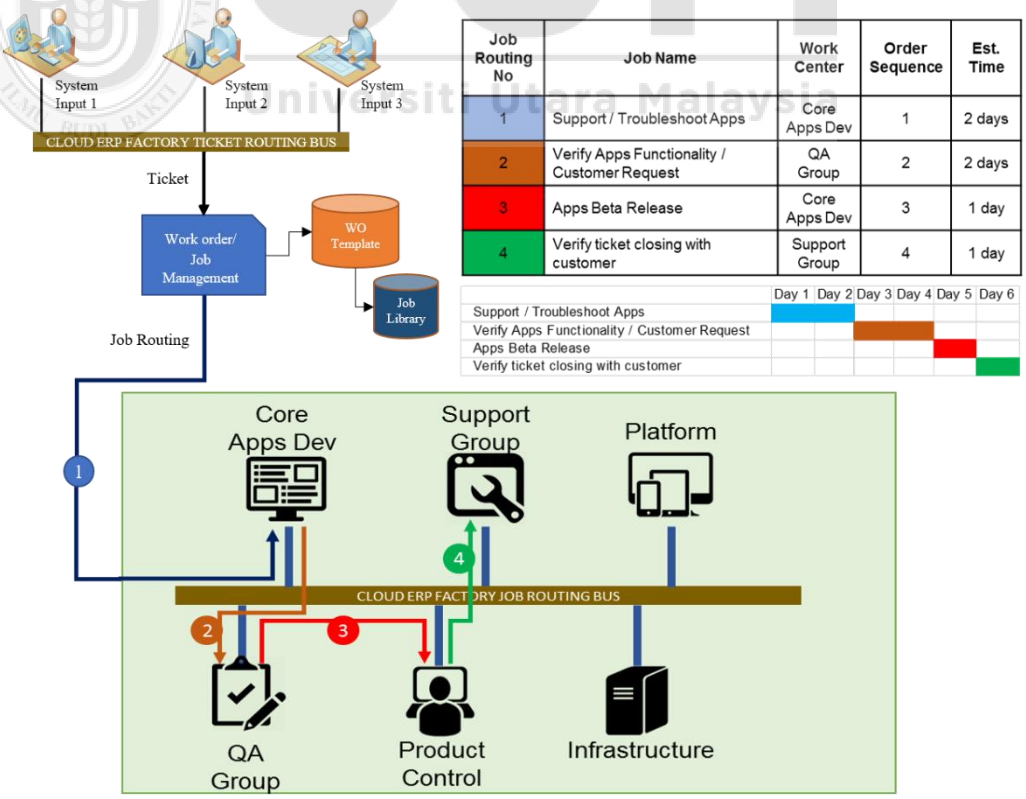



*Figure 5.9* Sample Work Order Routing Workflow

vii. Define order sequence for each Work Order template.

It has been defined that in a Work Order template, any job can share the same priority level. Furthermore, it was also decided that the priority level is set by default as incremental number starting from number 1 onwards.

viii. Create system to support the operation as specified with the parameters defined earlier. The system must provide operation enforcement and audit trail.

A system named Job Management System was developed, together with a Performance Monitoring capability to handle all workflow management requirements. Figure 5.10 and 5.11 shows the developed output for the CEF Workflow component being used to monitor the job management and performance monitoring. This system is detailed further in the section 5.5 Model to System Development.



*Figure 5.10* Performance Monitoring Agents

*Figure 5.11* Performance Monitoring Dashboard

### 5.4.4 Product Control

The CEF Product Control model implements the procedures of controlling the developed products and services to ensure its quality and consistency. The current research agreed to use the company's existing product management system with some minor additional development, which included the versioning control of the products deployed. The guidelines outlined in Chapter Four were then used to explain the steps needed to implement the CEF Product Line model.

i. Define part number convention.

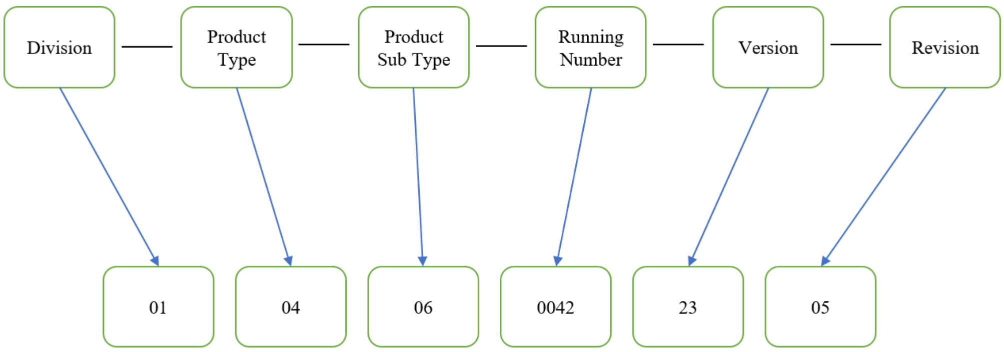



*Figure 5.12* Part Number Convention

In the Product Control component, the overall concept is to ensure that each product deployed is accountable. Based on Figure 5.12, the part number convention was defined. The definition of each convention is explained as per below:

a) Division = Department or Organization Number

b) Product Type = Product Family Group Identification

c) Sub Product Type = Actual Product Identification based on product specification

d) Running Number = A random running number

e) Version = Version Numbering for the specified product

f) Revision = Revision number for any modification or customization done

ii. Define serial number convention.

Once the part number convention has been decided, the serial number convention is then configured. Both the part number and the serial number are issued through the Product Management system. To generate the part number and even serial number, the system requires a set of input requirement to be entered accordingly.

iii. Define version control procedures.

To handle the version control aspect for the deployed products, the process flow has been defined to release a specific product or a customized version of the standard software. Figure 5.13 shows that there are several layers of testing to be conducted even before the product can be released to the production server.

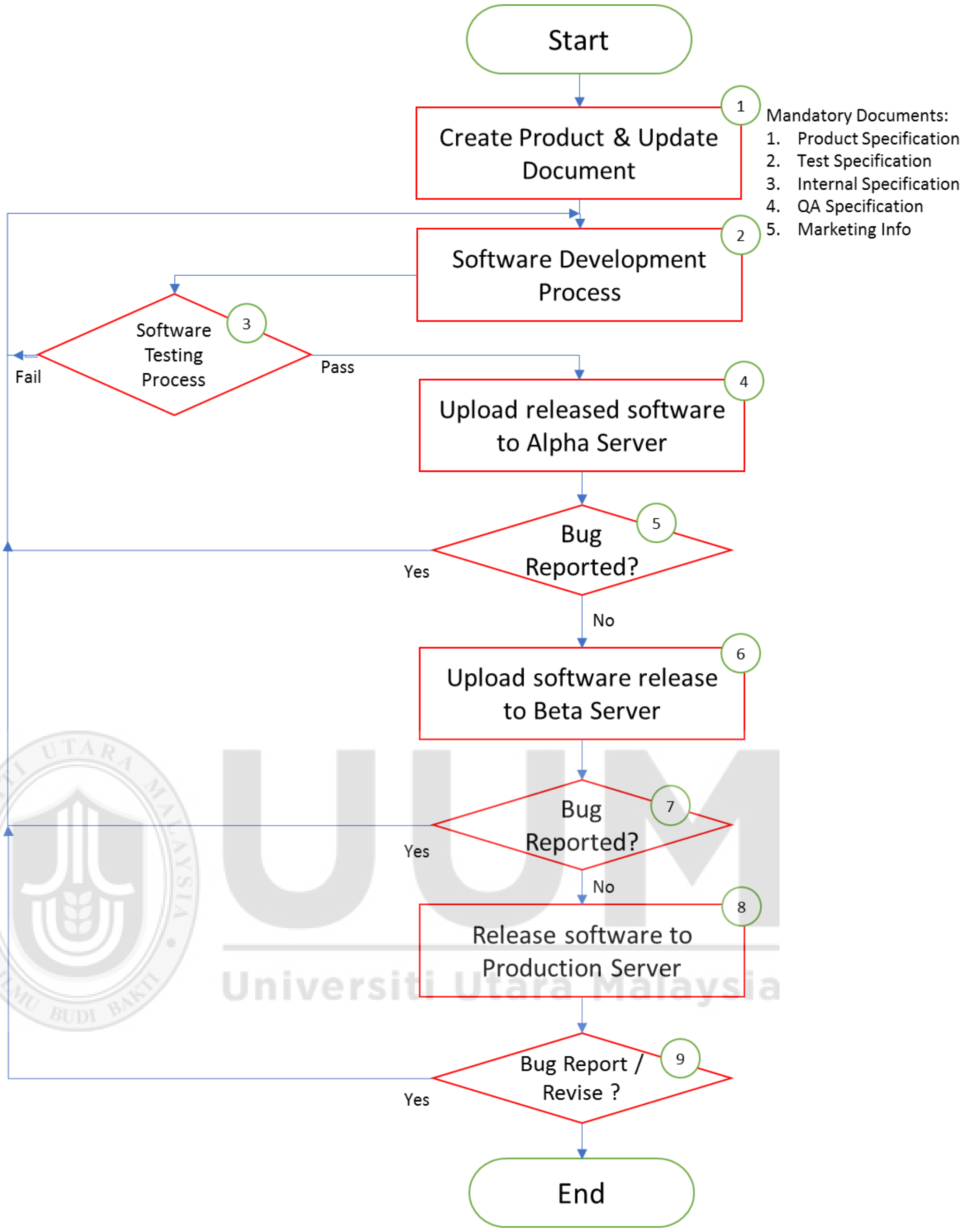



*Figure 5.13* Software Release Procedure

iv. Define a mechanism of inconsistent update proof.

In terms of software updates, a Marketplace was constructed where all the products are hosted. Each client is able to download any of the updated products but it would only function with the activation key. The activation key is provided once payment is made as a security measure.

v. Define software release procedures.

It has been defined that the software release for any new system development will go through a release protocol using the product management system.

vi. Define software testing procedures.

It has been defined that the software testing procedures will be based on the Usability Evaluation Method. In Table 5.11 is a sample of a software usability testing checklist.

Table 5.11

*Sample Software Usability Checklist*

| No | Usability Test Steps | Yes | No |
|---|---|---|---|
| 1 | Is the control of cursor compatible with movement? | | |
| 2 | Are the results of control entry compatible with user expectations? | | |
| 3 | Is the control matched to user skill? | | |
| 4 | Is the coding compatible with familiar conventions? | | |
| 5 | Is the wording familiar? | | |

vii. Define quality assurance mechanism.

Quality assurance items have been embedded in the CEF Model subcomponent, those are version, revision, and release control. A product management system has been built that takes care of all the version and revision control for the developed products.

viii. Create a system as per the CEF Workflow requirement.

A system called Job Management system with built-in audit trail was created to ensure that the system enforces the defined processes described in the CEF Workflow model.

### 5.4.5 Knowledge Management

The Knowledge Management model dictates methods to allow the knowledge amassed in the company to be retained and management properly. To do this, the researcher recommended for the development of a Learning Management System to handle the Knowledge Management requirement to implement the CEF Model. The guidelines outlined in Chapter Four were used to explain the steps needed to implement the Knowledge Management model.

i. Define technology knowledge, skill knowledge and operation knowledge.

   The three types of knowledge to be used in the Learning Management System were defined. In Table 5.12 are the samples of knowledge types that were defined by the current research endeavor.

Table 5.12
*Sample Knowledge Types*

| No | Knowledge Type | Sample Knowledge |
|---|---|---|
| 1 | Technology Knowledge | HTML, ASP, PHP, Java, Database |
| 2 | Skill Knowledge | Implement New System, End user training |
| 3 | Operation Knowledge | Generate Invoice, Generate Quotation |

ii. Define method of knowledge accessibility within the assigned job parameters.

   The knowledge repository has been defined as able to be accessed through multiple methods.

iii. Define the new knowledge logging procedures.

It was defined that with every new discovery of knowledge with regards to a product, technological advancement, or operation knowledge, that knowledge can be logged dynamically into the knowledge management system.

iv. Create a system as per the Knowledge Management model requirement.

A system called Learning Management System was developed to cater for the requirements set in the Knowledge Management model. The system is developed to allow for staff to have on-demand learning of the courses to assist in completing their day to day job.

### 5.4.6 Summary of Requirement Specifications

After the subcomponent elements were defined by the guidelines provided in Chapter Four, the requirement specification to implement the CEF Model for the company was summarized as illustrated in Table 5.13.



Table 5.13

*Summary of CEF Model Environment Requirement Specification*

| CEF Model Components | Requirement Specification |
|---|---|
| CEF Product Line<br>CEF Platform Architecture | • Use existing company's product and pseudo product<br>• Verified that the company's existing platform is able to support the CEF Model – contained user accessibility management, security options, data log capacity<br>• Create multiple instances of the platform on the Cloud as redundancy and backup |
| CEF Workflow | • Job Management System with Dashboard<br>• Create jobs for Work Centers complete with task lists<br>• Work Order templates based on the specific Work Centers<br>• All system inputs go through helpdesk management system – convert valid requests into Work Order – linked with Work Order templates<br>• Work Order templates with priority level settings distributes jobs to assigned Work Centers<br>• Create flexible priority level settings in Work Order template |
| CEF Product Control | • Use company's current Product Management System application to enforce CEF Product Control environment<br>• Decided on the Part Number convention for company's products and services |

Table 5.13 continued.

| | |
|---|---|
| Knowledge Management | • LMS (Learning Management System)<br>• Student Portal/Dashboard – allow employee to register for courses depending on the necessity<br>• Programs and Examination modules<br>• Each department compiled and upload its own course and syllabus in the LMS system<br>• Course Info that includes Project as a course name<br>• Project information to be stored in LMS as well<br>• Linked course topics with specific task lists in the Job Management System |

## 5.5 CEF Model to System Development

The purpose of this phase is to simulate the existing Cloud ERP production environment into the CEF concept by using the requirement specification formulated earlier. The actual prototype CEF Model consists of a Cloud ERP production cycle that is primarily a process oriented software development with software system tools that act like machineries in a manufacturing facility. The developed CEF tools were designed to be operated by human resources such as programmers, analysts, and support admins, which the current study refers to as actors. The subsequent sub-topics elaborate further the actual prototype environment for each component that makes up the entire CEF Model. The first two components of the CEF Model are a prerequisite for the implementation of a CEF environment but do not involve the creation of a specific system or software. Workflow, Product Control, and Knowledge Management components, however, were translated into system tools.

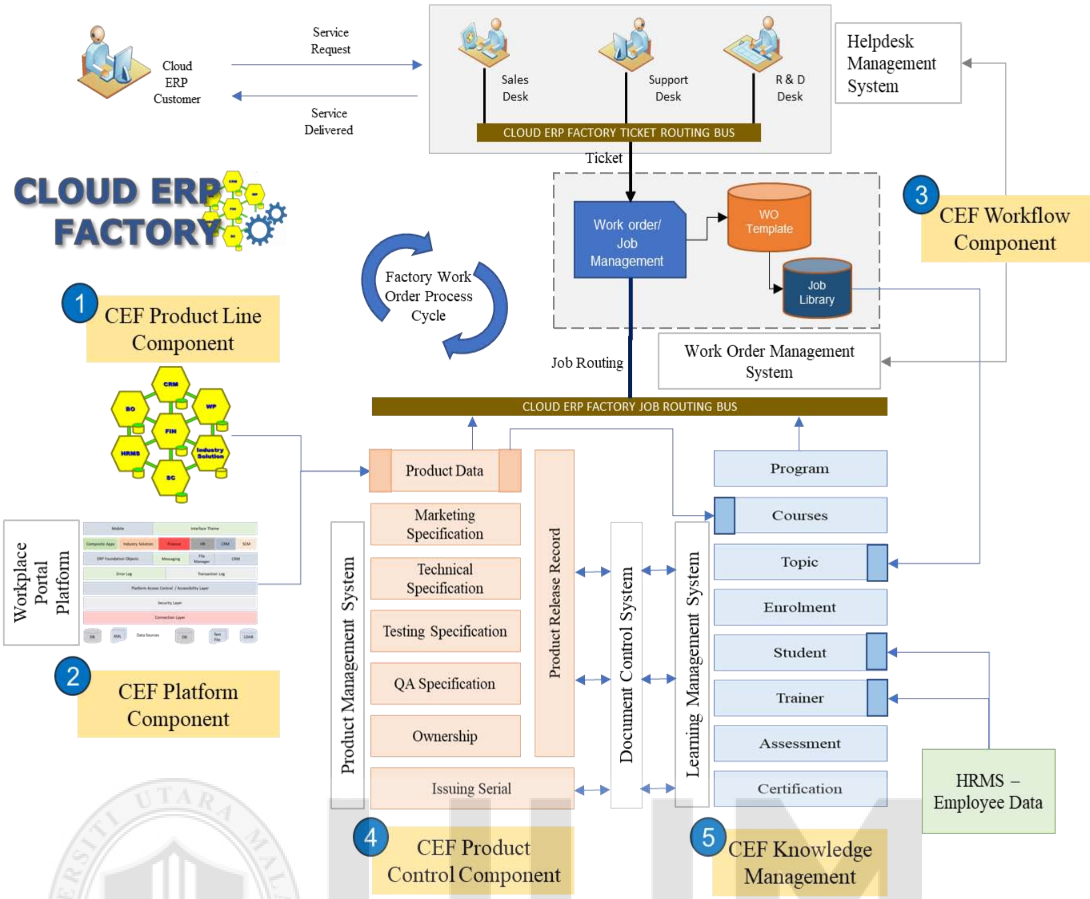



*Figure 5.14* Overall View of CEF Model Prototype Environment

Figure 5.14 shows the CEF Model prototype environment that was configured to be implemented into the selected company. The CEF Product Line and Platform component did not require any new system or tools since the company had its own existing products and platform, which were linked to CEF Product Control and CEF Knowledge Management by utilizing a Document Control System. Helpdesk and Work Order Management system makes up the CEF Workflow component. The Product Management System, which represents the CEF Product Control component manages all the product data and release methods. The Learning Management System was developed to represent CEF Knowledge Management component.

### 5.5.1 CEF Product Line

Based on the current Cloud ERP production environment and the CEF model design guideline, the outcome of this model development was to define the type of product and product grouping for the selected Cloud ERP production. The defined product and product grouping were stored in the Product Control system tool. Table 5.14 lists the existing main product grouping of the company that satisfies the requirement for a CEF environment.

Table 5.14
*Product Grouping*

| Item | Product Grouping | Product Name |
|---|---|---|
| 1 | Customer Relationship Management (CRM) | Helpdesk Management System<br>Customer Account Management System (CAMS)<br>Customer Portal |
| 2 | Supply Chain Management (SCM) | Supplier Account Management System (SAMS) |
| 3 | Financials | Financial Information System |
| 4 | Industry Solution | Manufacturing System<br>Facility Management System |

### 5.5.2 CEF Platform

The CEF Platform component was fulfilled with the User Portal Platform and with the System Architecture and Backup/Redundancy ability of the existing platform. However, no special tool was developed solely for this component, as the platform component depicted was more like a requirement to promote the component reusability function.


*Figure 5.15* Smart Lab Workplace User Platform

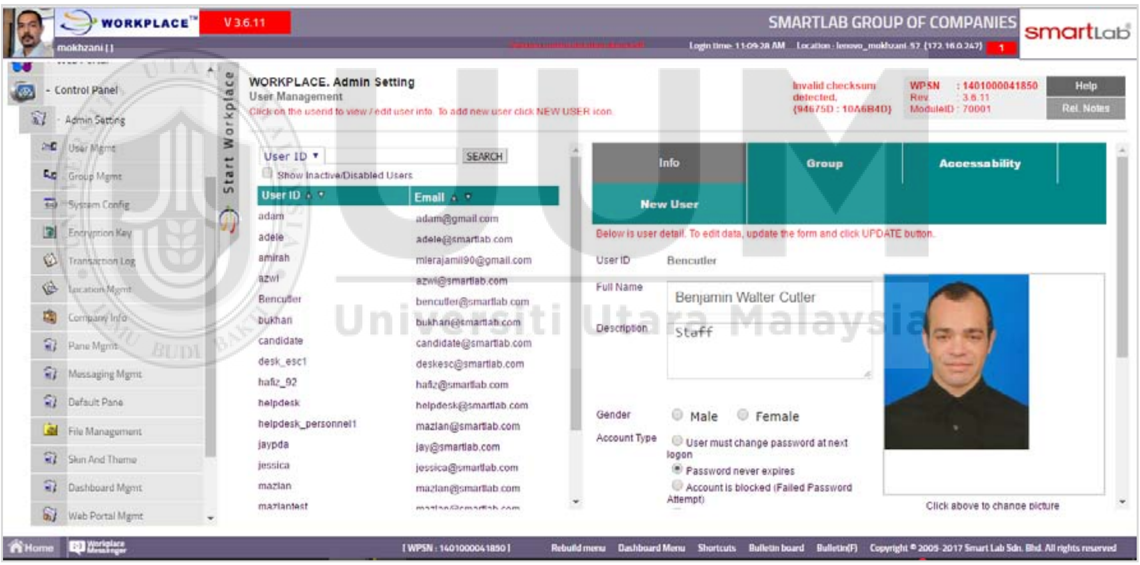



*Figure 5.16* User Management in Company an Existing Platform

Figure 5.15 and 5.16 are snapshots of the existing Workplace platform. The platform contains all of the features explained in CEF Platform guidelines such as single sign-on portal, user management, and accessibility configuration. The existing platforms have proven themselves to meet the requirements of our CEF Model environment, so no additional system or tools were needed to be developed.

### 5.5.3 CEF Workflow

Based on the design guidelines walkthrough result explained in section 5.3, a tool was developed to satisfy the requirement for the simulation of CEF Workflow. The system was developed to control and manage, while providing a mechanism to emulate the processes based on the CEF Model. The following subsection briefly describes the important features of the developed system tool.

#### 5.5.3.1 Workflow System Architecture

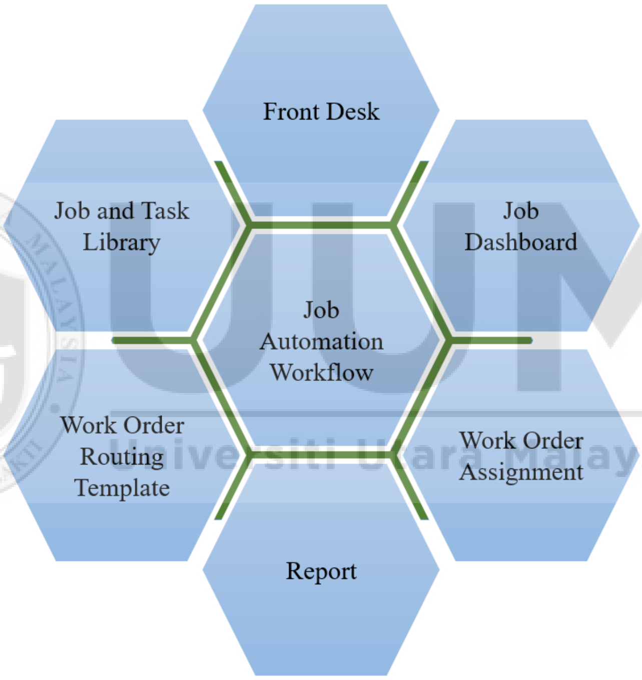



*Figure 5.17* CEF Workflow System Architecture Model

The CEF Workflow system consists of several modules as referred to Figure 5.17. The Front Desk module captures all the system inputs through ticketing and case logging. The ticket logged can be converted into jobs are displayed in the Job Dashboard module, which then be attended to by the specified personnel. The Job and Task library, Work Order Routing template, and Work Order Assignment were developed to set up and manage the library of tasks and jobs defined in the Cloud ERP production environment.

#### 5.5.3.2 Workflow System Database

The Figure 5.18 is a representation of the system database diagram used to develop the CEF Workflow tool.

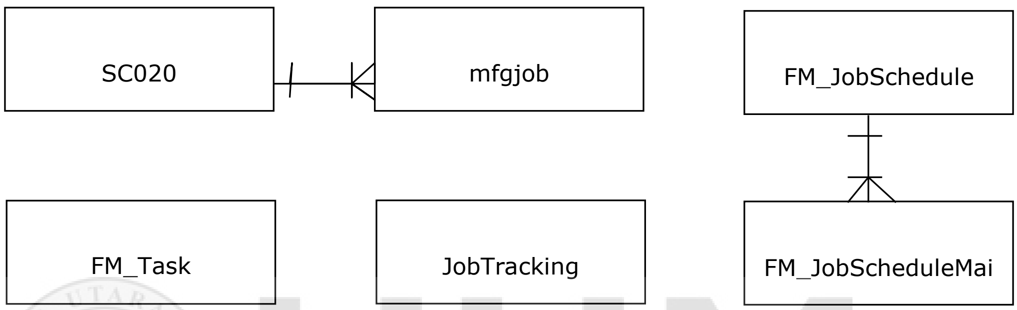



*Figure 5-18* CEF Workflow System Database Diagram

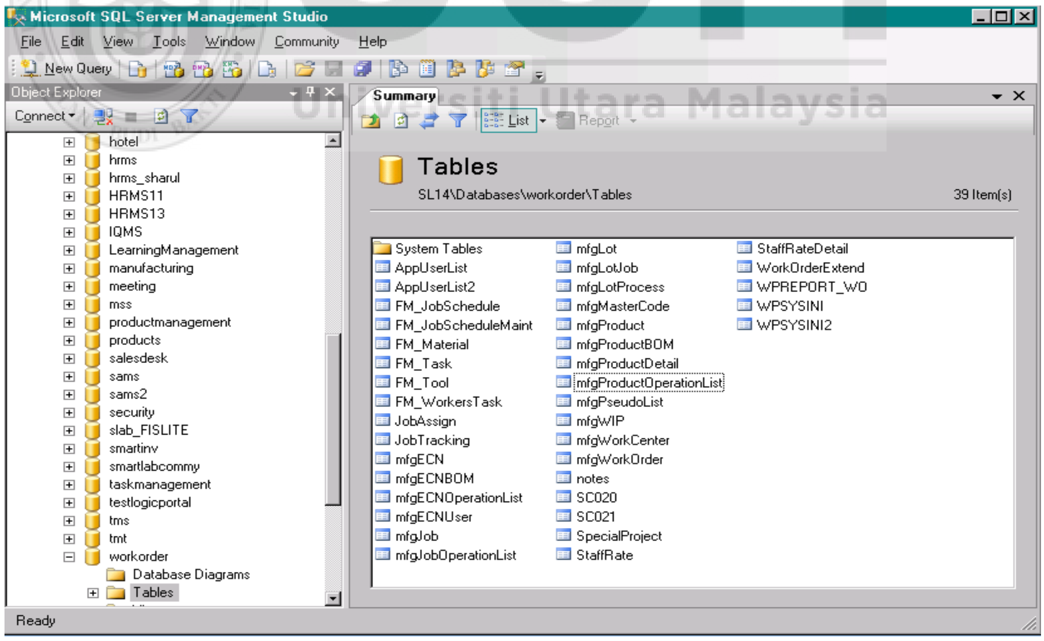



*Figure 5.19* CEF Workflow Tables Listing

In Figure 5.19, the CEF Workflow tables used in the development of the system tool are shown.

### 5.5.3.3 Workflow Application Menu

Table 5.15 illustrates the type of menu developed for the CEF Workflow system tool.

Table 5.15

*Job Management Application Menu Table*

| | |
|---|---|
| **JOB MANAGEMENT** | **Job Library** |
| | WO Template |
| | WO Assignment |
| | My Job |
| | Report |
| | Front Desk |

### 5.5.3.4 Workflow System Snapshot

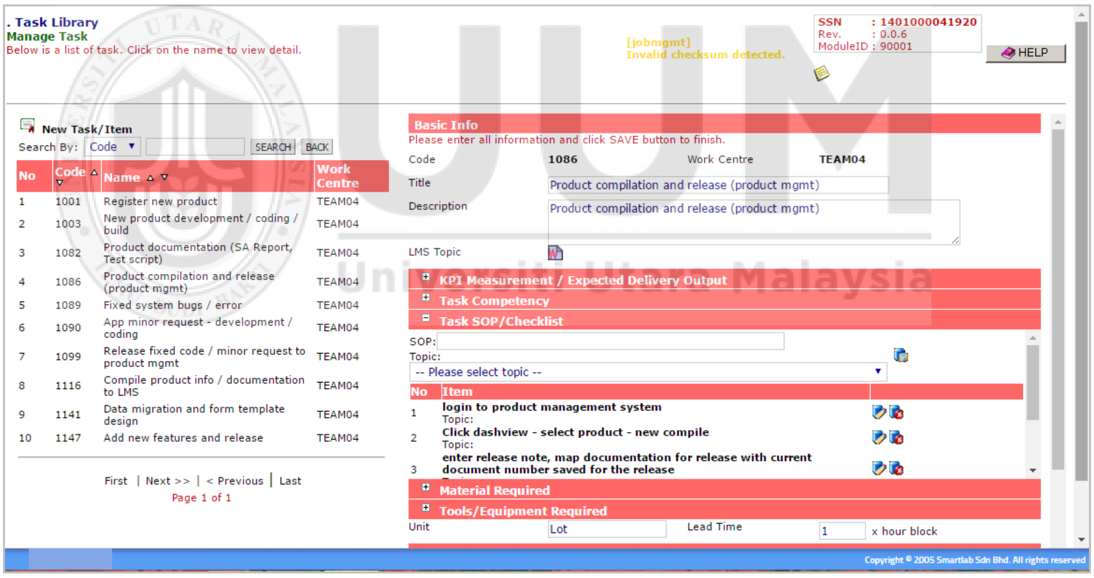



*Figure 5.20* Job and Task Library Set Up

Figures 5.20 and 5.21 show the display for Job and Task Library set up. This module of defining jobs and tasks is a direct correlation to the Job and Task element described in the CEF Workflow model. Each job contains multiple task lists or Standard Operating Procedures (SOP). In this set up, other variables can also be entered in, such as Expected Delivery Output, Task Competency, Material, and Tools Required.

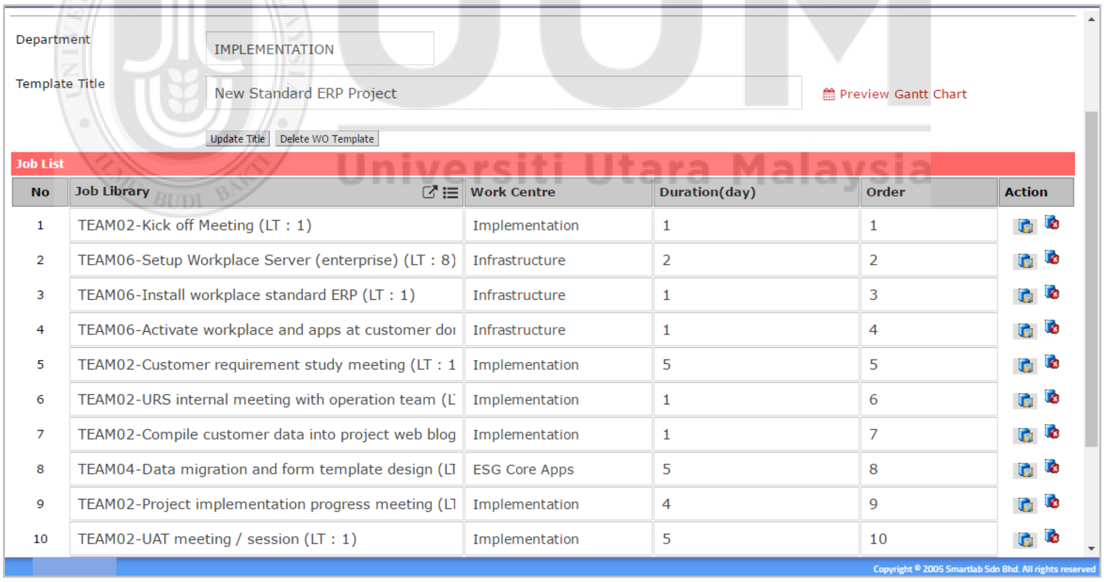
```
if mode="NEW" then
    rsmm6.open "select * from fm_task where tcode='"&tcode&"' and tname='"&tname&"'"
    if not rsmm6.eof then%>
        <script>
        alert("Sorry, this information is already exist in the list (<%=tcode%>). Please try again")
        self.history.back()
        </script><%
    else
        rsmm5.open "select * from fm_task order by convert(int,tcode) desc"
        if not rsmm5.eof then
            newtcode = rsmm5("tcode") + 1
        else
            newtcode = 1001
        end if
        rsmm5.close


        tid = getrunningno("id_task")
        cmdmm.commandtext= "insert into fm_task (tid,compid,parentid,tcat,tcode,tname,ttype,tdesc,tnoofw,tunit,trate,trate2,trate3,trate4,tlt,tltunit,syncb,synca) values "&_
        "('"&tid&"','"&appscompid&"','"&parentid&"','"&tcat&"','"&newtcode&"','"&tname&"','"&ttype&"','"&tdesc&"','"&tnoofw&"','"&tunit&"','"&trate&"','"&trate2&"','"&trate3&"',
        cmdmm.execute
        newlocid = updaterunningno("id_task",tid + 1)
        value = "insert"
        session("fcid")=tid
    end if
    rsmm6.close


end if '------------------------------------mode

task = value
```

*Figure 5.21* Job and Task Library Set Up Coding




| No | Job Library | Work Centre | Duration(day) | Order | Action |
|---|---|---|---|---|---|
| 1 | TEAM02-Kick off Meeting (LT : 1) | Implementation | 1 | 1 | |
| 2 | TEAM06-Setup Workplace Server (enterprise) (LT : 8) | Infrastructure | 2 | 2 | |
| 3 | TEAM06-Install workplace standard ERP (LT : 1) | Infrastructure | 1 | 3 | |
| 4 | TEAM06-Activate workplace and apps at customer do | Infrastructure | 1 | 4 | |
| 5 | TEAM02-Customer requirement study meeting (LT : 1 | Implementation | 5 | 5 | |
| 6 | TEAM02-URS internal meeting with operation team (L | Implementation | 1 | 6 | |
| 7 | TEAM02-Compile customer data into project web blog | Implementation | 1 | 7 | |
| 8 | TEAM04-Data migration and form template design (L | ESG Core Apps | 5 | 8 | |
| 9 | TEAM02-Project implementation progress meeting (L | Implementation | 4 | 9 | |
| 10 | TEAM02-UAT meeting / session (LT : 1) | Implementation | 5 | 10 | |

*Figure 5.22* Work Order Routing Template



Figures 5.22 and 5.23 illustrate the set up for Work Order routing template. The setting up for the Work Order routing template was associated with the combination of Work Order Routing element defined in the model development of CEF Workflow model. Each work order routing created can consists of different jobs from different Work Centers. During the setup, users were able to dictate the duration and the order sequence for the jobs listed for the Work Order.

```
rsmm1.open "select * from fm_jobschedule where jsid="&request.querystring("jsid")&""
if not rsmm1.eof then

if request.querystring("asid") <> "" then
    ntasid = request.querystring("asid")
else
    ntasid = rsmm1("asid")
end if

cmdmm.commandtext = "insert into fm_jobschedule (jsid,compid,jsname,jsdesc,asid,jsrefno,jsjobleader,jsrecurtype,jsdate,jstime,jstrday,jsedate,jsqt
"("&jsid&",'"&appscompid&"','"&replace(rsmm1("jsname"),"'","")&"','"&replace(rsmm1("jsdesc"),"'","")&"','"&ntasid&"','','',"&_
"'"&rsmm1("jsrecurtype")&"','"&rsmm1("jsdate")&"','"&rsmm1("jstime")&"','"&rsmm1("jstrday")&"','"&rsmm1("jsedate")&"','"&rsmm1("jsqty")&"','"&rsmm
cmdmm.execute

    rsmm2.open "select * from fm_jobschedulemaint where compid='"&appscompid&"' and jsid='"&request.querystring("jsid")&"'"
    if not rsmm2.eof then
    do while not rsmm2.eof
        cmdmm.commandtext = "insert into fm_jobschedulemaint (jsid,compid,mid,orderno,status,syncb,synca) values "&_
        "('"&jsid&"','"&appscompid&"','"&rsmm2("mid")&"','"&rsmm2("orderno")&"','Y','"&session("memberid")&"','"&request.cookies("location")&"')"
        cmdmm.execute
    rsmm2.movenext
    loop
    end if
    rsmm2.close

end if
rsmm1.close

if ntasid="0" then
    ntemplate = "Y"
else
    ntemplate = ""
```

*Figure 5.23* Work Order Routing Template Coding


*Figure 5.24* Helpdesk Request Enquiry Form

Figures 5.24 and 5.25 display the Helpdesk Request Enquiry form. This helpdesk request is the first step as it defines the type of System Input mentioned in the model development to be entered into the Job Management System. The ticket raised from this case log is then sent to the CEF Workflow Job Dashboard based on the assigned group or personnel selected during the case log process.

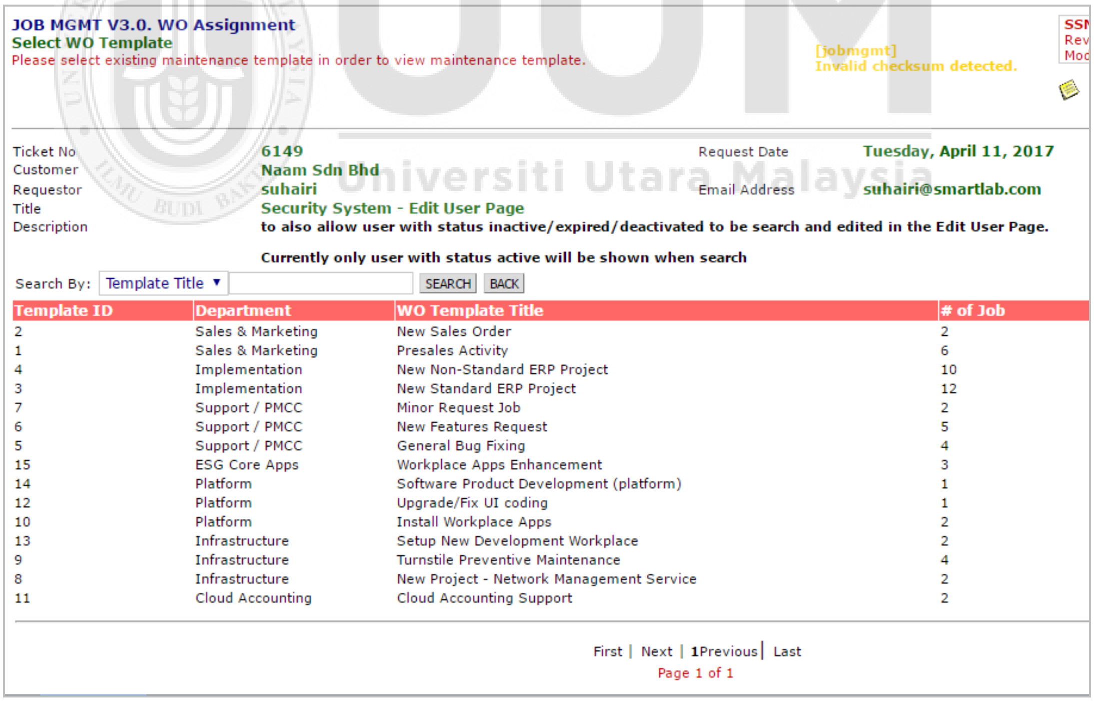
```
<%
    set dparm = server.CreateObject("scripting.dictionary")
    dparm.Add "DIV", div
    dparm.Add "REQUESTID", requestid
    dparm.Add "PREVIOUSID", previousid
    dparm.Add "REQUESTMODE", custid
    dparm.Add "CONTRACTID", contractid
    dparm.Add "ASSETID", assetid
    dparm.Add "REQUESTOR", requestor
    dparm.Add "EMAIL", email
    dparm.Add "CCEMAIL", ccemail
    dparm.Add "TITLE", title
    dparm.Add "DESCRIPTION", description
    dparm.Add "GROUPID", groupid
    dparm.Add "RESPONSELEVEL", responselevel
    dparm.Add "ATTACH1", attach1
    dparm.Add "REQUESTREF", request.querystring("wfcat")
    dparm.Add "SUPPORTREF", supportref
    dparm.Add "RDATE", rdate
    'dparm.Add "MODE", mode
    'set dparm = nothing

    if action = "edit" then
        dparm.Add "MODE", "EDIT"
        Process = Help.Int_Request(dparm)
        set dparm = nothing

    elseif action = "delete" then
        dparm.Add "MODE", "DELETE"
        Process = Help.Int_Request(dparm)
        set dparm = nothing
    else
        dparm.Add "MODE", "New"
        Process = Help.Int_Request(dparm)
        set dparm = nothing %>
```

*Figure 5.25* Helpdesk Request Enquiry Form Coding





JOB MGMT V3.0. WO Assignment
Select WO Template
Please select existing maintenance template in order to view maintenance template.

[jobmgmt]
Invalid checksum detected.

| | | | |
|---|---|---|---|
| Ticket No | 6149 | Request Date | Tuesday, April 11, 2017 |
| Customer | Naam Sdn Bhd | | |
| Requestor | suhairi | Email Address | suhairi@smartlab.com |
| Title | Security System - Edit User Page | | |
| Description | to also allow user with status inactive/expired/deactivated to be search and edited in the Edit User Page. Currently only user with status active will be shown when search | | |

Search By: Template Title SEARCH BACK

| Template ID | Department | WO Template Title | # of Job |
|---|---|---|---|
| 2 | Sales & Marketing | New Sales Order | 2 |
| 1 | Sales & Marketing | Presales Activity | 6 |
| 4 | Implementation | New Non-Standard ERP Project | 10 |
| 3 | Implementation | New Standard ERP Project | 12 |
| 7 | Support / PMCC | Minor Request Job | 2 |
| 6 | Support / PMCC | New Features Request | 5 |
| 5 | Support / PMCC | General Bug Fixing | 4 |
| 15 | ESG Core Apps | Workplace Apps Enhancement | 3 |
| 14 | Platform | Software Product Development (platform) | 1 |
| 12 | Platform | Upgrade/Fix UI coding | 1 |
| 10 | Platform | Install Workplace Apps | 2 |
| 13 | Infrastructure | Setup New Development Workplace | 2 |
| 9 | Infrastructure | Turnstile Preventive Maintenance | 4 |
| 8 | Infrastructure | New Project - Network Management Service | 2 |
| 11 | Cloud Accounting | Cloud Accounting Support | 2 |

First | Next | 1Previous | Last
Page 1 of 1

*Figure 5.26* Work Order Template Selection

Figures 5.26 and 5.27 depicts the Work Order Template listing and selection. This is the culmination of the core model elements of Work Center and Job and Task

Library together with the Work Order Routing to generate a system, which is able to enable the automation of job creation based on the type of Work Orders selected. Any approved request is converted into a Work Order, which contains a job or jobs depending on the template selected. Once confirmed, these jobs will then be distributed according to the Work Centers affected.

```
SQL="select * from fm_jobschedule where compid='"&appscompid&"' and jscat like '%"&sfield2&"%' and status<>'X' and jsrefno='TEMPLATE'
'response.write sql
RS.Open SQL, MyConn, adOpenStatic, adLockReadOnly, adCmdText    %>

If not Rs.eof Then
Page_Count = RS.PageCount

If 1 > Current_Page Then Current_Page = 1
If Current_Page > Page_Count Then Current_Page = Page_Count

RS.AbsolutePage = Current_Page
no = 1
Do While RS.AbsolutePage = Current_Page And Not RS.EOF

<tr>
<td width="9%" align="left" height="4"><p dynamicanimation="fpAnimformatRolloverFP1" fprolloverstyle="background-color: #c0c0c0; font-
<a href="<%=rfile%>jsid=<%=Rs("jsid")%>&requestid=<%=request("requestid")%>" ><font class="applabel"><%=Rs("jsid")%></font></a></p></t
<td width="10%" align="left" height="4"><p dynamicanimation="fpAnimformatRolloverFP1" fprolloverstyle="background-color: #c0c0c0; font
<a href="<%=rfile%>jsid=<%=Rs("jsid")%>&requestid=<%=request("requestid")%>" ><font class="applabel"><%=getname("department",Rs("jsdes
<td width="27%" align="left" height="4"><p dynamicanimation="fpAnimformatRolloverFP1" fprolloverstyle="background-color: #c0c0c0; font
<a href="<%=rfile%>jsid=<%=Rs("jsid")%>&requestid=<%=request("requestid")%>" ><font class="applabel"><%=Rs("jsname")%></font></a></p><
<td width="22%" align="left" height="4"><p dynamicanimation="fpAnimformatRolloverFP1" fprolloverstyle="background-color: #c0c0c0; font
<a href="<%=rfile%>jsid=<%=Rs("jsid")%>&requestid=<%=request("requestid")%>" ><font class="applabel"><%=noofj%></font></a><
</tr>
<%
Rs.MoveNext
no = no+1
Loop
sumbil = no
Rs.Close%>
```

*Figure 5.27* Work Order Template Selection Coding

Figures 5.28, 5.29(a) and 5.30(b) continues the process of selecting the appropriate Work Order routing. This represented the output of the Work Order Routing element providing the necessary tools for the user to automate the process flow. The admin or personnel in charge was able to change the available variables such as the designated person, planned duration of the job completion, as well as the order sequence of the job to be completed. Some of the jobs can have simultaneous order sequences, depending on the type of job required for recurring or common Work Order types.

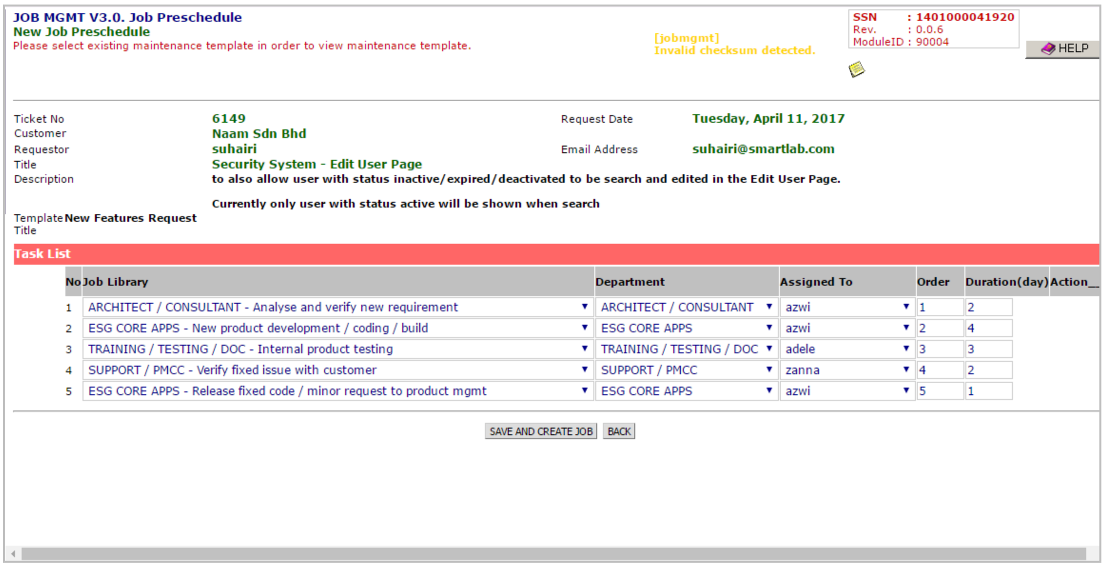



*Figure 5.28* Job Management and Scheduling

```
'2. '''''''''''''''''''''''''''''check workorder''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        rsmfg1.open "select * from sc020 where quotationno='HD' and lastmodifystaff='"&request("contractid")&"' " '---and pono = '
        if rsmfg1.eof then
            woid=getnewwono()
            rsmfg2.open "select a.*, b.orgname from (select * from "&hlpname&".dbo.contract where contractid='"&request("contracti
                Cmdmfg.commandtext = "insert into SC020 (woid,quotationno,pono,custpono,custid,Projectname,OrgName,contactperson,r
                "('"&woid&"','HD','spt"&request("requestid")&"','"&request("pic")&"','"&rsmfg2("custid")&"','"&rsmfg2("contractnam
                response.write "<br>"&Cmdmfg.commandtext
                Cmdmfg.execute
            rsmfg2.close
        else
            woid=rsmfg1("woid")
        end if
        rsmfg1.close

'3. '''''''''''''''''''''''''''''check job category (auto create job base on template)''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        rsmfg1.open "select * from "&hlpname&".dbo.trans_request where requestid='"&request("requestid")&"'"
        reqtitle = rsmfg1("title")
        rsmfg1.close

        if request("recurtype")<>"R01" then
        ''''''''''''''''''''''''''''''''''''''''create recurring job'''''''''''''''''''''''''''''''''
                rsmm3.open "select * from mfgmastercode where category='recurring type' and code='"&request("recurtype")&"'"
                if not rsmm3.eof then
                    rccid = rsmm3("value1")
                    rccno = rsmm3("value2")
                end if
                rsmm3.close
                noofday = datediff(rccid,request("rsdate"),request("redate"))
                jsdate = request("rsdate")
                for i = 0 to noofday
                    jsdate2 = dateadd(rccid,i*rccno,jsdate)
                    ''response.write "<br><b>create job at date "&jsdate2&"<br>"
```

*Figure 5.29* Job Management Listing Coding (a)

```
else
'''''''''''''''''''''''''''''''''''''''create single job'''''''''''''''''''''''''''''''

''''''''''''''''''''''''''''''''create main job''''''''''''''''''''''''
        newjobno = getnewjobno(woid)

        CmdMfg.commandtext = "insert into mfgjob (jobno,pano,name,jobcat,productno,qty,department,jobmode,description,requestdate,pic,jo
        "('"&newjobno&".0','"&woid&"','<b><u>MAIN</u></b> - "&replace(reqtitle,"'","")&"','"&request("jobcat")&"','HDT"&request("request
        "'"&replace(request("description"),"'","")&"','"&date()&"','SYSTEM','2','1','"&date()&"','5','','OPEN','"&session("memberid")&"'
        ''response.write "<br>"&Cmdmfg.commandtext
        CmdMfg.execute

        rsmfg1.open "select a.*, b.tname, b.tcat from (select mid,orderno,remark,ex1 from fm_jobschedulemaint where jsid='"&request.quer
        if not rsmfg1.eof then
            biljb = 1
            do while not rsmfg1.eof

                if bil > 1 then
                    jjsdate = cdate(jjedate)
                    jjedate = dateadd("d",rsmfg1("remark"),jjsdate)
                else
                    jjsdate = date()
                    jjedate = dateadd("d",rsmfg1("remark"),date())
                end if
                jobpic = rsmfg1("ex1")

                CmdMfg.commandtext = "insert into mfgjob (jobno,pano,name,jobcat,productno,qty,department,jobmode,description,requestdat
                "('"&newjobno&"."&biljb&"','"&woid&"','<b><u>"&replace(rsmfg1("tname"),"'","")&"</u></b> - "&replace(reqtitle,"'","")&"'
                "'"&replace(request("description"),"'","")&"','"&jjsdate&"','"&jobpic&"','3','"&replace(rsmfg1("remark"),"'","")&"','"&j
                ''response.write "<br>"&Cmdmfg.commandtext
                CmdMfg.execute


'''''''''''''''''''''''''''push data to helpdesk support''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        sptdesc = replace(getsptdesc(),"*job*",newjobno)
        sptdesc = "<b>"&rsmfg1("tname")&"</b> . "&replace(sptdesc,"*pic*",jobpic)
```

*Figure 5.30* Job Management Listing Coding (b)

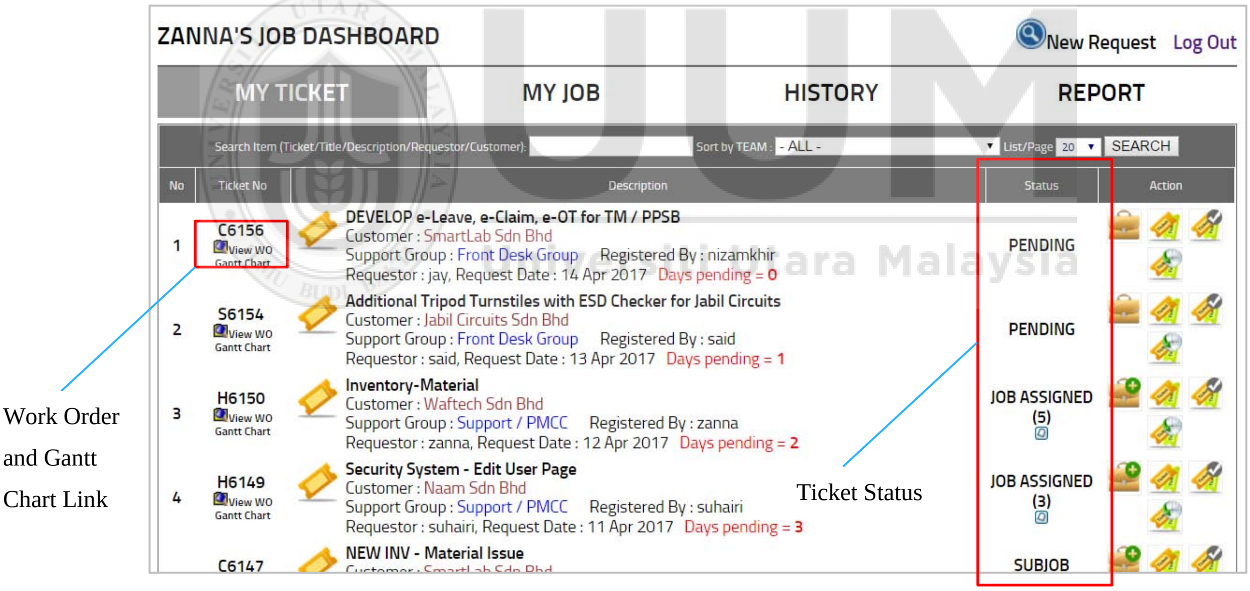



*Figure 5.31* Helpdesk Request Ticket Dashboard View

Based on the requirement described in Chapter Four, all of the system inputs would be channeled through a bus, and to do this, a Helpdesk Request Dashboard was developed, as shown in Figure 5.31.

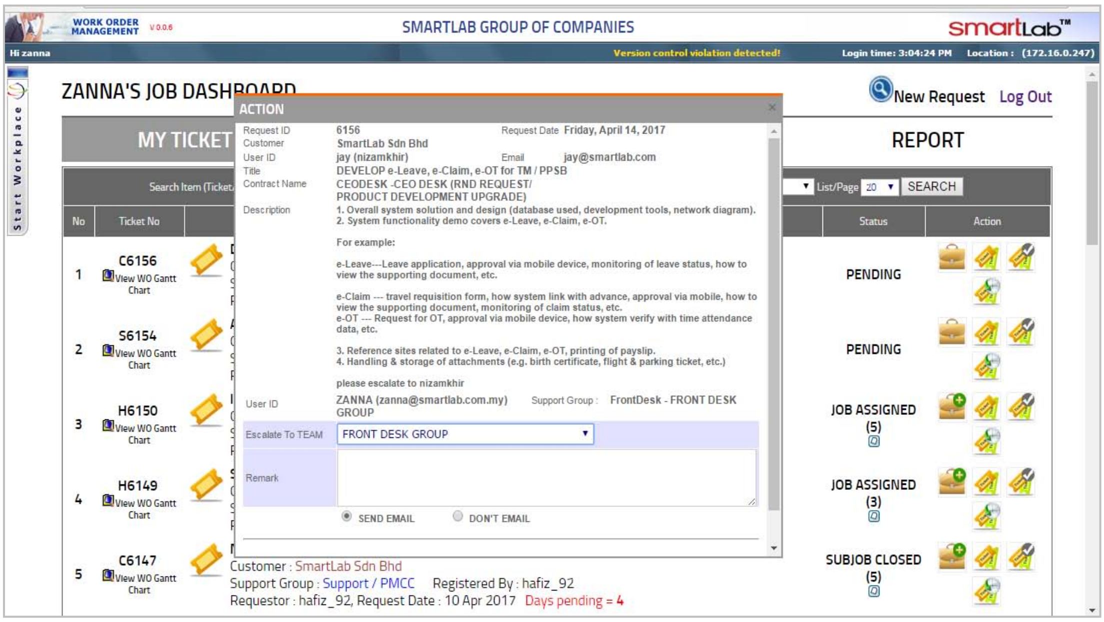



*Figure 5.32* Helpdesk Request Ticket Action Item

In Figure 5.32, users of the Helpdesk Request were able to log the cases and proceeded with the necessary action items to ensure that the ticket is closed.

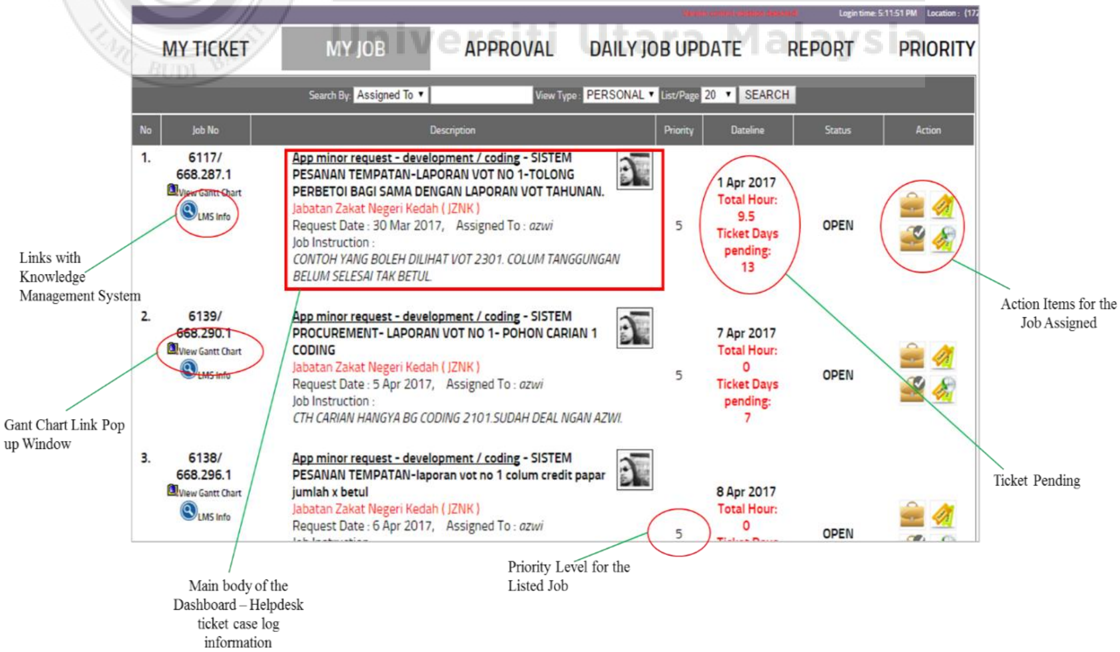



*Figure 5.33* Job Dashboard View

The System Output element was achieved with the development of the Job Dashboard as shown in Figures 5.33 and 5.34. By combining each of the CEF Workflow model elements' outputs, the Job Dashboard was able to provide the necessary tools and information to be used by a Cloud ERP Production environment. To further enhance the features of the dashboard, the module was combined with the ability to view a Gantt Chart based on the project's start date and users were also able to view all the necessary information and data related to the project or product by a linkage to the Learning Management system.

```
 Job Instruction :<br /><i><%=replace(rs("description"),vbcrlf,"<br />")%></i><br />
   <%if attach1 <> "" then%>
   <a href="/fm_helpdesk/<%=custname%>/<%=attach1%>" target="_blank">
   <img src="/helpdesk/images/detail.gif" width="14" height="14"><font class="applabel"><b>View Attachment</b></font></a>
<%end if%><%custname=""%><br />
 </div>
 </td>
<td align="center" <%=tdcc%> ><%=rs("priority")%></strong></td>
<td align="center" <%=tdcc%> ><strong><%if rs("dateline")<>"" then%><%=convertdate(rs("dateline"))%><%end if%>
<br><font color=red>Total Hour:<br><%=uahour%><%uahour=0%></font>
<br><font color=red>Ticket Days pending:<br><b><%=datediff("d",rs("requestdate"),date())%></b></font><br/></strong></td>

<td align="center" <%=tdcc%> ><strong><%=jjstat%><br><%=statlink%></strong></td>
<td  <%=tdcc%> align="center">
<%if jjstat <> "WAITING" then%>
   <%if mid(rs("jobno"),len(rs("jobno"))-1,len(rs("jobno")))=".0" then
      if jobappr="Y" then%>
      <img onclick="popurl(this,'/workorder/1azj_detail.wp?mode=escalate&jobno=<%=rs("jobno")%>')" src="images/btn-escalate-ticket
      <%end if
      if jmainclose="Y" or session("memberid")="azwi" then%>
      <img onclick="popurl(this,'/workorder/1azj_detail.wp?act=mainapprove&jobno=<%=rs("jobno")%>')" src="images/btn-closejob.png'
      <%end if

   elseif sfield<>"all" then%>
      <%if rs("status")="OPEN" then%>
      <img onclick="popurl(this,'/workorder/1azj_detail.wp?mode=update&jobno=<%=rs("jobno")%>')" src="images/btn-job.png" class="a
      <img onclick="popurl(this,'/workorder/1azj_detail.wp?mode=escalate&jobno=<%=rs("jobno")%>')" src="images/btn-escalate-ticket
      <img onclick="popurl(this,'/workorder/1azj_detail.wp?mode=close&jobno=<%=rs("jobno")%>')" src="images/btn-closejob.png" clas
      <%end if%>
   <%end if%>
<%end if
if jjstat = "WAITING" and jobappr="Y" then%>
<img onclick="popurl(this,'/workorder/1azj_detail.wp?mode=escalate&jobno=<%=rs("jobno")%>')" src="images/btn-escalate-ticket.png" cl
<%end if%>
<img onclick="popurl(this,'/helpdesk/frontdesk/frontdesk_viewdetail.asp?requestid=<%=replace(rs("productno"),"HDT","")%>')" src="ima
</td>
```

*Figure 5.34* Job Dashboard Coding

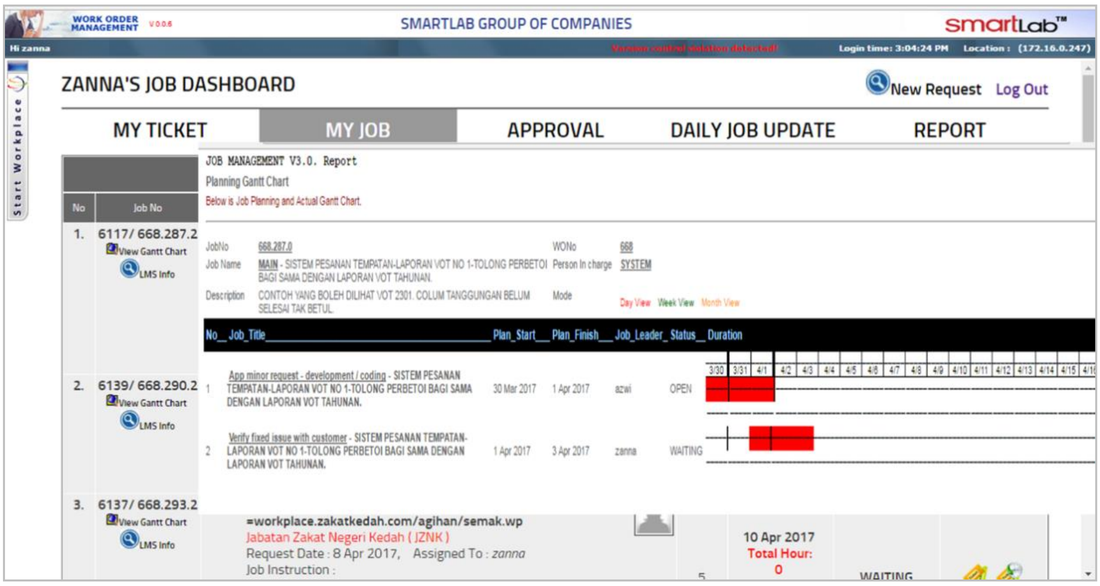



*Figure 5.35* Job Gantt Chart View

Every job created through the developed system had contained its own built-in Gantt chart, as illustrated in Figure 5.35. This feature has provided very good visibility to the other actors within the environment.

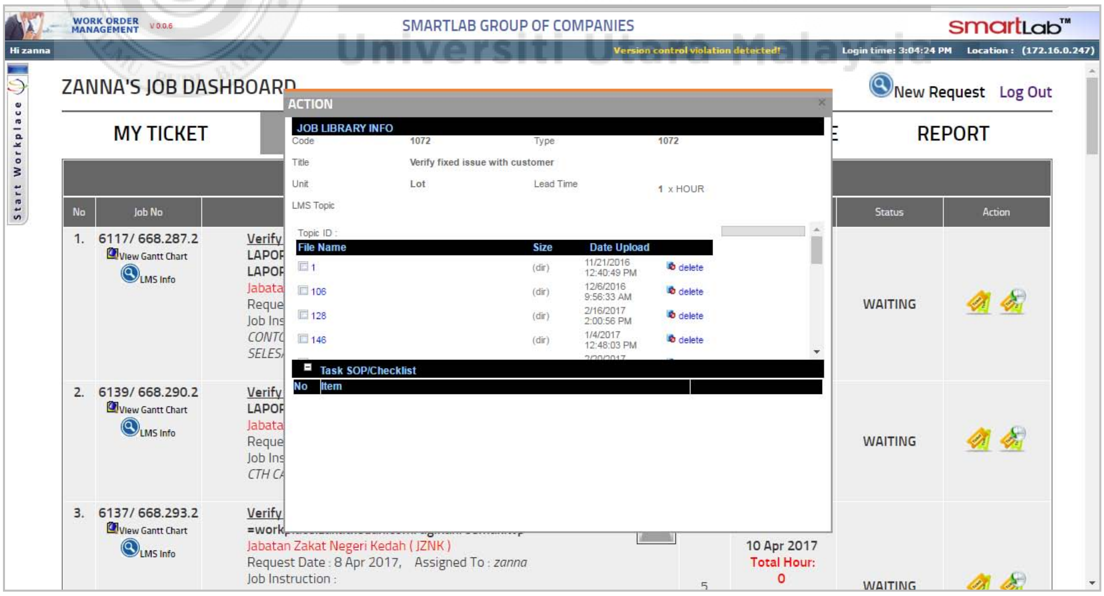



*Figure 5.36* LMS Link from Job Dashboard

The system output for the CEF Workflow was also connected with the Knowledge Management component of the CEF Model. This was demonstrated by the link that provided users with information and data with regards to the product or project in question as shown in Figure 5.36.

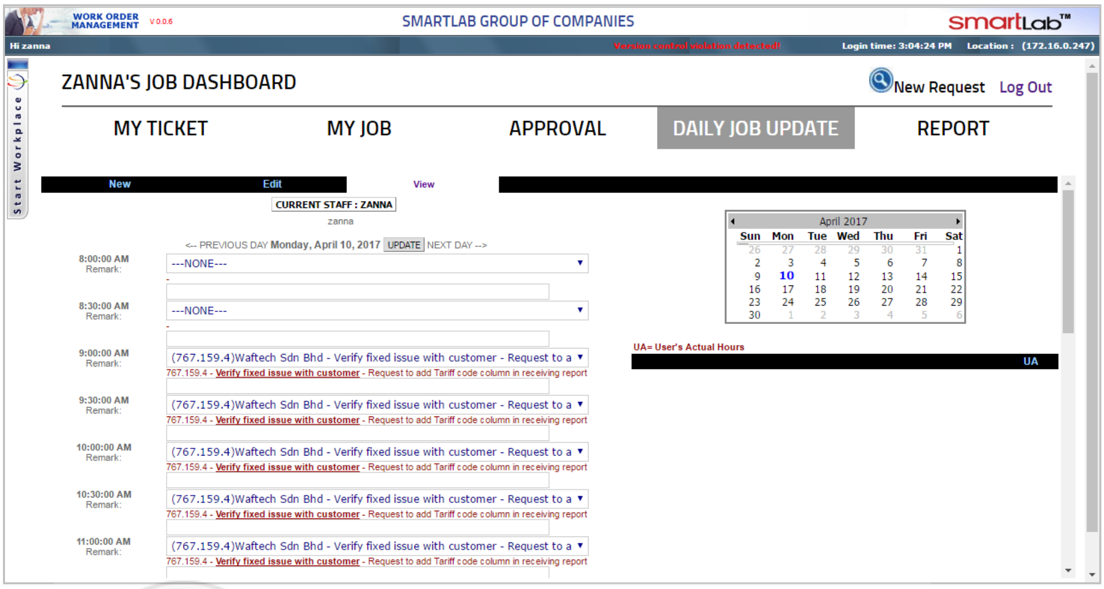



*Figure 5.37* Daily Job Update View

To keep track of the jobs completed and verify the actual man-hour consumption, a Job Update section was developed for the users of the CEF System to update the type of jobs, which they have spent their time on. This is shown in Figure 5.37.



The Performance Monitoring tool and Capacity Planning dashboard were developed as shown in Figures 5.38 and 5.39 and provide the planner in the software development environment with the tool to manage and control which job should be completed. The criteria for the priority level of each job request was determined through a combination of due date requirement and job complexity.

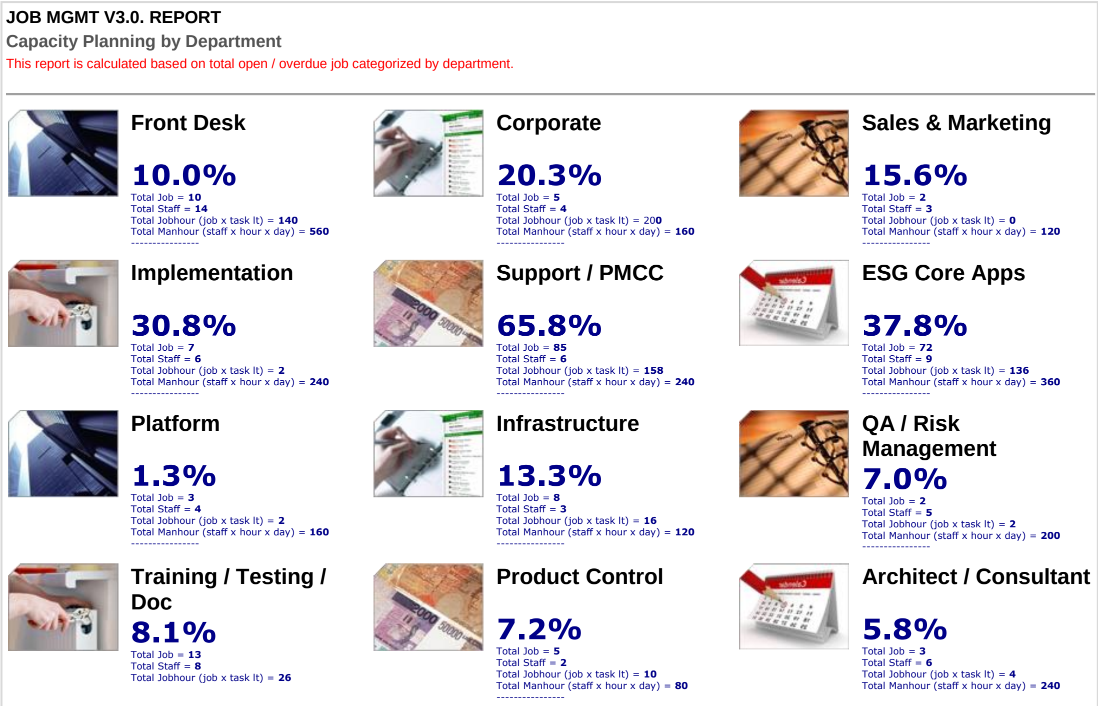



*Figure 5.38* Capacity Planning Dashboard

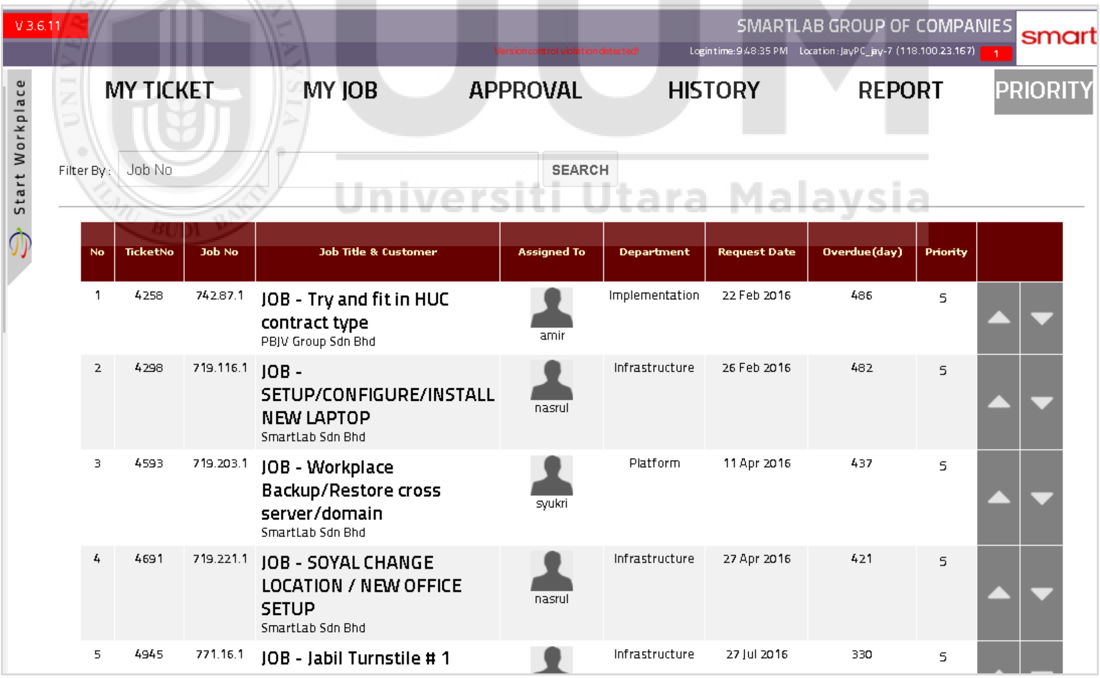





*Figure 5.39* Performance Monitoring with Tuning Capability

### 5.5.4 CEF Product Control

Based on the design guidelines walkthrough result explained in section 5.3, a tool was developed to satisfy the requirement for the simulation of CEF Product Control.

The system was developed to control and manage, providing a mechanism to emulate the processes to control product quality based on the CEF Model. The following subsection briefly the important features of the developed system tool.

#### 5.5.4.1 Product Control System Architecture




*Figure 5.40* Product Control System Architecture Diagram

The Product Management System is the tool developed to satisfy the requirement for the CEF Product Control component. The system consists of several modules, which are Desk View, Quality Control, Release Control, Change Management, and Product Issuing as shown in Figure 5.40.

#### 5.5.4.2 Product Control System Database

The Figure 5.41 is a representation of the system database diagram used to develop the CEF Workflow tool.

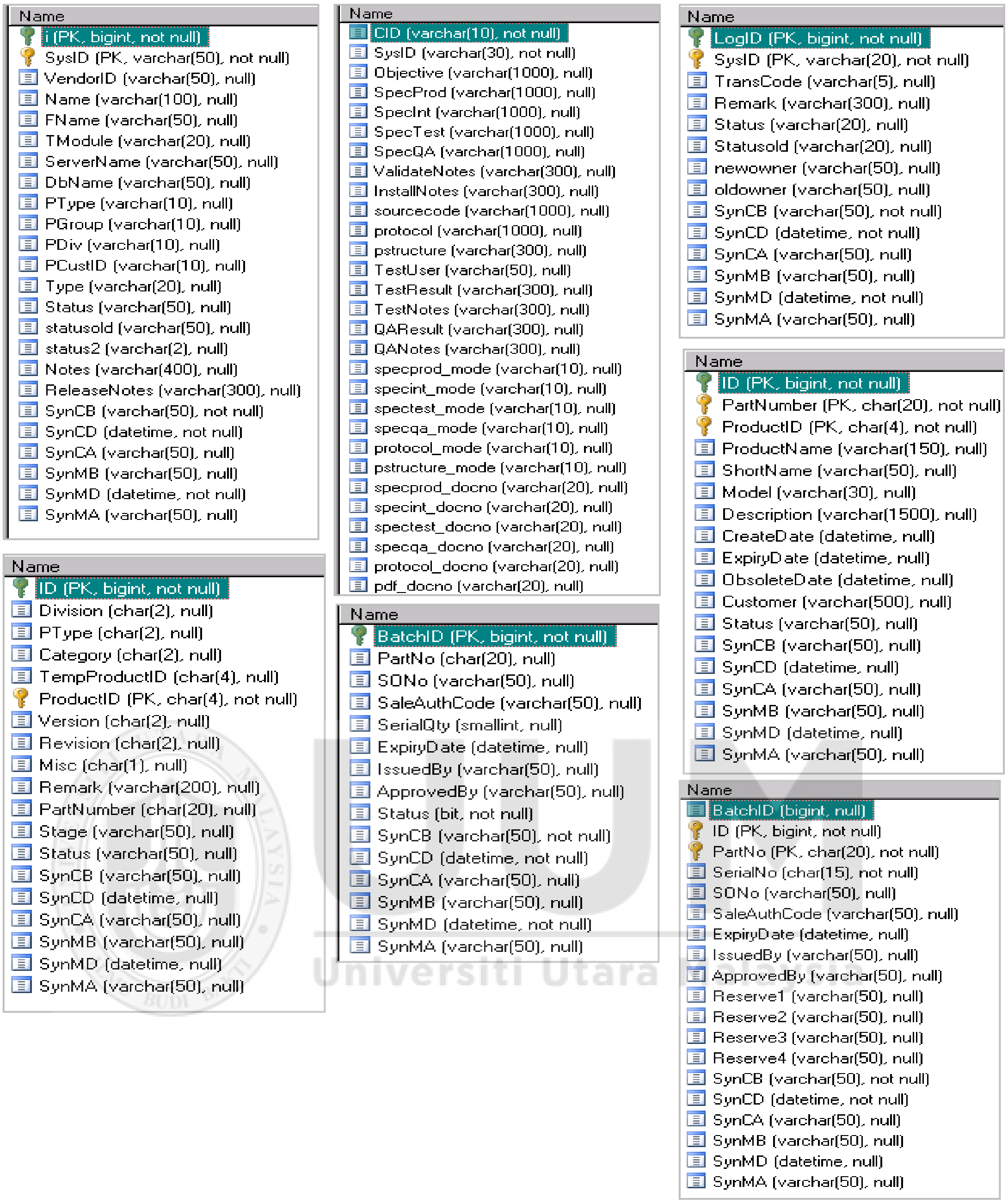



*Figure 5.41* Product Management Database Structure

### 5.5.4.3 Product Control Application Menu

Table 5.16 illustrates the type of menu developed for the CEF Workflow system tool.

Table 5.16

*CEF Product Control Application Menu*

| | | |
|---|---|---|
| **PRODUCT SYSTEM** | **Desk View** | |
| | Search Product | |
| | Test Approval | |
| | QA Approval | |
| | Release | |
| | Change Owner | Initiate Change |
| | | Accept Change |
| | Trans Log | |
| | Product Issuing | Serial Request |
| | | Serial Approval |
| | | Serial Issuing |
| | | Product Activation |



### 5.5.4.4 Product Control System Snapshot

Figures 5.42 and 5.43 display the list of products in a Desk View format. System Identification and Version/Revision element described in the CEF Product Control were represented using this module. Each product listed contains its own attributes, such as product part number, status, product type, and the actions that can be done for the product. In this view, the products listed will be associated to the assigned owners. This means only the owners are able to view contents of the product information and alter its contents.

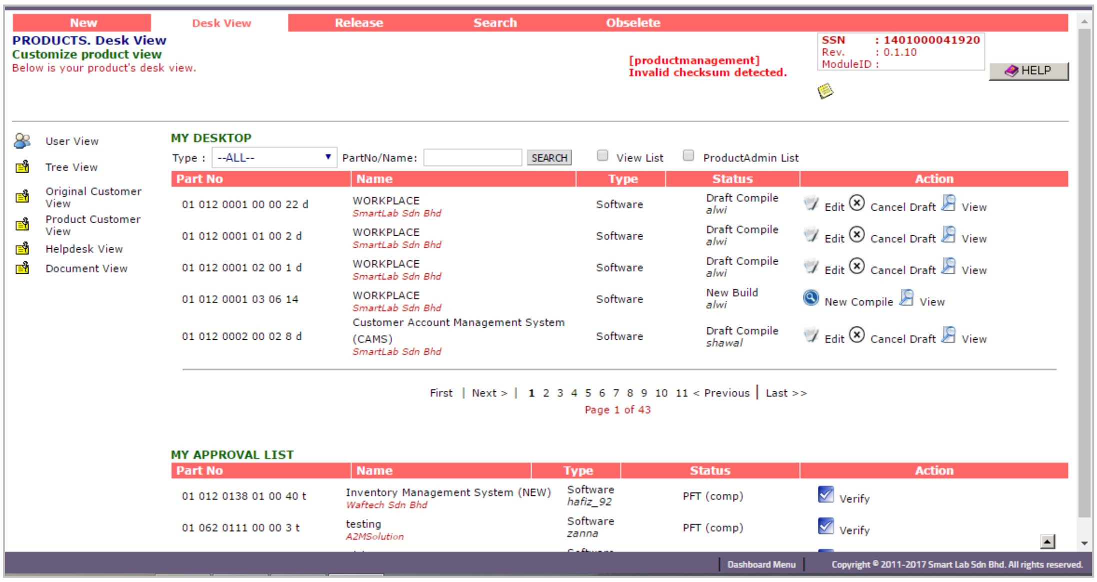



*Figure 5.42* Product Desk View

```
</td>
<td width="21%" align="left" height="4">
<%if mid(rs("status"),5,1)="d" then
        ptxt="Cancel Draft"
        p="CancelChangeOwner"
        picon="cancelnv3.jpg"
        vtarget=""%>
            <a href="product_compile.asp?mode=<%=rs("status")%>&sysid=<%=rs("sysid")%>" target="MyFrameTwo">
        <img border="0" src="images/gedit.jpg" alt="to edit" width="17" height="18" class="style1" >
        </a><font class="applabel">Edit</font>
        <a href="product_process.asp?p=<%=p%>&m=view2&t=<%=rs("status")%>&sysid=<%=rs("sysid")%>" <%=vtarget%>  >
        <%if picon<>"" then%><img border="0" src="images/<%=picon%>" width="17" height="18" class="style1" >
        </a><font class="applabel"><%=ptxt%></font><%end if%>
    <%elseif mid(rs("status"),5,1)="p" then%>
            <a href="release_detail.asp?t=<%=mid(rs("status"),1,4)%>&sysid=<%=rs("sysid")%>" target="MyFrameTwo">
        <img border="0" src="images/gapprovev2.jpg" alt="to release" width="17" height="18" class="style1" >
        </a><font class="applabel">Release</font>
    <%elseif mid(rs("status"),5,1)="t" then%>

    <%else%>
            <a href="product_compile.asp?newoption=compile&sysid=<%=rs("sysid")%>" target="MyFrameTwo">
        <img border="0" src="images/gviewv2.jpg" alt="to compile" width="17" height="18" class="style1" >
        </a><font class="applabel">New Compile</font>
    <%end if%>

<a href="product_viewdetail.asp?m=view4&t=<%=rs("status")%>&sysid=<%=rs("sysid")%>" onclick="return sendThis(this)" >
        <img border="0" src="images/gviewv1.jpg" alt="to preview" width="17" height="18" class="style1" >
        </a><font class="applabel">View</font>
</td>
```

*Figure 5.43* Product Desk View coding

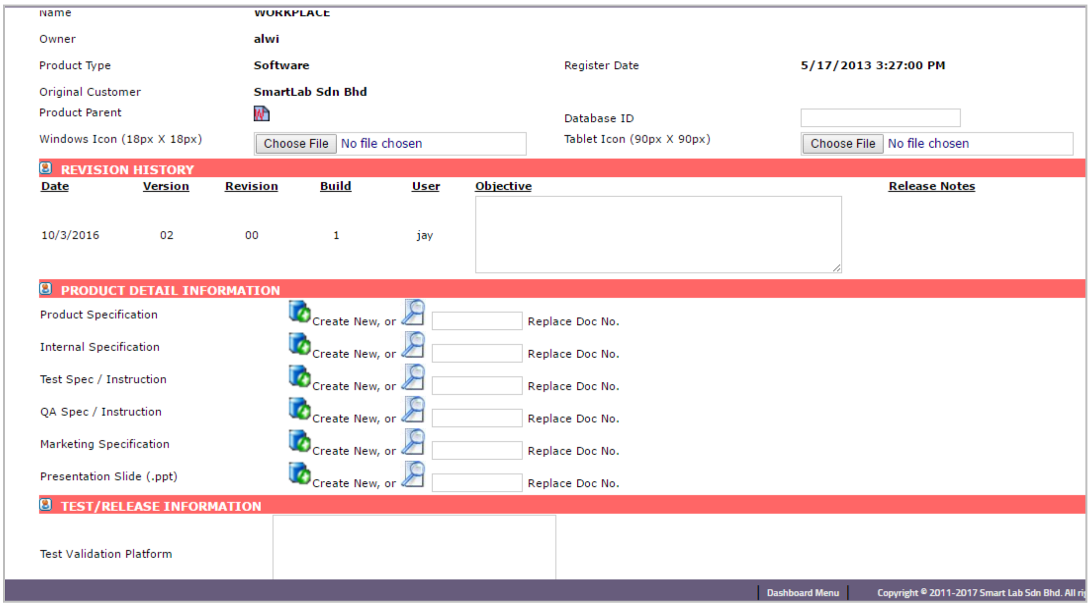



*Figure 5.44* Product Release

Figures 5.44 and 5.45 display the list of the product's detail information during product release process that illustrate the Release Control element described in the CEF Product Control model. Each product needs to have a complete list of documentations, such as Product Specification, Internal Specification, Test Specification, Quality Assurance Specification, Marketing Specification, and Presentation material before being submitted for approval. Once the product has been approved, the product is ready to be released. Figures 5.46 and 5.47 are snapshots of the Product Management system to test and release any developed product.

```
<%if mode = "REVISION" then

    pver = mid(request("sysid"),13,2)
    prev = fname

    response.write "pversion="&pver&",prevision="&prev&"<br>"
    'response.end

    '''''''''''''''''''''''''''''''''auto add to product system'''''''''''''''''
    id = prod_registration(name,name,servername,tmodule,"0"&pdiv,ptype,date(),apptype,pver,prev,"0",partno,ppid,"")

    cmdmm.commandtext = "insert into application (sysid,vendorid,name,fname,tmodule,servername,dbname,ptype,pgroup,pdi
    "('"&id&"','"&replace(vendorid,"'","")&"','"&replace(name,"'","")&"','"&replace(id,"'","")&"','"&replace(tmodule,"
    cmdmm.execute

    cmdmm.commandtext = "insert into applicationlog (sysid,transcode,remark,syncb,synca) values ('"&id&"','x006d','','
    cmdmm.execute

    cmdmm.commandtext = "update application set status='x006d' where sysid='"&id&"'"
    cmdmm.execute

    appfolder = wpuploadpath &"\"& id
    If not myFSO.FolderExists(appfolder) Then
        myFSO.CreateFolder(appfolder)
    end if
    set myFSO = nothing

    objapplication = id

elseif mode = "VERSION" then

    pver = fname '---(request("sysid"),13,2)
    prev = "00"

    '''''''''''''''''''''''''''''''''auto add to product system'''''''''''''''''
    id = prod_registration(name,name,servername,tmodule,"0"&pdiv,ptype,date(),apptype,pver,prev,"0",partno,ppid,"")
```

*Figure 5.45* Product Release Coding

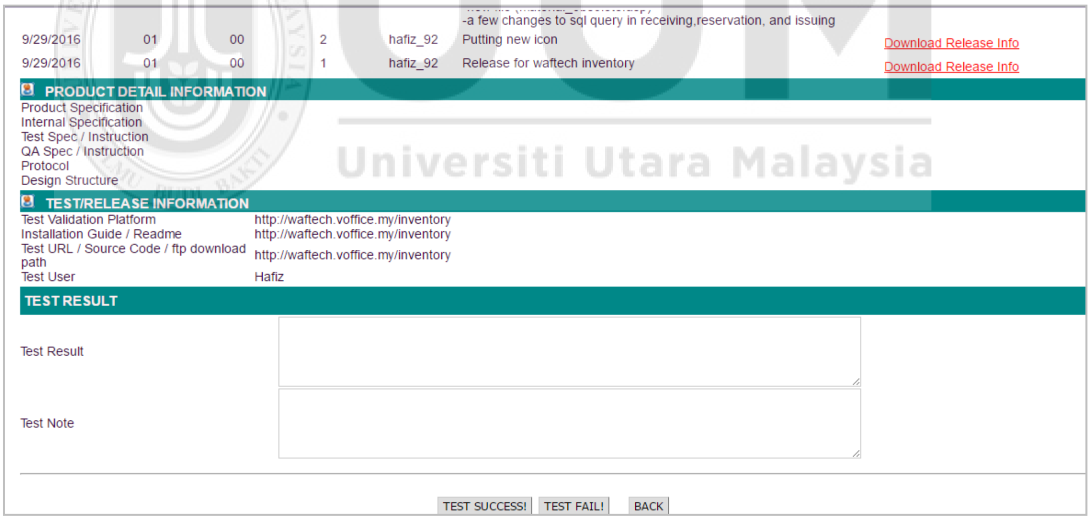



*Figure 5.46* Product Test Approval Page

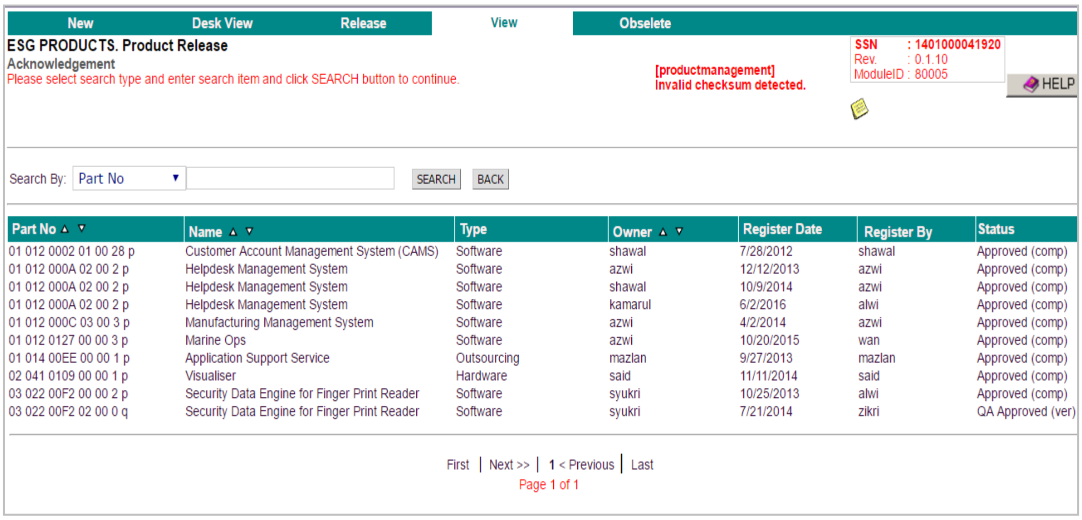



*Figure 5.47* Product Release Listing

Figure 5.48 shows a sample of a serial number issuance after a product release procedure has been completed. The issuance of the serial number illustrated one of the output for the Anti-Software Infringement element proposed in the CEF Product Control component. The serial number is provided once all of its criteria has been fulfilled. The serial number will then be used as a security measure coupled with a specific key to allow the system to be used.



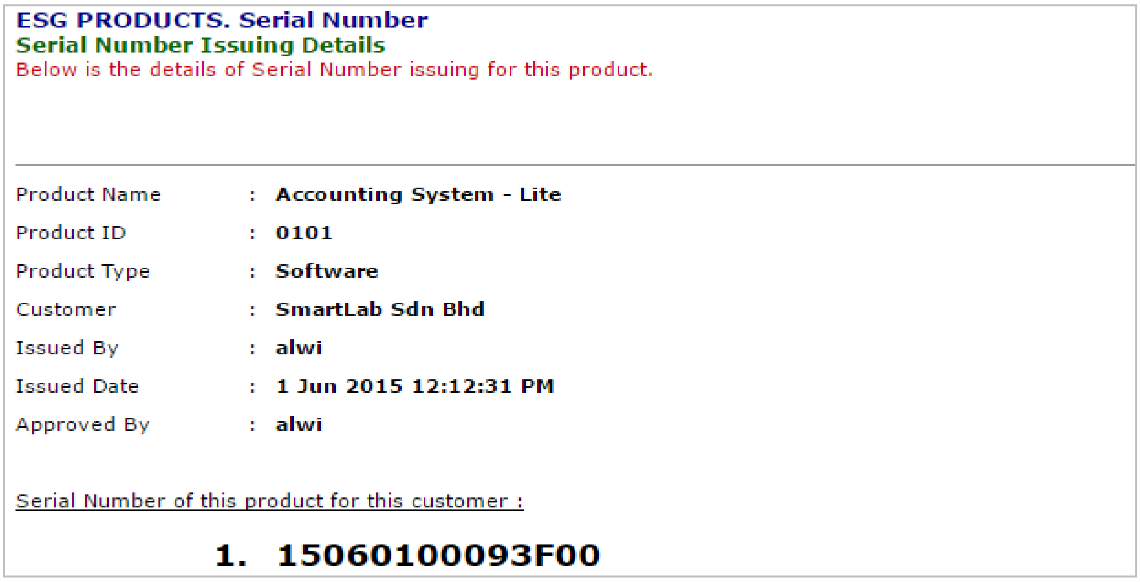



*Figure 5.48* Serial Number Generation


*Figure 5.49* Application Marketplace

The output of the CEF Product Control element was summarized by the development of the Apps Marketplace shown in Figure 5.49. With this, clients were able to log in and download the necessary updates for their subscribed modules in their own time and necessity.



### 5.5.5 Knowledge Management

Using the design guidelines walkthrough result explained in section 5.3, a tool was developed to satisfy the requirement for the simulation of the Knowledge Management component. The system was developed to handle the distribution and logging of knowledge materials based on the CEF Model. The following subsection briefly touches on the important features of the developed system tool.

#### 5.5.5.1 Knowledge Management System Architecture


*Figure 5.50* Knowledge Management System Architecture Diagram

The Knowledge Management System is the tool developed to satisfy the requirement for the Knowledge Management component. The system consists of several modules, which are Course Management, Program, Session, Certification, Classes and Examination, and Learning Materials illustrated in Figure 5.50.

#### 5.5.5.2 Knowledge Management System Database

The Figure 5.51 and 5.52 is a representation of the system database diagram and table listing used to develop the Knowledge Management tool.

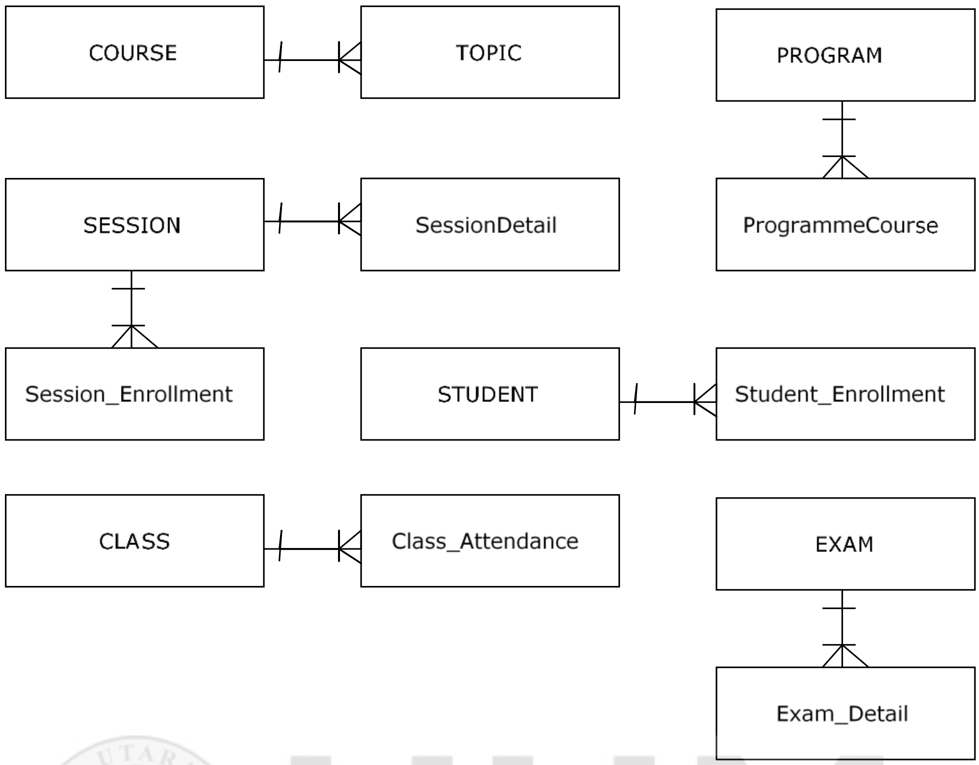



*Figure 5.51* CEF Knowledge Management System Database Diagram

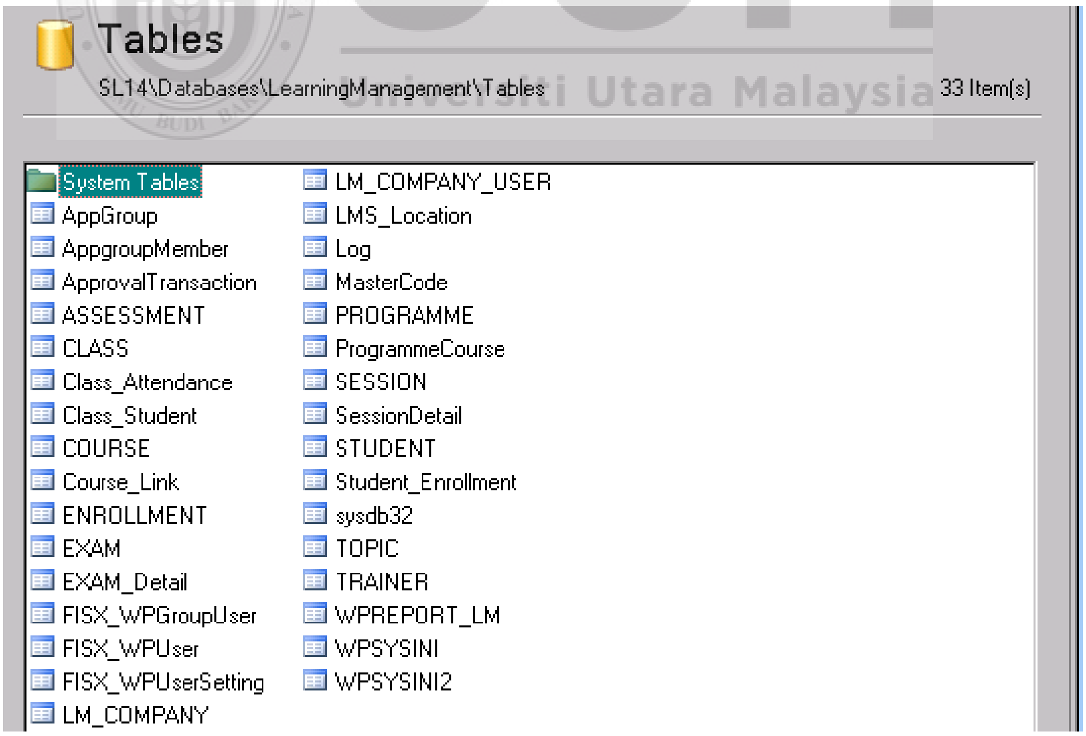



*Figure 5.52* CEF Knowledge Management Table Listing

#### 5.5.5.3 Knowledge Management Application Menu

Table 5.17 illustrates the type of menu developed for the CEF Workflow system tool.

Table 5.17

*Learning Management System Application Menu*

| **LEARNING MANAGEMENT SYSTEM** | **Courses** | **Info** |
|---|---|---|
| | | Topic |
| | | Material |
| | | Assessment |
| | Program | |
| | Session | |
| | Classes | |
| | Enrollment | |
| | Student Dashboard | |
| | Examination | |
| | Certification | |
| | DeskView | |
| | Report | |



#### 5.5.5.4 Knowledge Management System Snapshot

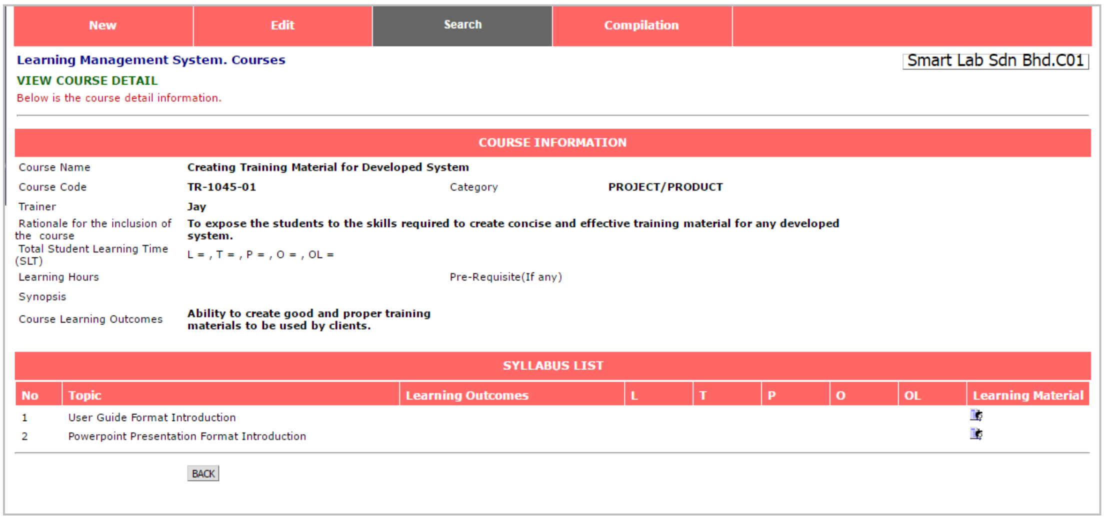



*Figure 5.53* Course Detail

Figures 5.53 and 5.54 display a course detail in the Knowledge Management System that was developed through the Knowledge Identification and Segmentation element.

Each course has its own course information based on the type of course registered by the course creator or trainer. Registering the course also requires a detailed syllabus to be created and completed with the learning material's link.

```
<%
if request("processcourse") <> "" then
    cdocid = request.querystring("CourseCode")
    cmode = "edit"
        cmdmfg.CommandText="update COURSE Set CourseCode='"&Trim(Fields("crscode").Value)&"',CourseName='"&Trim(Fields("cr
    "Trainer='"&Trim(Fields("Trainer").Value)&"',Incourse='"&Trim(Fields("incrs").Value)&"',Lecture='"&Trim(Fields("lectur
    "Synopsis='"&Trim(Fields("Synopsis").Value)&"',CourseLearning='"&Trim(Fields("crslearning").Value)&"',synmB='"&session
    cmdmfg.execute
    xy_msg1 = "Selected course has been successfully updated."
else

    rsmm1.open "select code from mastercode where category='id_course'"
    if not rsmm1.eof then
        curcrs = rsmm1("code")
        newcrs = rsmm1("code")+1
        crscode = Trim(Fields("org").Value)&"-"&curcrs&"-01"

    else
        crscode = Trim(Fields("crscode").Value)

    end if
    rsmm1.close

    ''cmode = "new"
    cmdmfg.commandtext = "insert into COURSE(CourseCode,CourseName,CompID,Category,Trainer,Incourse,Lecture,Tutorial,Pract
    "('"&crscode&"','"&Trim(Fields("crsname").Value)&"','"&appscompid&"','"&Trim(Fields("Category").Value)&"','"&Trim(Fiel
    "'"&Trim(Fields("lecture").Value)&"','"&Trim(Fields("tutorial").Value)&"','"&Trim(Fields("practical").Value)&"','"&Tr
    ''response.write "cmdmfg.commandtext="&cmdmfg.commandtext
    ''response.end
    cmdmfg.execute

    cmdmm.commandtext = "update mastercode set code='"&newcrs&"' where category='id_course'"
    cmdmm.execute

    xy_msg1 = "New course has been successfully registered."
```

*Figure 5.54* Course Detail Coding

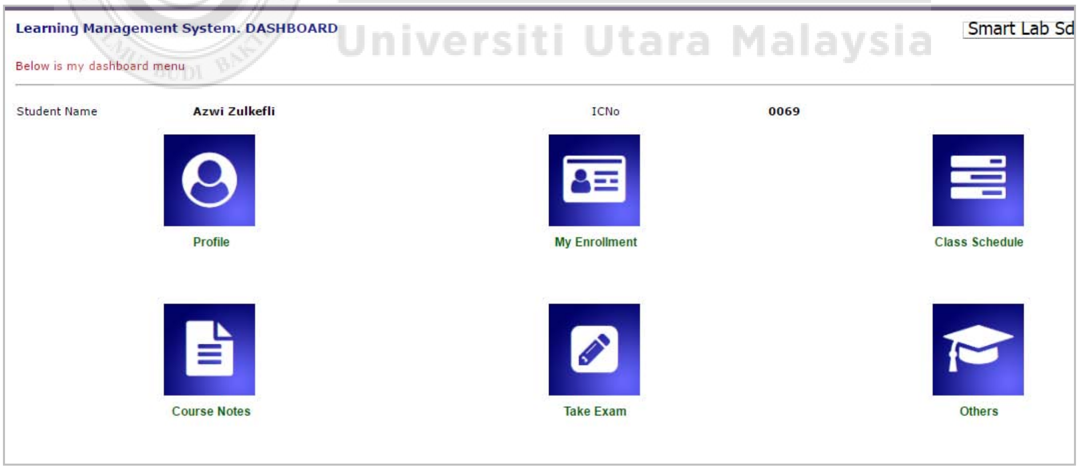



*Figure 5.55* Student Dashboard

Figures 5.55 and 5.56 are the representation of a student/employee dashboard in the Knowledge Management System. This dashboard was developed as the output by combing the Knowledge Retrieval and Dissemination and Knowledge to Task

Mapping element. The dashboard provides the employee with vital information such as Enrolment, Class Schedule, Course Notes, Examination portal, and others. From this dashboard, employees could easily access the learning materials based on the courses that they have registered for.

```
<table border="0" cellspacing="0" style="border-collapse: collapse" bordercolor="#111111" width="100%" height="69">
  <tr>
    <td width="33%" colspan=2 align="center" valign="top" height="21">
   <%=dashIcon("1200002","Profile","stu_profile.asp?stid="&request("stid")&"","images/myhr-myprofile.png")%>
    </td>
     <td width="33%" colspan=2 align="center" valign="top" height="21">
   <%=dashIcon("1200002","My Enrollment","stu_enroll.asp?stid="&request("stid")&"","images/myhr-enroll.png")%>
    </td>
    <td width="33%" colspan=2 align="center" valign="top" height="21">
   <%=dashIcon("1200001","Class Schedule","stu_class.asp?stid="&request("stid")&"","images/myhr-schedule.png")%>
    </td>
      </tr>
   <tr>
    <td width="33%" colspan=6 align="center" valign="top" height="21"> </td>
      </tr>
   <tr>
    <td width="33%" colspan=6 align="center" valign="top" height="21"> </td>
      </tr>
    <tr>
    <td width="33%" colspan=2 align="center" valign="top" height="21">
   <%=dashIcon("1200001","Course Notes","stu_crs.asp?stid="&request("stid")&"","images/myhr-notes.png")%>
    </td>
    <td width="33%" colspan=2 align="center" valign="top" height="21">
   <%=dashIcon("1200001","Take Exam","exm2.asp?src=dash&stid="&request("stid")&"","images/myhr-exam.png")%>
    </td>
     <td width="33%" colspan=2 align="center" valign="top" height="21">
   <%=dashIcon("1200001","Others","#","images/myhr-result.png")%>
    </td>
      </tr>
   <tr>
    <td width="33%" colspan=6 align="center" valign="top" height="21"> </td>
      </tr>
```

*Figure 5.56* Student Dashboard Coding



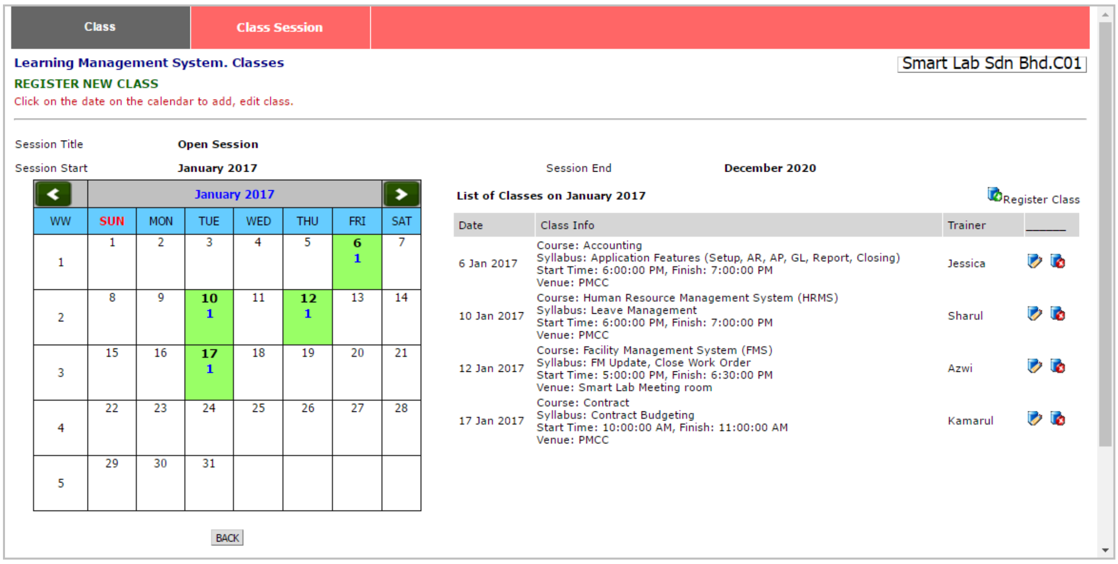



*Figure 5.57* Course Schedule for Registration

Figure 5.57 displays the Course Schedule and List of Classes, which have been entered into the Knowledge Management System that can be related to all of the core model elements in Knowledge Management component. Employees were able to view this Course Schedule and register for the classes offered in the Schedule.

### 5.6 Model Verification

To verify that the system complies with the guidelines and the CEF Model provided, each model to system development have gone through a series of checklists based on the inspection checklist generated after the expert review feedback. The verification was completed by the deployment team as described in section 5.3. The process of model verification was an iterative process in which all the team members eventually agreed that each checklist item was fulfilled as attached in Appendix B. Otherwise, the related items were revisited and refined until it satisfies all of the team members. In Table 5.18, each of the CEF model component is verified through a checklist after the CEF system is developed.



Table 5.18

*CEF System Verification Checklist*

| **Component** | **No** | **Checklist Description** | **Comply (Y/N)** |
|---|---|---|---|
| CEF Product Line | 1 | The product line grouping should cover intended product family and its sub product as well as future product can be fit into of the product family. | Y |
| | 2 | Important and common ERP modules should be defined and grouped as Core Module. | Y |
| | 3 | Model can be deployed to customer individually or in integrated mode (loosely coupled). | Y |
| | 4 | Core system, industry solutions and legacy can be integrated with standard interface protocols. | Y |

Table 5.18 continued

| | | | |
|---|---|---|---|
| CEF Platform | 5 | Future upgrades for one particular product should not affect the other system through backwards compatibility. | Y |
| | 6 | Each product has its own configuration parameter. | Y |
| | 7 | All product should fit in a common platform which has built-in product components. | Y |
| | 8 | All product information is systematically through a system. | Y |
| | 9 | Audit process and enforcement model are available. | Y |
| | 10 | The product line grouping should cover intended product family and its sub product as well as future product can be fit into of the product family. | Y |
| | 11 | It should be able to integrate with existing legacy product through API. | Y |
| | 12 | Model can be deployed to customer individually or in integrated mode (loosely coupled). | Y |
| | 13 | Future upgrades for one particular product should not affect the other system through backwards compatibility. | Y |
| | 14 | Each product has its own configuration parameter. | Y |
| CEF Workflow | 15 | The jobs can be across the department depending on the responsibility the task performed. | Y |
| | 16 | Each department need to have their own department specific job complete with checklists. | Y |
| | 17 | Each job can be used for other department's Work Template. | Y |
| | 18 | Upon confirmation of a Work Order, work order template will be used to distribute the jobs assigned to the designated staff within the system. | Y |
| | 19 | Each task should carry its own time frame to allow for tracking of progress and at which task. | Y |
| CEF Product Control | 20 | Part number generation must be done with the event of new product released. (Concurrent only with a sale of a new product). | Y |
| | 21 | Product database should contain all documentation which includes product specification, release notes, system algorithm and actual coding files. | Y |
| | 22 | Each product development owner is responsible for the release of the developed product. | Y |
| | 23 | Serial number activation is required to allow for keeping tabs on the number of products being sold and/or deployed to clients. | Y |
| CEF Knowledge Management | 24 | The knowledge creator has full power over the content of the knowledge provided. | Y |
| | 25 | The updates of contents, training materials, guides, and related documentation are conducted by the administrator, through the main knowledge management engine. | Y |
| | 26 | Able to ensure knowledge retrieval and logging through procedures in the system. | Y |
| | 27 | The system ensures that all knowledge can be referred to from a single focal point i.e. dashboard. | Y |

## 5.7 CEF Model Deployment Training and Implementation

During the development of the CEF System, a series of CEF Model Deployment Offsite Training sessions were conducted for the selected company's employees. In each of the training session, the employees were exposed to the developed CEF System, which were Job Management System, Product Management System, and Learning Management System. The employees were trained in using each of the systems depending on their responsibility and job function in the software development environment.

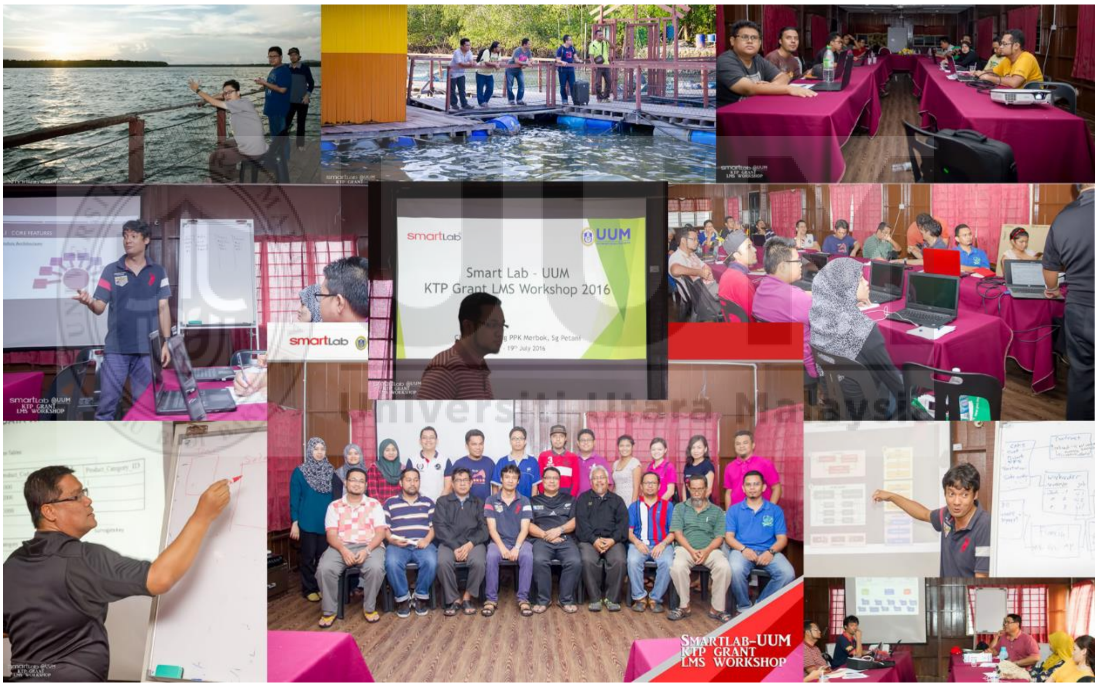



*Figure 5.58* CEF Model Deployment Training at PPK Merbok, Kedah

The first off site training session was held at the Floating Chalet PPK Merbok in Kedah. The pictures in Figure 5.58 show the participants involved from the selected company being trained by the researcher from the university. The training focused on the role of the actors as described in the CEF Model. The three days of training has achieved its objectives in training the key actors as internal trainers. All of them were certified as CEF System trainers so that they were able to train their subordinates.

*Figure 5.59* CEF Model Implementation Review in Langkawi, Kedah

The second off site training session was held at the De Baron Hotel in Langkawi, Kedah as shown in Figure 5.59. The picture shows the review and fine tuning of the CEF System deployment meetings between key actors and the researcher. With the completion of our two off-sites training and model verification stage, the selected company has seen been adapting the CEF model in the Cloud ERP production for more than three months. The deployment has concluded that the model is technically feasible, thus meeting our second specific objective.



### 5.8 Summary

Using the CEF model, the research has developed the CEF prototype environment in the selected organization and verified that the model is technically feasible. The CEF prototype environment implemented has also been verified as described in the previous section. Using the five CEF model components and their guidelines, the model is proven to be technically feasible to be adopted in an existing commercial software production environment.

The environment prototype stages provided different challenges and issues as the environment was somewhat a new change compared to the previous method employed by the software development company.

In the product line implementation, the current existing products developed by the company were in line with the guidelines provided in the CEF model. Each of the product were modular in design and able to be integrated when required. This feature is important as Cloud ERP is not a single application but more of a group of integrated business applications. The platform used by the company demonstrated its ability to be reusable and met the criteria and guidelines provided. The researcher discovered that the first two major component of the CEF model can be easily achieved as the organization is already a Cloud ERP solution provider.

The stages following the CEF Platform however proved to be different as each of the component implementation required a system to be developed. The combination of a job workflow system, product control management system and a knowledge management system were essential to ensure that the CEF Model environment can be implemented successfully. Most of the research period was focused on the development of these systems based on the sheer complexity and scope of the systems required.

The success of this implementation also acknowledges that the model is able to be replicated in other similar organizations to create a CEF environment setup. The CEF Model clarity and understandability were validated by the success of the incorporation of the features into the environment system software as shown in Figure 5.14. However, a longer period of time can be expected in utilizing the CEF environment before it is able to be more efficient and useful. At this stage, it has been demonstrated that the model is able to be deployed in a commercial software production environment. However, in order to achieve the main objective of solving the issues explained in the problem statement, three field usability case studies were conducted and are explained in the next chapter.

# CHAPTER SIX
# MODEL VALIDATION

## 6.1 Introduction

In this stage, the CEF Model was validated using real-world field usability case studies. The developed prototype CEF Model environment is used as a validation vehicle in order to validate the model. Three field usability case studies have been conducted to focus on the usability of the CEF Model implemented in the selected Cloud ERP company. Each case study was focused on the usability aspect of the model and whether it directly tackle the problems mentioned in the problem statement. To achieve this, the survey questionnaire has been separated to test the usability of the developed system as well as the ability of the environment to tackle the research problem components. The results for each subcomponent in the three field usability case studies were compiled and compared as to the impact of the implemented CEF Model environment.



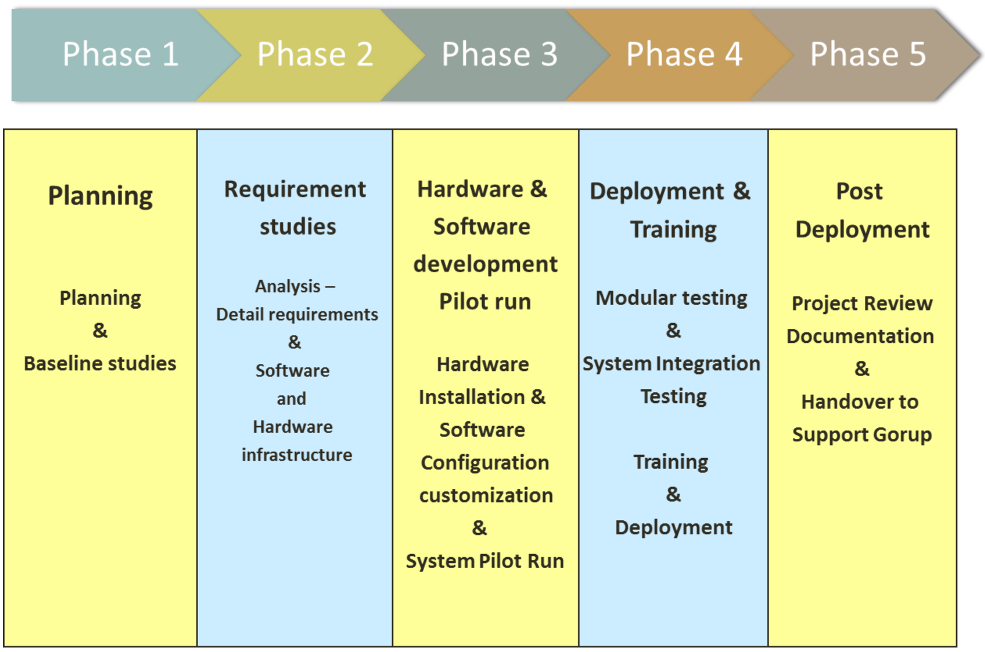



*Figure 6.1* Traditional ERP Implementation Stages

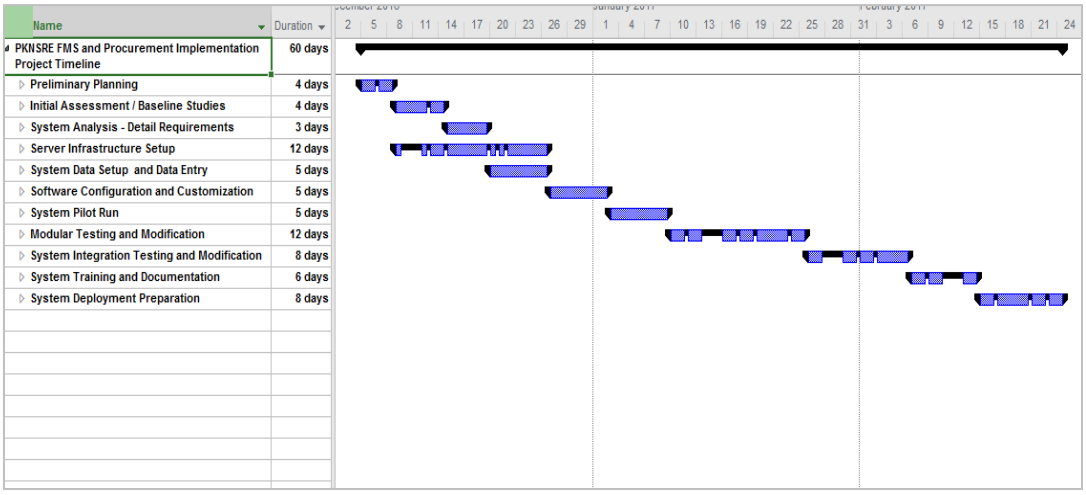



*Figure 6.2* Traditional Cloud ERP Implementation Project Scheduling

In a traditional ERP implementation, most of the planning and execution are done manually. The implementation would normally be initiated with a Work Order, project team mobilization, and project implementation schedule via Gantt Chart. Figure 6.1 and 6.2 both illustrate a typical Cloud ERP Project Scheduling and Implementation stages.



In this research, three case studies were conducted from the perspective of the developers, implementers, clients, and other actors which impacted from the implementation of the CEF model environment conducted in Chapter 5. Although there were three case studies, the actors for each of the case studies were the same to ensure consistency and reliability in obtaining the results. Below were the procedures for each of the case study survey:

i. Interviewed the actors (developers, implementers, etc.) on the perceive usefulness of the CEF environment

ii. Interviewed the actors on the ability of the CEF environment to solve the research problem components.

Based on the case studies conducted, it was verified that through the CEF System implementation, the selected Cloud ERP production company was able to use and benefit from its ability to address the issues mentioned in this research.

## 6.2 Selection of Field Usability Case Studies

To conduct the field usability case study, the criteria to select the type of organization that validates the CEF Model in terms of its usability was identified. The field usability case studies were then picked based on the criteria as per below and summarized by Table 6.1.

i. Scope of the Cloud ERP project

ii. Sizeable implementation

iii. ERP deployment site with users between 100-1000 employees

iv. Cloud ERP with customization

v. Available during the research time frame



Table 6.1

*Field Usability Case Study Criteria*

| Criteria Description | Company A | Company B | Company C |
|---|---|---|---|
| Company Business Nature | Tourism | Realty and Facilities Management | Oil and Gas |
| Size of Company | 200 employees | 400 employees | 250 employees |
| Implemented Modules | CRM, SCM, HRMS, Industry Solution | SCM, FMS, Inventory | CRM, SCM, HRMS |

## 6.3 Field Usability Case Studies Actor Identification

Referring to Table 6.2, twelve Work Centers were identified for the company, which are Platform, Apps Developers, Infrastructure, Quality Control, Support Group, Implementer, Solution Architect, Sales Consultant, Core Apps Developer, Product Control, Training, and Corporate. Each Work Center has its own specific role to play in the CEF Model complete with its own set of services provided.

Table 6.2
*Work Center and Actors Identification Table*

| No. | Name | Role | WCID |
|---|---|---|---|
| 1 | Platform | Maintain and upgrade the platform environment |  |
| 2 | Apps Developer | Maintain, configure, customize, improve |  |
| 3 | Infrastructure | Build, maintain, and manage the Server and network Infra |  |
| 4 | Quality Control | Enforce and audit QA procedure |  |
| 5 | Support Group | Maintain and support post implementation job |  |
| 6 | Implementer | Plan and deploy new implementation order |  |
| 7 | Solution Architect | Business Consultation and Product and Solution Architect |  |
| 8 | Sales Consultant | Manage CRM scope, quotation, and invoicing |  |
| 9 | Core Apps Developer | Maintain and Upgrade standard product, develop new products |  |
| 10 | Product Control | Manage product management policy and server |  |
| 11 | Training | Manage Product training |  |
| 12 | Corporate | Finance, HR, and CEO office role |  |

In a CEF Product Line environment, the Sales Consultant receives requirement from the customer for a new system development. Upon agreeing the design for the new system, the Solution Architect would then instruct the Core Apps developer to develop the system, which is illustrated by Figure 6.3.

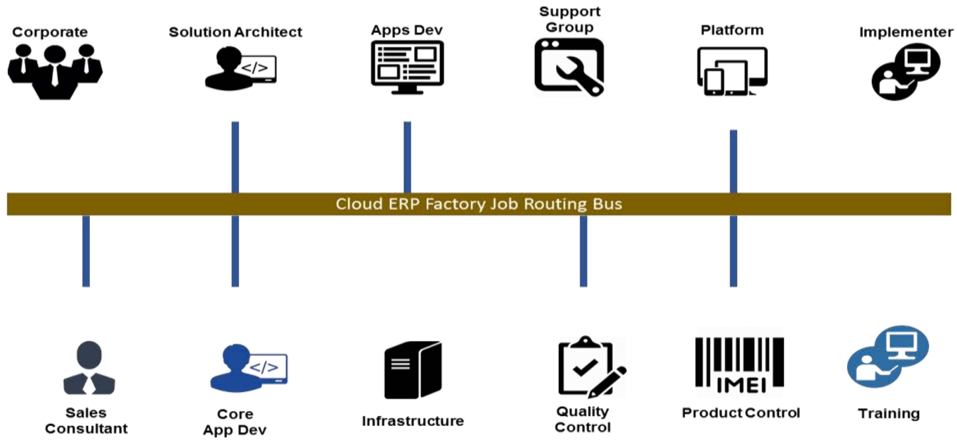



*Figure 6.3* CEF Product Line Actors

Based on Figure 6.4, it is acknowledged by the research that for the Platform Architecture model, both the core apps developers and application developers are affected. This is due to the fact that even before any core module application or customization for a specific application can be done, the platform architecture is referred to.

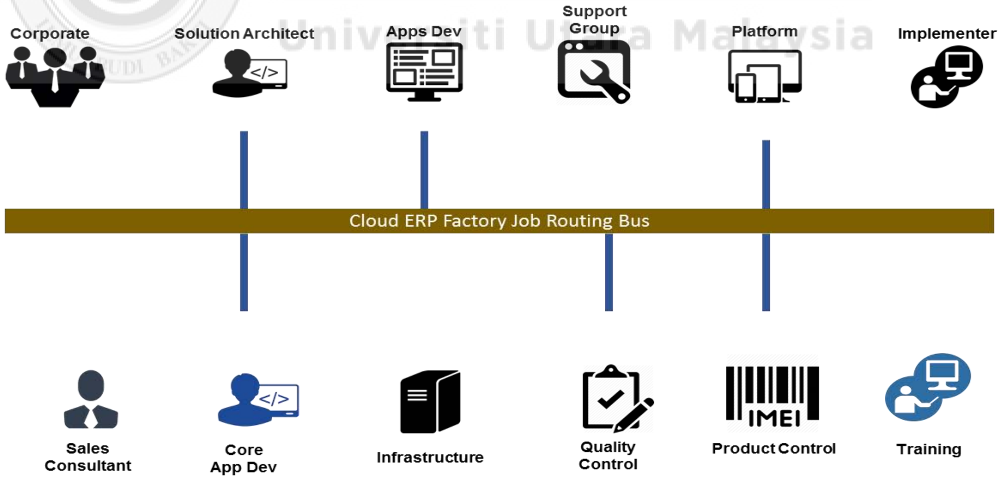



*Figure 6.4* CEF Platform Architecture Actors

As shown in Figure 6.5, the CEF Workflow environment affects all Work Centers in the company. Each Work Center receives jobs from the system with tasks assigned.

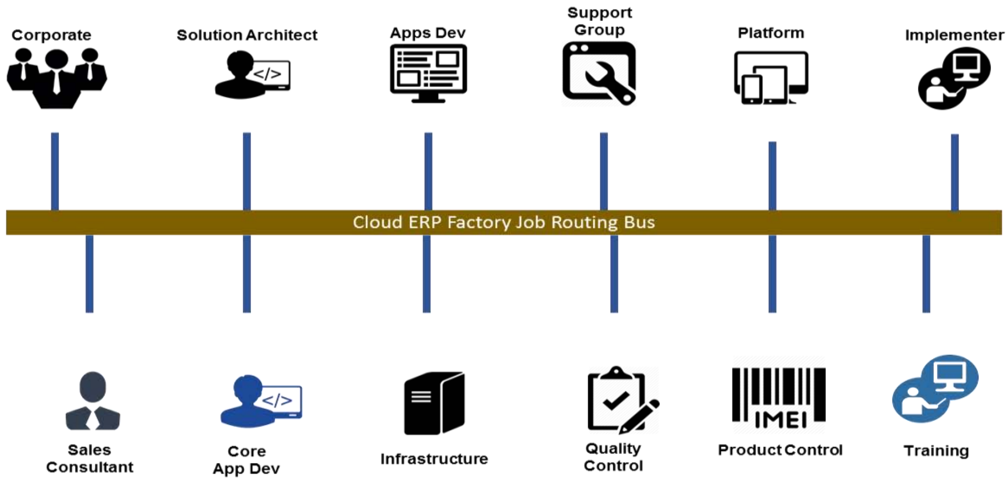



*Figure 6.5* CEF Workflow Actors

Referring to Figure 6.6, Product Control environment involves both the Cloud ERP developer company and the client as well.

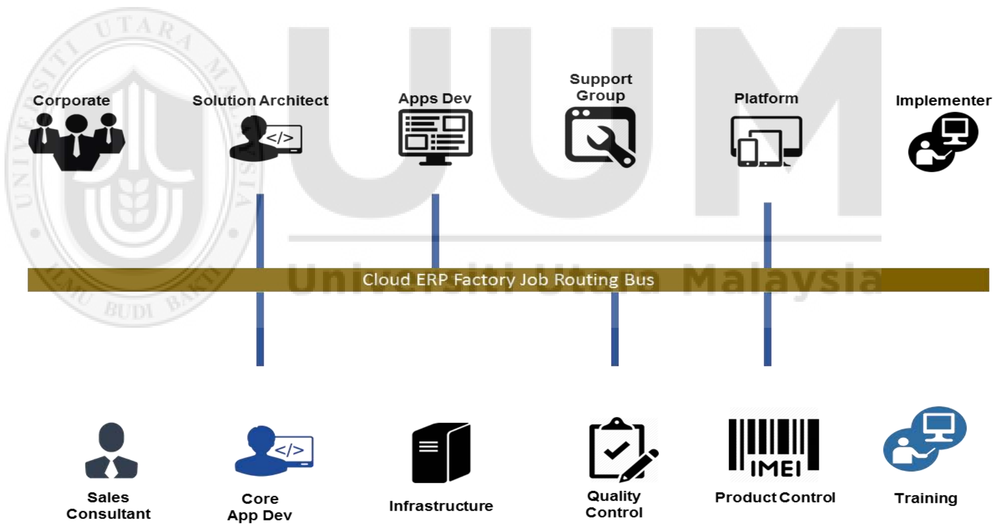



*Figure 6.6* CEF Product Control Actors

The Knowledge Management environment is perhaps the most controlled environment as it does not include any outside parties as illustrated in Figure 6.7.

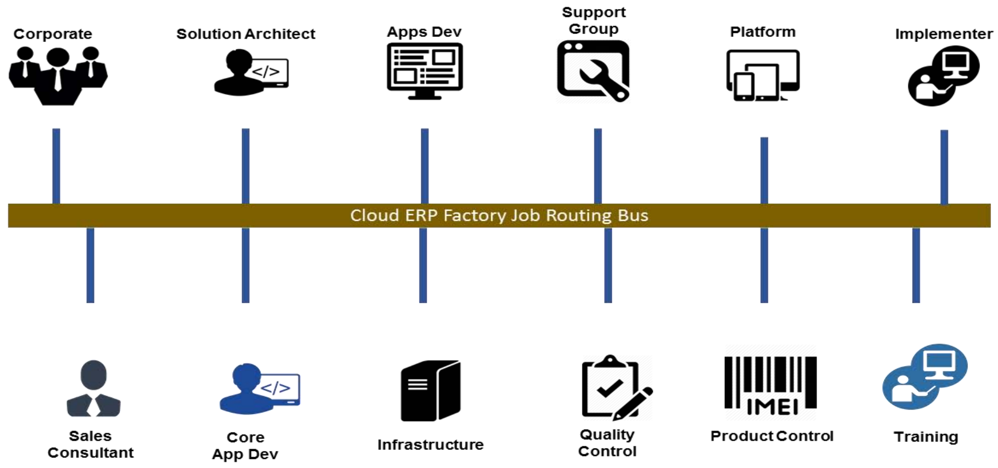



*Figure 6.7* Knowledge Management Actors

## 6.4 Field Usability Instrument

For each field usability case study, below are the PUEU instrument, which is used to gather information from the affected parties with regards to assessing whether the developed system is usable as shown in Table 6.3 and its attached in Appendix A. Additional questions are added as well to allow our field usability evaluation to validate the model's ability to resolve the research problem components. Within 30 minutes average time spent, the following steps were observed for all the respondents during the survey interview feedback; (i) brief respondents on the interview objective and background, (ii) conduct quantitate usability questionnaire interview, and (iii) gather constructive feedback through negative and positive comments.

Table 6.3

*PUEU Validation Instrument*

| No | Perceived Usefulness | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Using the CEF system environment in my job would enable me to accomplish tasks more quickly. | Unlikely | | | | | | | | Likely |
| 2 | Using the CEF system environment would improve my job performance. | Unlikely | | | | | | | | Likely |
| 3 | Using the CEF system environment in my job would increase my productivity. | Unlikely | | | | | | | | Likely |
| 4 | Using the CEF system environment would enhance my effectiveness on the job. | Unlikely | | | | | | | | Likely |
| 5 | Using the CEF system environment would make it easier to do my job. | Unlikely | | | | | | | | Likely |
| 6 | I would find the CEF system environment useful in my job. | Unlikely | | | | | | | | Likely |

Table 6.3 continued

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **Perceived Ease of Use** | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | |
| 7 | Learning to operate the CEF system environment would be easy for me. | Unlikely | | | | | | | | Likely |
| 8 | I would find it easy to get the CEF system environment to do what I want it to do. | Unlikely | | | | | | | | Likely |
| 9 | My interaction with the CEF system environment would be clear and understandable. | Unlikely | | | | | | | | Likely |
| 10 | I would find the CEF system environment to be flexible to interact with. | Unlikely | | | | | | | | Likely |
| 11 | It would be easy for me to become skillful at using the CEF system environment. | Unlikely | | | | | | | | Likely |
| 12 | I would find the CEF system environment easy to use. | Unlikely | | | | | | | | Likely |
| | **Reduce Workload** | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | |
| 1 | The CEF system environment was able to reduce my daily workload compared to previous method. | Strongly Disagree | | | | | | | | Strongly Agree |
| 2 | The CEF system environment was able to improve the overall management of workload in the company compared to the previous method. | Strongly Disagree | | | | | | | | Strongly Agree |
| | **Inconsistent Quality** | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | |
| 3 | The CEF system environment was able to maintain the software quality being developed compared to previous method. | Strongly Disagree | | | | | | | | Strongly Agree |
| 4 | The CEF system environment was able provides better step by step procedures to ensure consistency compared to previous method. | Strongly Disagree | | | | | | | | Strongly Agree |
| | **Reduce Complexity** | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | |
| 5 | Managing Cloud ERP solution is easier compared to the previous method. | Strongly Disagree | | | | | | | | Strongly Agree |
| 6 | The CEF system environment has provided me with easier management tools for my work compared to the previous method. | Strongly Disagree | | | | | | | | Strongly Agree |
| | **Knowledge Retention** | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | |
| 7 | I am able to execute my task with the help of CEF knowledge management system compared to previous method. | Strongly Disagree | | | | | | | | Strongly Agree |

Table 6.3 continued

| | | | |
|---|---|---|---|
| 8 | The CEF system environment was able to provide better knowledge retention compared to previous method. | Strongly Disagree | Strongly Agree |

## 6.5 Field Usability Case Study Respondent Demography

The field usability case studies were conducted by presenting the survey questionnaire to respondents for feedback. The respondents were made up of all the actors directly involved in the ERP implementation for each case study. Table 6.4 shows the list of respondents involved in the field usability case studies.

Table 6.4

*Demographic Background of Respondents*

| **ERP Working Experience (years)** | **Frequency** |
|---|---|
| More than 10 years | 10 |
| Between 5 – 10 years | 7 |
| Between 2 – 5 years | 8 |
| Less than 2 years | 10 |
| Total | 35 |



The majority of the respondents have been in the ERP implementation activity for more than two years and all of them are familiar with the existing traditional ERP implementation practices.

## 6.6 Field Usability Case Study 1

### 6.6.1 Overview

Company A, which was registered on the 28th of September 2002, has effectively from the 6th of January 2003, taken over the operation of the cable car system in Langkawi from the former operator. The cable car operation has received numerous world-class recognitions and awards after it has been managed by Company A such as The Outstanding Tourism Product in Langkawi Tourism Industry Awards 2010/2011, The Malaysia Book of Records for having the Longest Free Span Mono-Cable Car and the First Stand Alone 360°Panorama Screen Dome Theatre as well as the Certificate of Excellence by TripAdvisor in 2014 and 2015. In this field usability case study, the system installed were CRM, SCM, HRMS, and Tourism Industry Solution module, which caters for 200 employees. The CEF exercise was started during the deployment and training phase of the system implementation, which provided a challenge on its own and based on the actor definition, a total of 35 respondents were interviewed based on the survey questionnaires.

### 6.6.2 Case Study 1 Validation Result

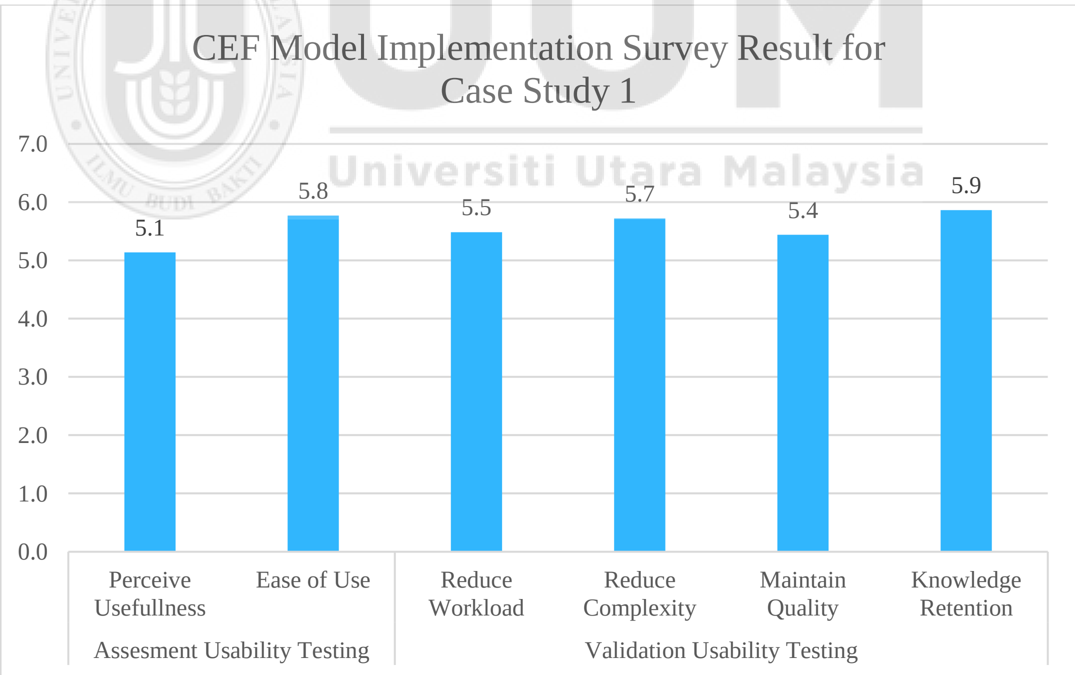



*Figure 6.8* CEF Model Implementation Survey Result for Case Study 1

For the first CEF Model environment implementation result, it was anticipated that the actors involved in the survey would be responsive to the environment's ability to improve the overall process of delivering a Cloud ERP solution to their client as shown in Figure 6.8. On average, most respondents felt that the new CEF model environment provided positive changes as compared to the manual way of deploying Cloud ERP solution. This was proven by the result that shows all of the respondent answered more than 5 on the Likert Scale rating with regards to the model's ability to handle the research problem components, as well as the assessment usability testing for the developed system. Table 6.5 lists the mean value for each of the item measured in the case study.

Table 6.5

*Mean Value for Items Measured in Case Study 1*

| No | Items Measured | Result |
|---|---|---|
| 1 | Using the CEF system environment in my job would enable me to accomplish tasks more quickly. | 5.15 |
| 2 | Using the CEF system environment would improve my job performance. | 5.12 |
| 3 | Using the CEF system environment in my job would increase my productivity. | 5.12 |
| 4 | Using the CEF system environment would enhance my effectiveness on the job. | 5.09 |
| 5 | Using the CEF system environment would make it easier to do my job. | 5.33 |
| 6 | I would find the CEF system environment useful in my job. | 5.00 |
| 7 | Learning to operate the CEF system environment would be easy for me. | 5.91 |
| 8 | I would find it easy to get the CEF system environment to do what I want it to do. | 5.79 |
| 9 | My interaction with the CEF system environment would be clear and understandable. | 5.64 |
| 10 | I would find the CEF system environment to be flexible to interact with. | 5.61 |
| 11 | It would be easy for me to become skilful at using the CEF system environment. | 5.82 |
| 12 | I would find the CEF system environment easy to use. | 5.88 |
| **No** | **Additional Questions** | |
| 1 | The CEF system environment was able to reduce my daily workload compared to previous method. | 5.91 |
| 2 | The CEF system environment was able to improve the overall management of workload in the company compared to the previous method. | 5.76 |
| 3 | The CEF system environment was able to maintain the software quality being developed compared to previous method. | 5.76 |

Table 6.5 continued

| | | |
|---|---|---|
| 4 | The CEF system environment was able provides better step by step procedures to ensure consistency compared to previous method. | 5.76 |
| 5 | Managing Cloud ERP solution is easier compared to the previous method. | 5.76 |
| 6 | The CEF system environment has provided me with easier management tools for my work compared to the previous method. | 5.79 |
| 7 | I am able to execute my task with the help of knowledge management system compared to previous method. | 5.85 |
| 8 | The CEF system environment was able to reduce my daily workload compared to previous method. | 5.91 |

## 6.7 Field Usability Case Study 2

### 6.7.1 Overview

Company B is a real estate company established in 2012 as an investment arm for a state government's group of companies. As a vital unit under the state government, Company B spearheads real estate related investment for the whole Group. The company's diversified real estate portfolio includes investment, asset management, retail management, and advisory. Company B intends to acquire more viable commercial assets in the future. In this field usability case study, the system installed were SCM, Inventory, and Facilities Management Solution (FMS) module, which caters for 400 employees. The implementation of the CEF Model for Case Study 2 was conducted much earlier compared to the first implementation through an actual system deployment. The CEF exercise was stated during Phase 3 of the system implementation, which allowed the CEF production team to be more prepared and able to handle the different requirements through the CEF system, derived by the actor identification phase; thus, a total of 35 respondents were interviewed based on the survey questionnaires.

### 6.7.2 Case Study 2 Validation Result

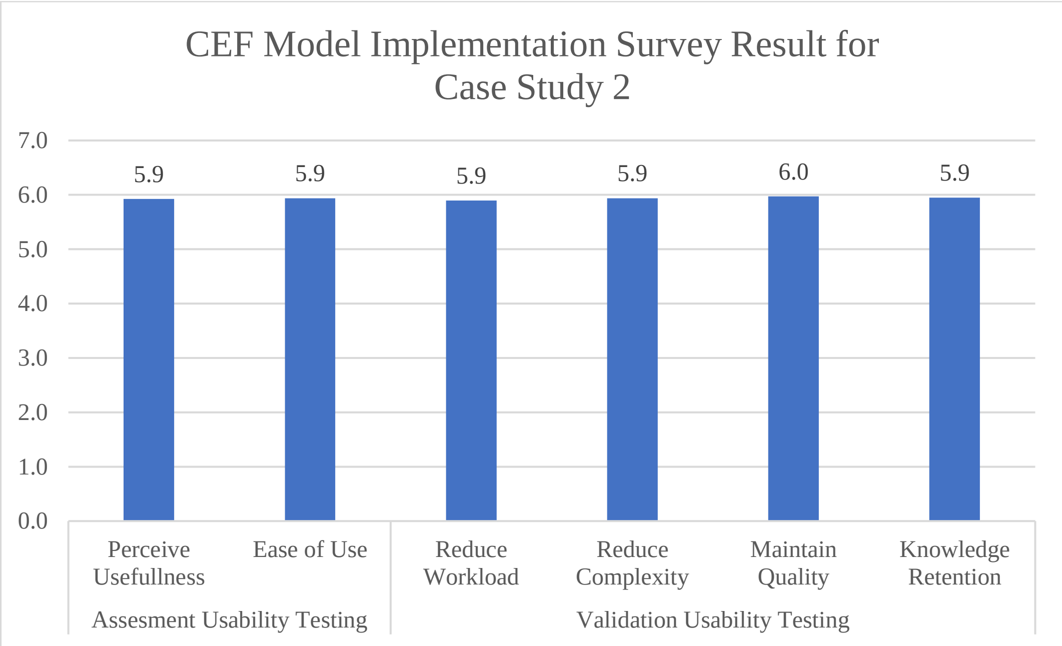



*Figure 6.9* CEF Model Implementation Survey Result for Case Study 2

For the second CEF Model environment implementation result, having used the CEF in the previous implementation, the respondents have a more positive attitude towards using the CEF System. Most respondents felt that in the second time around, they were able complete their job more efficiently as the CEF system provided with a more process dimensioned perspective as shown in Figure 6.9. Both the assessment and validation usability testing result indicated an increase for each of items measured with an average of 5.9 in the Likert Scale rating. Table 6.6 lists the mean value for each of the item measured in the case study.

Table 6.6
*Mean Value for Items Measured in Case Study 2*

| No | Items Measured | Result |
|---|---|---|
| 1 | Using the CEF system environment in my job would enable me to accomplish tasks more quickly. | 5.88 |
| 2 | Using the CEF system environment would improve my job performance. | 5.85 |

Table 6.6 continued

| | | |
|---|---|---|
| 3 | Using the CEF system environment in my job would increase my productivity. | 5.97 |
| 4 | Using the CEF system environment would enhance my effectiveness on the job. | 6.06 |
| 5 | Using the CEF system environment would make it easier to do my job. | 5.91 |
| 6 | I would find the CEF system environment useful in my job. | 5.88 |
| 7 | Learning to operate the CEF system environment would be easy for me. | 6.00 |
| 8 | I would find it easy to get the CEF system environment to do what I want it to do. | 5.97 |
| 9 | My interaction with the CEF system environment would be clear and understandable. | 5.91 |
| 10 | I would find the CEF system environment to be flexible to interact with. | 5.88 |
| 11 | It would be easy for me to become skilful at using the CEF system environment. | 5.97 |
| 12 | I would find the CEF system environment easy to use. | 5.91 |
| **No** | **Additional Questions** | |
| 1 | The CEF system environment was able to reduce my daily workload compared to previous method. | 5.91 |
| 2 | The CEF system environment was able to improve the overall management of workload in the company compared to the previous method. | 5.82 |
| 3 | The CEF system environment was able to maintain the software quality being developed compared to previous method. | 5.94 |
| 4 | The CEF system environment was able provides better step by step procedures to ensure consistency compared to previous method. | 5.94 |
| 5 | Managing Cloud ERP solution is easier compared to the previous method. | 6.00 |
| 6 | The CEF system environment has provided me with easier management tools for my work compared to the previous method. | 5.97 |
| 7 | I am able to execute my task with the help of knowledge management system compared to previous method. | 5.88 |
| 8 | The CEF system environment was able to reduce my daily workload compared to previous method. | 5.91 |

### 6.8 Field Usability Case Study 3

#### 6.8.1 Overview

From a humble beginning as pipeline services provider in 2000, Company C and its subsidiaries have grown to be a main turnkey contractor to the oil and gas industry. It basically provides 4 different type of business; those are pipeline services, offshore logistics, onshore pipelines, construction, and finally major maintenance and hookup commissioning. In this field usability case study, the system installed were CRM, SCM, and HRMS modules, which caters for 250 employees. The implementation of the CEF Model for Case Study 3 was conducted from the beginning of an actual system deployment. The CEF exercise was initiated during the initial phase of the system implementation, which allowed the CEF production team to really go through the full deployment cycle using the CEF System and using the actor definition; therefore, a total of 35 respondents were interviewed based on the survey questionnaires.



#### 6.8.2 Case Study 3 Validation Results

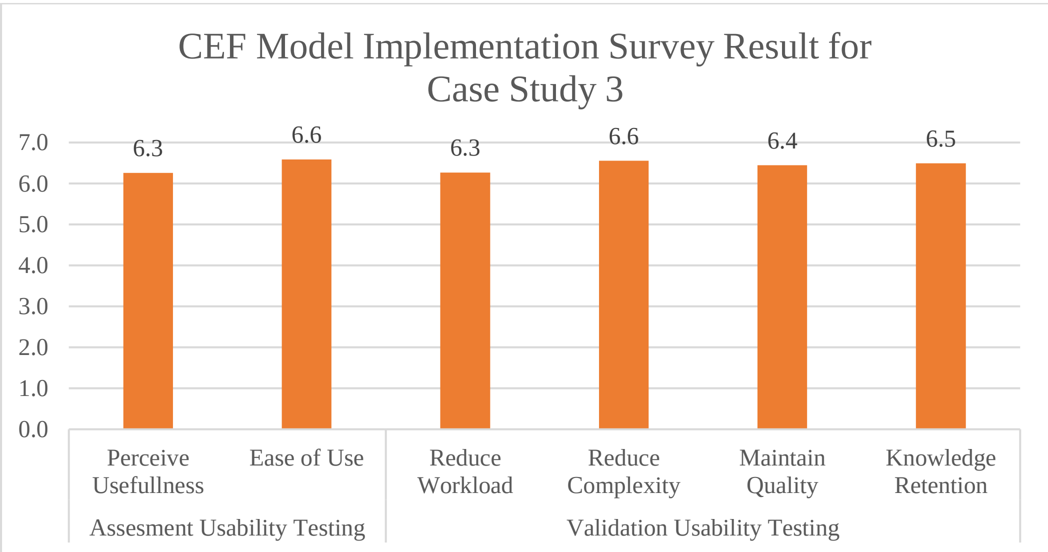



*Figure 6.10* CEF Model Implementation Survey Result for Case Study 3

For the third CEF Model environment implementation result, all of the respondents involved used the CEF System from the beginning of a Cloud ERP deployment. As all of the respondents in the software production company have done this before, the acceptance of completing their jobs through the system was evident with the significant increase of the result for each of the issues asked in the survey questionnaires. As can see from Figure 6.10, each of the issues asked in the survey has increased to a favorable result, for both the usability assessment and the validation usability testing, with an average score of 6.5 on the Likert scale rating. Table 6.7 lists the mean value for each of the item measured in the case study.

Table 6.7

*Mean Value for Items Measured in Case Study 3*

| No | Items Measured | Result |
|---|---|---|
| 1 | Using the CEF system environment in my job would enable me to accomplish tasks more quickly. | 6.18 |
| 2 | Using the CEF system environment would improve my job performance. | 6.21 |
| 3 | Using the CEF system environment in my job would increase my productivity. | 6.33 |
| 4 | Using the CEF system environment would enhance my effectiveness on the job. | 6.30 |
| 5 | Using the CEF system environment would make it easier to do my job. | 6.21 |
| 6 | I would find the CEF system environment useful in my job. | 6.30 |
| 7 | Learning to operate the CEF system environment would be easy for me. | 6.55 |
| 8 | I would find it easy to get the CEF system environment to do what I want it to do. | 6.58 |
| 9 | My interaction with the CEF system environment would be clear and understandable. | 6.64 |
| 10 | I would find the CEF system environment to be flexible to interact with. | 6.61 |
| 11 | It would be easy for me to become skilful at using the CEF system environment. | 6.61 |
| 12 | I would find the CEF system environment easy to use. | 6.55 |
| **No** | **Additional Questions** | |
| 1 | The CEF system environment was able to reduce my daily workload compared to previous method. | 6.06 |
| 2 | The CEF system environment was able to improve the overall management of workload in the company compared to the previous method. | 6.48 |

Table 6.7 continued

| | | |
|---|---|---|
| 3 | The CEF system environment was able to maintain the software quality being developed compared to previous method. | 6.67 |
| 4 | The CEF system environment was able provides better step by step procedures to ensure consistency compared to previous method. | 6.61 |
| 5 | Managing Cloud ERP solution is easier compared to the previous method. | 6.64 |
| 6 | The CEF system environment has provided me with easier management tools for my work compared to the previous method. | 6.64 |
| 7 | I am able to execute my task with the help of knowledge management system compared to previous method. | 6.64 |
| 8 | The CEF system environment was able to provide better knowledge retention compared to previous method. | 6.24 |

## 6.9 Field Usability Findings

In the next section, the field usability results were analyzed and tabulated in each of the case studies conducted.



### 6.9.1 Overall Results for Each Objective

The model's ability to address five areas of concern was analyzed, those areas are Usability, Workload, Quality, Complexity, and Knowledge Management.

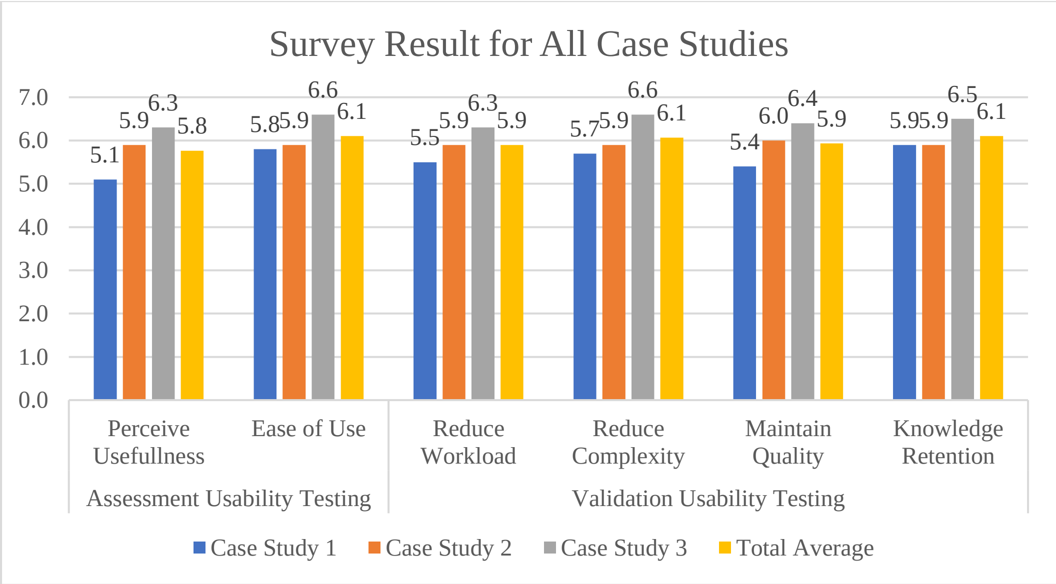



*Figure 6.11* Case Studies Overall Results by Issues

The overall response from the respondent based on the issues faced by Cloud ERP providers has shown an increase with every CEF Model implementation as shown in Figure 6.11. This was also further proven by the average mean of >5 on each of the survey form answered for each case studies described in the validation result section. It was discovered that most respondents were very excited with the Knowledge Management element, which in their view could really provide them with the tools and training needed to complete their tasks. This has solidified our belief that with CEF Model, a Cloud ERP production environment is able to manage and control its Cloud ERP deployment and implementation to its customer.



Table 6.8

*Compiled Negative Qualitative Feedback*

| No | Feedback | Issues Affected |
|---|---|---|
| 1 | Need mobile interface for mobility purposes. | Usability |
| 2 | Block of 30 minutes for job execution – need break between job completion. | Workload |
| 3 | New programmer need few month training to get acclimated. | Quality |

Table 6.8 continued

| | | |
|---|---|---|
| 4 | Reduce overall complexity but require more documentation. | Complexity |
| 5 | Still a lot more knowledge data missing from the system. | Knowledge Management |

In the survey form for each of the case studies, the actors were also requested to provide negative and positive feedback on the overall CEF Model environment implementation. All of the data was sorted out from all of the survey responses and compiled the constructive feedback provided as shown in Tables 6.8 and 6.9.

Table 6.9

*Compiled Positive Qualitative Feedback*

| No | Feedback | Issues Affected |
|---|---|---|
| 1 | User-friendly and better method of managing job. | Usability |
| 2 | Relatively reduce workload for the same type of job. | Workload |
| 3 | More consistent quality as all programmers are involved and not dependent on dedicated staff. | Quality |
| 4 | Simpler in focusing prioritization of job. | Complexity |
| 5 | Good method to allow new hires to keep up to date with any project and product support. | Knowledge Management |

The negative comments provided were mostly with the additional features and suggestions that the respondents felt would improve the CEF model's implementation, such as providing mobile interface for mobility purposes and more training to increase the users' ability in managing and using the CEF environment. Note was taken regarding all of the feedback and will be addressing these suggestions in the future. The positive comments were mostly with regards to the ability of the CEF environment to help with the current challenges faced in the Cloud ERP production environment.

## 6.10 Conclusion

Both quantitative and qualitative survey results have shown that the CEF model is usable and able to tackle the research problem components positively. The field usability case studies have proven that a more structured and process oriented procedure is a very productive way in managing a Cloud ERP production environment. Although the research did not include the correlation factor as a study component, it was observed that both quantitative and qualitative result of each case study has a positive correlation and there was positive trend in the result with the successive case studies.

Due to the different stages of implementation of the CEF environment, the reaction of the actors was very much as expected. As this was their first time using the CEF environment in implementing and supporting a Cloud ERP solution to a customer, most actors felt that more training is required to fully utilize the provided environment. However, the actors realized the significance of the CEF environment in providing better and quicker response time to the client's request and management of the Cloud ERP solution.

Continuing from that momentum, the second case study survey showed a much better improvement and response by the actors. Having gone through the CEF environment the first time, the actors were more in tuned and familiarized with the overall system which in turn allowed them to use it to their advantage. The last case study was expected to be more or less better than the second as most of the actors were very familiar with the systems in the CEF environment. The overall response from the actors has proven this fact.

The overall result based on the issues and the CEF component is a solid representation of the increase of positive reaction towards the use of the CEF system in a Cloud ERP production environment. Apart from just the direct extraction of the results, it was also discovered that the successive results each of the overall CEF model implementations have kept improving with every new case study. This is proven with the positive feedback provided by the respondents with every new implementation of the CEF environment.

# CHAPTER SEVEN
# CONCLUSION

## 7.1 Introduction

This chapter summarizes all the research activities that has been carried out and further review the outcome based on the objectives and the research questions. The Research Findings and Discussion section revisits the three research questions and the research objectives with in-depth reviews. The chapter then elaborates the research contribution artifacts and discuss the research limitation and recommendation before finally concluding the research with a closing summary.

## 7.2 Research Findings and Discussion

This research is based on the research problem encountered by Cloud ERP developers, those are high complexity, high workload, inconsistent software quality, and knowledge retention issue. Literature review reveals that most research were tackling the issue on a specific subject matter and lacking a solution that can produce a model that can resolve all the research problem components. Software Factory was identified as the suitable approach in developing the model and theories such as Mass Customization, Resource Based Theory, Work System Theory, and Sustainability Theory were used to formulate the intended model. The research was then carried out based on these three research questions.

i. What are the components effected and how can the Cloud ERP Factory model be constructed?
ii. How to verify that the Cloud ERP Factory model is technically feasible to be deployed into a Cloud ERP production environment?

iii. Is the proposed model usable in solving the research problem components?

In tandem with seeking the answers for the research questions posed, this study aimed to meet its main objective, which was to propose a Cloud ERP Factory (CEF) Model that will help ERP solution providers to automate and simplify the complexity of software production and management. The research formulated the following specific research objectives in order to satisfy its primary objective.

i. To develop the model by identifying the components with its elements and compile them into the CEF Model.
ii. To verify the model's technical feasibility by deploying it into an existing Cloud ERP Production environment.
iii. To validate the model field usability by using a prototype environment in a real Cloud ERP production case studies.



The specific objectives one, two and three has been achieved and elaborated in chapter four, five and six respectively. With the achievement in meeting all the three specific objectives, it has been demonstrated that the model is usable in the real Cloud ERP production environment to address the challenges described in the problem statement, therefore accomplishing the research primary objective. The following subsections further discuss the outcome of the research based on the research objectives.

### 7.2.1 The CEF Model and its component.

In answering the first research question as well as to meet the first specific objective, a thorough comparative analysis of previous literature and relationship mapping of concept and theories, the model components were identified. The components were then adapted while constructing the CEF Model. The research has demonstrated the ability to apply numerous concepts and theories such as Software Factory, Resource Based Theory and Work System Theory, among others. The researcher utilized Resource-Based Theory, which handles segmentation of processes and skills, to be incorporated in the CEF Model. In a project based approach, these high value resources could only be utilized in a single project implementation at a time. The CEF Model however allows the environment to utilize the best resources such as programmers for all projects. Without this model, those best resources could only be deployed to one project at a particular duration. This research enabled Cloud ERP Production model to adapt mass customization. Instead of avoiding customization, with the CEF Model, Cloud ERP providers are now able to promote dynamic Cloud ERP customization. Likewise, Work System Theory was applied to the Workflow Model, which provided a standard set of operating procedures for any given task related to the Cloud ERP Implementation. After which, Sustainability Theory was presented in the form of the Learning Management System, which allows the software production entity's body of knowledge to be transferred directly to new hires without dependence being placed on any one specific person. Furthermore, Quality Assurance was achieved and bolstered by utilizing key aspects of Sustainability Theory. Software Product Line Theory and Dynamic Software Product Line Theory were both demonstrated in the first component of the CEF Model, that of Product Line. Additionally, Software Configuration Management Theory was adapted and demonstrated through the method and approach of the model regarding product versions and their releases. Similarly, Knowledge Management Theory was used in conjunction with that of Sustainability Theory in order to develop and fine-tune the Knowledge Management System. Continuous Quality Improvement Theory (CQI) was adapted and the CEF Model was found to be able to support CQI, whereby business processes can continuously be improved and streamlined. The CEF Model allowed the current research to achieve this objective in that the CEF Model allows for change. The model development process underwent the entire model-generation cycle, from the creation of the CEF Model itself, verification of the model's usability in the system environment, to the real-world test of the CEF Model; the results of this model development endeavor prove that the model is usable in a real Cloud ERP Production Environment, thus one of the specific objectives of the current research was accomplished.

The Cloud ERP production model is composed of five separate components, which are CEF Product Line, CEF Platform Architecture, CEF Workflow, CEF Product Control, and Knowledge Management. The components for the CEF Model were identified through a series of comparative analyses on manufacturing systems as well as several concepts and theories, such as Software Factory, Work System Theory, and Software Product Line Engineering to name a few. The CEF Product Line, which was the first component of the CEF Model was based on the combination of Software Product Line and the concepts taken from a manufacturing system. This component was concerned with the product identification of a Cloud ERP production and the type of Cloud ERP products which were deployed over the Cloud. The second component of the CEF Model was identified as CEF Platform, which was closely related to the CEF Product Line as both components rely on each other. In this component, Platform was identified as separable, in terms of User Portal, a developer's platform, and redundancy activities. The CEF Product Line and Platform were both essential in identifying the type of products and platform to support Cloud

ERP deployment. The third component, however, was the most crucial, as it provided the engine to manage the overall CEF Model environment. The CEF Workflow component provided the necessary elements to control system inputs, identify the work centers, the actual workflow itself, and as an output, all of this was displayed on the CEF Dashboard. The CEF Product Control was the main component for quality assurance activities, in which every single product that was developed by the Cloud ERP Production environment had to go through quality control procedures. This component was also concerned with providing the necessary information that comes with the developed product in terms of identification part number, version, and revision. Software Configuration Management concept was largely used to identify the elements used in this component. With everything that has been accumulated in terms of knowledge and skills, Knowledge Management was the most logical and novel component to be included in the CEF Model.

The Cloud ERP Factory model was constructed using specific core model elements with distinct guidelines and checklists. Each component of the CEF Model was identified with its own set of guidelines and checklists, which were added later after expert review was conducted. The CEF Product Line component consisted of three core model elements, which were System Functional Architecture, Module Architecture, and Integration Model. The System Functional Architecture defined necessary type of product and product family grouping to support Cloud ERP. Modularity was one of the key factors, allowing the product to function individually yet able to be integrated as and when required. The core model elements for the CEF Platform were User Portal Platform, Developer's Platform, and Backup or Redundancy. User Portal was linked heavily with the ability of the platform to provide access to the applications, security management, user accessibility management, and interoffice communication. The CEF Workflow core model elements were identified to be System Input identification, Work Centers identification, Job and Task listing, Work Order Routing, and System Output. Each element provided the necessary tools to run the whole workflow in a more process-centric software production environment. The CEF Product Control has explained the focus on product quality as well as product identification through deployment and updates. Its core model elements were identified as System Identification, Version and Revision control, Release Control model, Software Updates, and Anti-Software Infringement. The Knowledge Management component was closely related to the Learning Management System, as the present research based the component and core model elements from the theory.

### 7.2.2 Model Verification: The CEF Prototype Environment.

Once the model was constructed and validated through Expert Review, the first research question was answered and the first specific objective was met. However, in order to achieve the main objective, the model needs to be verified to be deployable in an existing Cloud ERP production facility. This activity constitutes the second research objective as well as the answer to the second research question. The Cloud ERP Factory model was proven to be technically feasible by being able to be successfully deployed through the model environment prototyping stage (See Chapter Five). Using the CEF Model Deployment Steps, those are Modeling Requirement Study, Model to System Development, Model Checklist Verification and finally Environment Training and Implementation, the model was successfully deployed into the existing Cloud ERP Production environment. This was further verified by a built-in checklist and has satisfied all of the checklist requirements.

The process of deploying the CEF Model into the validation vehicle can be utilized as a guide for future replication of the model deployment. CEF Modeling Requirement Study is particularly important as it is assumed that all Cloud software production facility may be unique. The CEF Model is designed to be flexible to be adapted into various software production facility. The next phase of converting the requirement into a solution was done by developing the necessary software system to fulfill the CEF Modeling Requirement. Four systems were developed, which are Job Management System, Product Management System, Document Control System and Learning Management System to create the CEF environment. The next process was verifying the systems with the built-in verification checklist. The previous steps were reiterated until the checklist was fully satisfied. The final and important step, which is user training was successfully conducted in three phases.

Many valuable lessons were learned throughout the entire process of model development as well as during the prototyping and real-world environment evaluations. This crucial information was discovered by first-hand observations made through interaction with the actual process of testing the CEF Model. Furthermore, the CEF Model can even be utilized by industry professionals as a blueprint to design a Software Factory for a Cloud ERP Software Development Facility.

### 7.2.3 Model Validation: Field Usability Findings

In answering the third research question, the model was validated using three field usability case studies. The three case study findings prove that the model is usable in solving the research problem components. The following listing elaborates the observed outcome for each of the research problem components to the CEF Model implementation.

i. Complexity

The CEF Model has demonstrated that it is able to reduce the complexity of any ERP implementation project by breaking down the complexity of the whole project into many smaller and specific jobs assigned to actors. The actors whom are already skillful in their job scope will not be burdened by the complexity of managing the whole project. By doing the same process over a period of time, the actors would then be more skillful in the specific task, thus be able to handle the process more efficiently. Work order template with job routing table element in the CEF Workflow component introduces the automation of the development work. It is in a way can be regarded as an artificial intelligence in the development process. The CEF Dashboard output and CEF Performance Monitoring element enable Capacity Planning features that can further simplify the complexity of managing a Cloud ERP solution. The elements in CEF Product Control further reduces the complexity from managing the quality aspect of Cloud ERP product. Finally, by having a knowledge repository with dynamic retrieval and logging, the difficulty of managing the wide body of knowledge is significantly reduced.

ii. High Workload

Through the model validation result, the CEF Model has demonstrated that it is able to tackle the high workload issue by distributing the overall workload across the whole team. Although the overall workload to develop Cloud ERP solution remains, the effective workload for each actor is significantly reduced. The component based oriented features in CEF Product Line and CEF Platform promote high degree of reusability, thus minimizing the effort required to handle high number of Cloud ERP products. It is observed that dynamic knowledge retrieval and logging reduces the workload by reducing the effort and time spent in searching for the related documentation or resources for the task. The automation of job scheduling by work order template and CEF Dashboard reduces the human interaction effort in communication and project meetings.

iii. Inconsistent Software Quality

The feedback gathered in the field usability case studies indicated that the CEF Model with process based approach clearly improved the inconsistent software quality issue. By allowing actors to focus on a specific process, they would be more skillful in their respective area. There will be no more issues of competent developers managing a particular project and lesser competent ones managing another. In general, the software quality would be more consistent as all the software is required to go through the same series of processes. Therefore, most actors will have hands-on involvement in all projects and the software inconsistency issue is minimized. The CEF Workflow component further guarantees the software consistency by enforcing the process workflow using Work Order routing template. In the long run, knowledge retrieval and dissemination in the CEF Knowledge Management component will improve the learning cycle of new hires or less competent actors.

iv. Knowledge Retention Problem

Theoretically, the CEF Knowledge Management component should solve the knowledge retention issue in the Cloud ERP production environment. The validation survey data has also shown that most actors believe that the model will be effective in retaining the knowledge. While this is true, the adaptation of the LMS culture may require time and perhaps an enforcement policy is required to make it happen. On the other hand, it is observed that knowledge identification and knowledge to job mapping elements are critical in ensuring the effectiveness of the CEF Knowledge Management component. The CEF Knowledge Management component enables the accumulation of new knowledge acquired from each deployment project over time. After the implementation of the model, it is noted that the impact of staff turnover is lessened to a certain degree as the critical knowledge has been captured in the knowledge repository. However, the challenge lies in the ability to ensure that all the critical information has been logged flawlessly.

The field usability case studies validated the model's ability to address the management complexities of Cloud ERP, high workload, inconsistent software quality, and knowledge retention management within the implemented CEF Model in the Cloud ERP Production environment and returned with the result that the model is usable. For each of the research problem component, the field usability case studies have yielded survey results with an average of more than 5 on the Likert Scale. The model's general usability has scored with an average of 5.93/7 and the model's ability to address the research problem components, which are Complexity, Workload, Quality, and Knowledge Management, which scored an average mean of 5.9/7, 6.1/7, 5.9/7, and 6.1/7 respectively. This is a clear indication that the model was able to address the research problem components and is usable.

## 7.3 Research Contribution and Deliverables

The following subsections will elaborate on the deliverables of this study. The CEF Model and its environment prototyping processes constitute the two significant contribution deliverables of this research.

### 7.3.1 The CEF Model

The main contribution of this research is the CEF model and its development guides. The innovative aspect of this research is a new paradigm concept of Cloud ERP system development model that can minimize the inherent Cloud ERP development problems by using process base oriented approach mimicking factory model. The CEF Model along with its' five submodels; those are Product lines, Platform Architecture, Product Control, Workflow, and Knowledge Repository Architecture are presented with a comprehensive guideline that focuses on easy deployable into Cloud ERP production environment. The CEF model was presented with the intention to be replicated by other Cloud ERP production environment with detailed design guideline elements, which were grouped by the five subcomponents of the model. It is also can be viewed as design model, which abstractly provides a blueprint of the important elements needed to be considered in any Cloud ERP production. Built-in model verification checklist should also help in ensuring that the model is implemented correctly. The following list further discuss the submodel components and their functional aspects.

i. CEF Product Line

This component assists the organization to define Cloud ERP products into a scalable model that emphasizes on reusable component based. System functional architecture, system module component and integration make up the major elements for this submodel.

ii. CEF Platform

The second component requires the Cloud ERP to have a platform as a foundation for the developed products. This is an extension of Product Line that emphasizes on a core asset for product's scalability. The model specifies user platform, system platform, backup and redundancy as the major element to support the submodel.

iii. CEF Workflow

CEF Workflow submodel depicted a structure that makes the Cloud ERP production to work like a factory operation. It is similar to a production shop floor where system input, work center, job and task library, work order routing template, system output dashboard and the system output monitoring are the elements described in the submodel. This submodel makes use of ticketing and job bus line to emulate software development workflow. This component enables productivity and performance monitoring using the system dashboard.

iv. CEF Product Control

Product control represents quality assurance aspect of the Cloud ERP products. System identification, version and revision control, release control model, software updates and anti-software infringement control make up the major elements of this submodel. This component delivers the challenge of high quality expectation expected for Cloud ERP products.

v. CEF Knowledge Management

Knowledge management component promotes sustainability aspect and business continuity by managing the knowledge within the organization's operation. Knowledge identification and segmentation, knowledge retrieval and dissemination, dynamic knowledge logging and knowledge to task/job mapping make up the major elements of this submodel.

The model is further supported with the deployment steps to guide other Cloud ERP in deploying the CEF Model into their environment. System modelling requirement study, model to system development, model checklist verification and CEF training implementation are the steps and guide for the model implementation. Model verification step is expected to make use of the verification checklist provided in each submodel.

It is expected that the model artifact can help industry practitioners and academician as theoretical and practical guidance resource.

### 7.3.2 Prototype of the model environment in the form of a commercial CEF production and support environment

The development of the prototype environment explained in Chapter Five is another main contribution of this research. The steps involved during the implementation those are System Modeling Requirement Study, Model To System Development, Model Checklist Verification and CEF Training Implementation can be used as a reference to replicate the model into the Cloud ERP environment or even to a similar environment from other industries. The prototype environment model and the system developed along with it is another indirect contribution that can be used as a reference to practitioners or academician.

Although, the CEF model is meant for Cloud ERP industry, there are infinite number of possibilities in adapting the model into other domains. The model or its subcomponent itself could be deployed into other industries where applicable. For example, the Workflow, Product Control, Knowledge Management subcomponents are universally adaptable in developing the process oriented approach or system in other service based organizations. The Artificial Intelligence of the Work Order Routing Template from Workflow Component could be adapted in converting the standard office practice into automated job delegation automation process. Quality control processes explained in the CEF Product Control component could also be used in software product or service quality policy. On the other hand, CEF Knowledge Management component could obviously be applied to any organization that require knowledge retention solution.



Finally, the lesson learned during the verification and validation processes could also help future implementations. Below are the listing of observations and findings throughout the entire process of model verification and validation for the CEF Model.

i. Require a huge commitment from the management and all actors as a deployment can be considered as a major business process reengineering.
ii. The company is expected to be very knowledgeable in developing software systems.

iii. Although the prototyped environment has been validated successfully, continuous improvement such as system enhancement and comprehensive reporting features are expected to further enhance the environment.

iv. CEF Workflow component stands out as the heart of the CEF model as it reflects the factory-like automation. It can also be adapted in other research such as Industry 4.0.

v. The CEF Model has demonstrated that splitting the complex project based system into process component based could produce higher degree of productivity and efficiency of Cloud ERP production environment. It could also mean that it can be applied to other industries and be expected to yield similar results.

vi. Defining Cloud ERP product line is a major step in creating the CEF environment as it forces the Cloud ERP provider to clearly define a scalable product family, which are also customizable.

vii. This research has also demonstrated that the ability of Cloud ERP industry to adapt mass customization and perhaps the CEF Model component and its steps could also be applied to other industries to enable mass customization.

viii. The company should adopt serious quality policy commitment as it is aligned with product control and as Cloud ERP products require a very stringent quality expectation.

ix. LMS component promises to solve knowledge retention management and sustainability of an organization. However, the culture requires time to be cultivated to the actors. Dynamic data logging for example requires a major cultural shift.

## 7.4 Limitations and Recommendations

There were limitations during our research, as model development and the validation requires time, due to the fact that a software system was needed to be developed in order to provide the tools. As with any real case studies, the overall system implementation of the CEF Model was based on the availability of the Cloud ERP production company, hence causing the overall time frame to drag.

It was also discovered that Cloud ERP itself was such a huge undertaking, wherein the scope of each component is recommended to be researched further to study other possibilities. During the field usability, it was realized that with every single cycle, there was a significant increase in terms of speed, quality, and overall adaptation of the CEF Model environment. The system capability was also expanded with every new implementation, which evidently has provided benefits in terms of efficiency and effectiveness.



The research result recommends the CEF model to be adapted as the development model of Cloud ERP for system providers. The benefit is particularly true for the ERP providers with significant numbers of customers in order to achieve economy of scale. Traditional Project Based Approach could still be useful for in house one-time Cloud ERP development effort. Besides, other external factor that have been scoped out such as Financial and human resource limitations may result in favors to the Project based model. The serious effort in setting up the CEF model, the lengthy duration of deployment and its maturity may be the challenges to the adoption.

Despite the mentioned challenges, the model is recommended to be deployed and further researched to improve it in all aspects. The adaptation of the model into other industries is obviously possible especially if we look at the CEF Workflow component that promotes systematic automation of any processes. The Artificial Intelligence of Job Routing Engine from the Workflow component could lead to a better decision making compared to decision made by human project manager. Along with the other four components, Knowledge Management component automatically provide a self-learning mechanism of the development facility. It gets better over time. The more it is being used, the more intelligent it becomes. These ideas deserve future research exploration that could reveal more potentials of CEF model or software factory. The CEF model or its components could also be explored with other research area such as Internet of Things (IoT) and Industry 4.0. While IoT and Industry 4.0 focus on machine automation, CEF model provides human resource automation. One thing these technologies have in common is that they promote higher productivity.

What makes the CEF Model novel and significantly better than the project development approach is its ability to maximize productivity and streamline the process oriented approach in managing Cloud ERP production. The model enables each actor to focus on a specific job thus creating a better work environment compared to project-based method which requires a lot of travelling and human interaction through meetings. The process based focus in the CEF Model promotes the actors to be more skillful over time in the specific tasks. This could also promote reduction of development time but future research may be required to validate this observation. Indirectly, the CEF Model implementation has also proved to be able to enhance quality regardless of the type of products produced.  Capacity Planning and the Job Management Dashboard were combined to form the ultimate outcome of the entire CEF model demonstration, wherein they act as a central brain designating incoming jobs to the appropriate Work-Center for that specific task. It is also observed that the CEF Model evolves with maturity with each consecutive implementation cycle being carried out. Due to the huge research scope of Cloud ERP, it has taken approximately four years to complete the whole research cycle. It is obviously possible that the model can be further improved and expanded upon with future research by focusing on each subcomponent in detail and future research into similar models in other domains is also possible.

Software factory and other theories has been well collectively incorporated into the model. Therefore, it is technically possible to adapt the CEF model to solve similar problems from other industries. The knowledge from the research artefact, which is the model and its five components, can be a useful reference to be embraced by other industries in constructing relevant development tools, systems or models. As such, corporate or academic institution should be able to utilize the CEF Workflow and Knowledge Management component to introduce an automation process dimension in line with Industry 4.0 automation concept. This also allows for a long sustainable knowledge management program by utilizing the Learning Management System suggested in the CEF Knowledge Management component.



### 7.5 Conclusion

In summary, the research has been able to meet its primary objective and all its specific objectives. The model was successfully developed, validated, and refined through expert review. It was proven to be technically feasible as it was successfully deployed into the selected Cloud ERP production facility. The model validation stage has provided clear insight on how to deploy the CEF Model into a real Cloud ERP production environment, thus making the model replicable in other similar Cloud

ERP production environments. Through the case studies conducted with the survey results analysis, it has been confirmed that the proposed CEF Model can be used to tackle the complexity of managing Cloud ERP solutions, high workload, inconsistent software quality, as well as knowledge retention problem. This research was successfully carried out in a rigorous and systematic manner and has achieved its main objective, which is to produce a model that can be used to solve Cloud ERP developers' inherent problems and consequently fill in the gaps discovered in the presented previous research.

# REFERENCES


Abu Bakar, M. S. (2012). *Model Reka Bentuk Konseptual Operasian Storan Data Bagi Aplikasi Kepintaran Perniagaan*. Universiti Utara Malaysia.

Abu Bakar, Z. (2015). *The Moderating Effect of Information Technology Capability on the Relationship Between Business Continuity Management Factors and Organizational Performance*.

Ahmad, M. O., Liukkunen, K., & Markkula, J. (2014). Student perceptions and attitudes towards the software factory as a learning environment. *IEEE Global Engineering Education Conference, EDUCON*, (April), 422–428. https://doi.org/10.1109/EDUCON.2014.6826129

Alouane, M., & Bakkali, H. E. L. (2015). Security , Privacy and Trust in Cloud Computing : A Comparative Study. *Cloud Technologies and Applications (CloudTech), 2015 International Conference on*, 1–8. https://doi.org/10.1109/CloudTech.2015.7336995

Alter, S. (2013). Work System Theory : Overview of Core Concepts , Extensions , and Challenges for the Future Work System Theory : Overview of Core Concepts , Extensions , and Challenges for the Future. *Journal of the Association for Information Systems*, *14*(February), 72.

Alter, S., & Ph, D. (2011). Using Work System Theory to Achieve Greater Business Value from ERP by Exploring Alternative Design Spaces ERP Projects or Work System Projects Enabled by ERP ? Background about Work System Theory, (January), 1–12.

Anwar, M., & Imran, A. (2015). Access Control for Multi-tenancy in Cloud-Based Health Information Systems. *2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing*, 104–110. https://doi.org/10.1109/CSCloud.2015.95

Appandairajan, P., Zafar Ali Khan, N., & Madiajagan, M. (2012). ERP on Cloud: Implementation strategies and challenges. *2012 International Conference on Cloud Computing Technologies, Applications and Management (ICCCTAM)*, 56–59. https://doi.org/10.1109/ICCCTAM.2012.6488071

Arinze, B., & Anandarajan, M. (2003). A Framework for Using OO Mapping Methods to Rapidly Confiure ERP Systems. *Communications of the ACM*, *46*(2), 61–65.

Armbrust, M., Joseph, A. D., Katz, R. H., & Patterson, D. A. (2009). Above the Clouds : A Berkeley View of Cloud Computing. *Science*, *53*(UCB/EECS-2009-28), 07–013. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.149.7163&rep=rep1&type=pdf

Arnesen, B. S. (2013). Is a Cloud ERP Solution Right for You? *Strategic Finance*, *95*(2), 45–50. Retrieved from http://libaccess.mcmaster.ca/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=85354154&site=bsi-live

Ashalatha, R., & Agarkhed, J. (2016). Multi Tenancy Issues in Cloud Computing for SaaS Environment.

Badger, L., Grance, T., Patt Corner, R., & Voas, J. (2011). Cloud Computing Synopsis and Recommendations. *Recommendations of the National Institute of Standards and Technology*, *C O M P*, 84. https://doi.org/2012

Bakshi, K. (2014). Secure hybrid cloud computing: Approaches and use cases. *IEEE Aerospace Conference Proceedings*, 1–8. https://doi.org/10.1109/AERO.2014.6836198

Ballou, D. P., & Pazer, H. L. (2003). Modeling completeness versus consistency tradeoffs in information decision contexts. *IEEE Transactions on Knowledge and Data Engineering*, *15*(1), 241–244. https://doi.org/10.1109/TKDE.2003.1161595

Banerjee, M., & Kumari, R. (2016). Dynamic Software Product Line : An Approach to Dynamic Binding.

Batini, C., Cappiello, C., Francalanci, C., & Maurino, A. (2009). Methodologies for data quality assessment and improvement. *ACM Comput. Surv.*, *41*(3), 1–52. https://doi.org/10.1145/1541880.1541883

Benalil, A., Asri, B. E. I., & Kriouile, H. (2015). Cloud Environment Assignment : A Context-aware and Dynamic Software Product Lines-Based Approach.

Bento, F., & Costa, C. J. (2013). ERP measure success model; a new perspective. *Proceedings of the 2013 International Conference on Information Systems and Design of Communication - ISDOC '13*, 16. https://doi.org/10.1145/2503859.2503863

Betts, D., Homer, A., Jezierski, A., Narumoto, M., & Zhang, H. (2012). *Developing Multi-tenant Applications for the Cloud on Windows Azure (Microsoft patterns & practices)*.

Bezemer, C. P., Zaidman, A., Platzbeecker, B., Hurkmans, T., & Hart, A. (2010). Enabling multi-tenancy: An industrial experience report. *IEEE International Conference on Software Maintenance, ICSM*. https://doi.org/10.1109/ICSM.2010.5609735

Bhattacharya, P. (2009). Dynamic Updates for Web and Cloud Applications. *Analysis and Programming Languages for Web Applications and Cloud Applications*.

Boehm, B. (1991). Software risk management: principles and practices. *IEEE Software*, *8*, 32–41. https://doi.org/10.1109/52.62930

Boehm, B. W. (1979). Guidelines for Verifying and Validating Software Requirements and Design Specifications. *Euro IFIP 79*, (1), 711–719. https://doi.org/10.1109/MS.1984.233702

Bond, B., Genovese, Y., Miklovic, D., Zrimsek, B., & Rayner, N. (2005). 4 October 2000 ERP Is Dead — Long Live ERP II, (October 2000).

Bosch, J., Florijn, G., Greefhorst, D., Kuusela, J., Obbink, H., & Pohl, K. (2002). Variability Issues in Software Product Lines. *Software ProductFamily Engineering*, *2290*, 13–21. https://doi.org/10.1007/3-540-47833-7_3

Bovee, M., Srivastava, R., and Mak, B. (2001). A conceptual framework and belief-function approach to assessing overall information quality. Paper presented at the Proceedings of the 6th International Conference on Information Quality, *18*, 311–328.

Bratman, H., & Court, T. (1975). The Software Factory. *Computer*, *8*(5), 28–37. https://doi.org/10.1109/C-M.1975.218953

Bryan, G. E. (2012). Not all programmers are created equal - Redux. *Aerospace Conference, 2012 IEEE*, (March 1980), 1–10. https://doi.org/10.1109/AERO.2012.6187328

Cai, H., Wang, N., & Zhou, M. J. (2010). A transparent approach of enabling saas multi-tenancy in the cloud. *Proceedings - 2010 6th World Congress on Services, Services-1 2010*, 40–47. https://doi.org/10.1109/SERVICES.2010.48

Camisón, C., & Villar López, A. (2010). An examination of the relationship between manufacturing flexibility and firm performance. *International Journal of Operations & Production Management*, *30*(8), 853–878. https://doi.org/10.1108/01443571011068199

Canuel, V., & Robert, M. (2012). Toward A Better Software Factory.

Castellina, N. (2011). SaaS and Cloud ERP - Trends, Observations, and Performance, (December), 27. https://doi.org/10.1017/CBO9781107415324.004

Castro, H., Putnik, G. D., & Shah, V. (2012). A review of agile and lean manufacturing as issues in selected international and national research and development programs and roadmaps. *The Learning Organization*, *19*(3), 267–289. https://doi.org/10.1108/09696471211220064

Chen, H., Yu, J., Hang, H. C., Zang, B., & Yew, P. C. (2011). Dynamic software updating using a relaxed consistency model. *IEEE Transactions on Software Engineering*, *37*(5), 679–694. https://doi.org/10.1109/TSE.2010.79

Cisl, W. P. (2013). Challenges in Implementing Enterprise Resource Planning ( ERP ) system in Large Organizations : Similarities and Differences Between Corporate and University Environment Goeun Seo Challenges in Implementing Enterprise Resource Planning ( ERP ) System in L, (May).

Comitz, P. H., & Pinto, A. (2006). A software factory for air traffic data. *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, 1–7. https://doi.org/10.1109/DASC.2006.313758

Conradi, R., & Westfechtel, B. (1998). Version Models for Software Configuration Management. *ACM Comput. Surv.*, *30*(2), 232–282. https://doi.org/10.1145/280277.280280

Corrall, S. (2010). Cloud ERP: a New Dilemma to Modern Organisations? *Journal of Computer Information Systems*, *64*, 49–67. https://doi.org/10.1258/itt.2010.100803

Cuatrecasas-Arbós, L., Fortuny-Santos, J., Ruiz-de-Arbulo-López, P., & Vintró-Sanchez, C. (2015). Monitoring processes through inventory and manufacturing lead time. *Industrial Management & Data Systems*, *115*(5), 951–970. https://doi.org/10.1108/IMDS-12-2014-0375

Dahlstrom, E., Brooks, D. C., & Bichsel, J. (2014). The Current Ecosystem of Learning Management Systems in Higher Education: Student, Faculty, and IT Perspectives. *EDUCAUSE Research Report*, 27. Retrieved from http://www.educause.edu/ecar

Davenport, T. H. (2016). Putting the Enterprise into the Enterprise System, (JULY 1998).

Davis, S. M. (1989). From “future perfect”: Mass customizing. *Planning Review*, 16-21.

Davis, F. . (1989). Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology. Retrieved December 30, 2016, from http://hcibib.org/bs.cgi?searchtype=question&query=J.MISQ.13.3.319

de Jesus Souza, Magno Lua Santos, A. R., & de Almeida, E. S. (2015). Towards the Selection of Modeling Techniques for Dynamic Software Product Lines. *Proceedings of the Fifth International Workshop on Product LinE Approaches in Software Engineering*, 19–22. https://doi.org/10.1109/PLEASE.2015.13

De Silva, D., & Alahakoon, D. (2010). Incremental knowledge acquisition and self learning from text. *Proceedings of the International Joint Conference on Neural Networks*. https://doi.org/10.1109/IJCNN.2010.5596612

Decker, M. D. (1992). Continuous Quality Improvement.

Delikostidis, I. (2007). *Methods and techniques for field-based usability testing of mobile geo-applications*.

Demarco, T., & Lister, T. (1985). Programmer performance and the effects of the workplace. *ICSE ’85 Proceedings of the 8th International Conference on Software*, 268–272.

Dillon, T., Wu, C., & Chang, E. (2010). Cloud Computing: Issues and Challenges. *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, 27–33. https://doi.org/10.1109/AINA.2010.187

Dittrich, Y., & Vaucouleur, S. (2008). Practices around customization of standard systems. *Proceedings of the 2008 International Workshop on Cooperative and Human Aspects of Software Engineering - CHASE '08*, 37–40. https://doi.org/10.1145/1370114.1370124

Dittrich, Y., Vaucouleur, S., & Giff, S. (2009). ERP customization as software engineering: Knowledge sharing and cooperation. *IEEE Software*, *26*(6), 41–47. https://doi.org/10.1109/MS.2009.173

Donohoe, P. J. (2009). Introduction to software product lines, 305. Retrieved from http://dl.acm.org/citation.cfm?id=1753235.1753282

Duan, J., Faker, P., Fesak, A., & Stuart, T. (2012). Benefits and Drawbacks of Cloud-Based Versus Traditional ERP Systems.

Eleutério, J. D. A. S., Gaia, F. N., Bondavalli, A., Lollini, P., Rodrigues, G. N., & Rubira, C. M. F. (2016). On the Dependability for Dynamic Software Product Lines A Comparative Systematic Mapping Study. *Proc. of SEAA*, 323–330. https://doi.org/10.1109/SEAA.2016.40

Emig, C., Langer, K., Krutz, K., & Link, S. (2006). The SOA's Layers. *Cooperation & Management, Universität Karlsruhe, Karlsruhe*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.172.21&rep=rep1&type=pdf

Estublier, J. (2000). Software Configuration Management: A Roadmap. *Future of Software Engineering (FOSE 2000), Conference on The*, 279–289. https://doi.org/10.1145/336512.336576

Fernstrom, C., Narfelt, K. H., & Ohlsson, L. (1992). Software factory principles, architecture, and experiments. *IEEE Software*, *9*(2), 36–44. https://doi.org/10.1109/52.120600

Gabriel, T. J. (2013). Manufacturing Complexity : the Effects of Common Attributes of Manufacturing System Design on Performance. *Academy of Information and Management Sciences Journal*, *16*(1), 2013.

Gartner. (2016). Gartner Says By 2016, the Impact of Cloud and Emergence of Postmodern ERP Will Relegate Highly Customized ERP Systems to "Legacy" Status.

Ghahfarokhi, A. D., & Zakaria, M. S. (2009). Knowledge Retention in Knowledge Management System : Review. *2009 International Conference on Electirical Engineering and Informatics*, (August), 343–347.

Ghosh, R. (2012). A Comprehensive Study on ERP Failures Stressing on Reluctance to Change as a Cause of Failure. *Journal of Marketing and Management*, *3*(May), 123–134.

Gomaa, H., & Saleh, M. (2006). Software Product Line Engineering and Dynamic Customization of a Radio Frequency Management System. *IEEE International Conference on Computer Systems and Applications, 2006.*, 345–352. https://doi.org/10.1109/AICCSA.2006.205112

Greenfield, J., & Short, K. (2003). Software factories: assembling applications with patterns, models, frameworks and tools. *Management*, 16–27. https://doi.org/10.1145/949344.949348

Gu, T., Cao, C., Xu, C., Ma, X., Zhang, L., & Lu, J. (2012). Javelus: A low disruptive approach to dynamic software updates. *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, *1*, 527–536. https://doi.org/10.1109/APSEC.2012.55

Gulati, V. P., & Srivastava, S. (2014). Knowledge Management Strategy and Structure in Service Sector. *2014 International Conference on Computational Science and Computational Intelligence*, 334–338. https://doi.org/10.1109/CSCI.2014.64

Gupta, B., S.Iyer, L., & Aronson, J. E. (2000). Knowledge Management: Practices and Challenges, *100*(1), 17–21.

Gupta, V., Acharya, P., & Patwardhan, M. (2013). A strategic and operational approach to assess the lean performance in radial tyre manufacturing in India. *International Journal of Productivity and Performance Management*, *62*(6), 634–651. https://doi.org/10.1108/IJPPM-Jun-2012-0057

Helton, D. (2010). The Pontential of Software Factories, *3*(7), 260–266.

Hevner, A., March, S. T., & Park, J. (2004). Design Science in information systems research. *Mendeley Desktop*, (March), 1–16. https://doi.org/10.1007/s11576-006-0028-8

Hofmann, P., & Woods, D. (2010). Cloud computing: The limits of public clouds for business applications. *IEEE Internet Computing*, *14*(6), 90–93. https://doi.org/10.1109/MIC.2010.136

Huynh, S., & Cai, Y. (2007). An Evolutionary Approach to Software Modularity Conformance Checking. *Cs.Drexel.Edu*. https://doi.org/10.1109/ACOM.2007.1

IEEE. (1998). IEEE Standard for Software Verification and Validation.

Isa, W. A. R. W. M., Lokman, A. M., Wahid, E. S. A., & Sulaiman, R. (2014). Usability testing research framework: Case of Handicraft Web-Based System. *2014 2nd International Conference on Information and Communication Technology (ICoICT)*, 199–204. https://doi.org/10.1109/ICoICT.2014.6914065

Ishak, M. K., & Ahmad, N. B. (2016). Enhancement of Learning Management System with Adaptive Features. *2016 Fifth ICT International Student Project Conference (ICT-ISPC) Enhancement*. https://doi.org/10.1109/ICT-ISPC.2016.7519230

Jain, S. K. (2012). An evaluation of ISO 9000 initiatives in Indian industry for enhanced manufacturing performance. *International Journal of Productivity and Performance Management*, *61*(7), 778–804. https://doi.org/10.1108/17410401211263863

Jarke, M., List, T., & Köller, J. (2000). The Challenge of Process Data Warehousing. *26th International Conference on Very Large Databases*, 473–483.

Jarke, M., & Vassiliou, Y. (1997). DataWarehouseQualityAReviewofDWQProject.pdf. *2nd Conference on Information Quality, Massachusetts Institute of Technology, Cambridge.*

Jegadeesan, H., & Balasubramaniam, S. (2009). A method to support variability of enterprise services on the cloud. *CLOUD 2009 - 2009 IEEE International Conference on Cloud Computing*, 117–124. https://doi.org/10.1109/CLOUD.2009.69

Jepsen, C. (2015). Does multi-tenancy really matter anymore?

Jhanwar, R., & Yaryan, T. (2007). Dynamic software update. *US Patent …*, (March). Retrieved from http://www.google.com/patents?hl=en&lr=&vid=USPAT7251812&id=oMWAAAAAEBAJ&oi=fnd&dq=Dynamic+Software+Update&printsec=abstract

Jiao, J., Simpson, T. W., & Siddique, Z. (2007). Product family design and platform-based product development: A state-of-the-art review. *Journal of Intelligent Manufacturing*, *18*(1), 5–29. https://doi.org/10.1007/s10845-007-0003-2

Ju, J., Wang, Y., Fu, J., Wu, J., & Lin, Z. (2010). Research on Key Technology in SaaS. *Intelligent Computing and Cognitive Informatics (ICICCI), 2010 International Conference on*, 384–387. https://doi.org/10.1109/ICICCI.2010.120

Junqueira, D. C., Bittar, T. J., & Fortes, R. P. M. (2008). A fine-grained and flexible version control for software artifacts. *Proceedings of the 26th Annual ACM International Conference on Design of Communication*, 185–192. https://doi.org/http://doi.acm.org/10.1145/1456536.1456576

Kamma, D., G, G., & J Neela, P. (2013). Countering Parkinson ' s Law for Improving Productivity Categories and Subject Descriptors.

Kamoun, A., Kacem, M. H., & Kacem, A. H. (2016). Multiple software product lines for service oriented architecture. *Proceedings - 25th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE 2016*. https://doi.org/10.1109/WETICE.2016.21

Kamrani, A., Smadi, H., & Salhieh, S. M. (2012). Two-phase methodology for customized product design and manufacturing. *Journal of Manufacturing Technology Management*, *23*(3), 370–401. https://doi.org/10.1108/17410381211217425

Kang, K. C. (2013). Modularity in the Context of Product Line Variability. *Proceedings of the 12th Annual International Conference Companion on Aspect-Oriented Software Development*, 3–4. https://doi.org/10.1145/2457392.2457394

Kantner, L., Sova, D. H., & Rosenbaum, S. (2003). Alternative methods for field usability research. *Proceedings of the 21st Annual International Conference on Documentation SIGDOC 03Kantner, L., Sova, D. H., and Rosenbaum, S. 2003. "Alternative Methods for Field Usability Research," Proceedings of the 21st Annual International Conference on Documentatio*, 68. https://doi.org/10.1145/944868.944883

Kiadehi, E. F., & Mohammadi, S. (2012). Cloud ERP: Implementation of Enterprise Resource Planning Using Cloud Computing Technology. *J. Basic. Appl. Sci. Res*, *2*(11), 11422–11427.

Kim, W., Kim, S. D., Lee, E., & Lee, S. (2009). Adoption issues for cloud computing. *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia - MoMM '09*, 2. https://doi.org/10.1145/1821748.1821751

Kotiadis, K. (2008). Proceedings of the 2008 Winter Simulation Conference S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds., 951–958.

Krebs, R., Momm, C., & Kounev, S. (2012). Architectural Concerns in Multi-tenant SaaS Applications. *Closer*, 426–431. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-84864882112&partnerID=40&md5=a71b38428af9f8364a66ee67f8c8b2f1%5Cnhttps://sdqweb.ipd.kit.edu/publications/pdfs/KrMoKo2012-multitenant-sass.pdf

Krejcie, R. V, & Morgan, D. W. (1970). Determining Sample Size for Research Activities Robert. *Educational and Psychological Measurement*, *38*(1), 607–610. https://doi.org/10.1177/001316447003000308

Kriouile, H., El Asri, B., El Haloui, M., & Benali, A. (2015). Towards deployment of a user-aware tenancy using rich-variant components for SaaS applications. *Proceedings of 2015 International Conference on Cloud Computing Technologies and Applications, CloudTech 2015*. https://doi.org/10.1109/CloudTech.2015.7337019

Kristianto, Y., Helo, P., & Takala, J. (2010). Strategic inventory allocation for product platform strategy. *Journal of Advances in Management Research*, *7*(2), 233–249. https://doi.org/10.1108/09727981011085011

Kumar, K., & Hillegersberg, J. V. (2000). ERP Experiences and Evolution. *Communications of the ACM*, *43*(4), 23–26.

Kumara, I., Han, J., Colman, A., Nguyen, T., & Kapuruge, M. (2013). Sharing with a difference: Realizing service-based SaaS applications with runtime sharing and variation in dynamic software product lines. *Proceedings - IEEE 10th International Conference on Services Computing, SCC 2013*, 567–574. https://doi.org/10.1109/SCC.2013.30

Kung, C. H., & Sölvberg, A. (1986). Activity modeling and behavior modeling. In *Proc. of the IFIP WG 8.1 working conference on Information systems design methodologies: improving the practice* (pp. 145–171). North-Holland Publishing Co. Amsterdam, The Netherlands, The Netherlands ©1986.

Landuyt, D. Van, Walraven, S., & Joosen, W. (2015). Variability Middleware for Multi-tenant SaaS Applications Categories and Subject Descriptors. *SPLC 2015 Nashville*, (i), 211–215. https://doi.org/10.1145/2791060.2791080

Lenart, A. (2011). ERP in the cloud - Benefits and challenges. *Lecture Notes in Business Information Processing*, *93 LNBIP*(Cc), 39–50. https://doi.org/10.1007/978-3-642-25676-9_4

Li, C., Li, H., & Li, M. (2001). A Software Factory Model based on ISO9000 and CMM for Chinese small organizations. *Proceedings - 2nd Asia-Pacific Conference on Quality Software, APAQS 2001*, 288–292. https://doi.org/10.1109/APAQS.2001.990033

Liao, C.-F., Chen, K., & Chen, J.-J. (2013). Modularizing tenant-specific schema customization in SaaS applications. *Proceedings of the 8th International Workshop on Advanced Modularization Techniques - AOAsia'13*, 9–12. https://doi.org/10.1145/2451469.2451473

Lim, N. K., Ang, J. S. K., & Parvi, F. N. (2000). Diffusing software-based innovation with a software factory approach for software development. *Proceedings of the 2000 IEEE International Conference on Management of Innovation and Technology: , 2*, 549–555. https://doi.org/10.1109/ICMIT.2000.916749

Linsbauer, L., Egyed, A., & Lopez-Herrejon, R. E. (2016). A variability aware configuration management and revision control platform. *Proceedings of the 38th International Conference on Software Engineering Companion - ICSE '16*, 803–806. https://doi.org/10.1145/2889160.2889262

Liu, K., Zou, D., & Jin, H. (2015). UaaS: Software Update as a Service for the IaaS Cloud. *Proceedings - 2015 IEEE International Conference on Services Computing, SCC 2015*, 483–490. https://doi.org/10.1109/SCC.2015.72

Liu, Z., Wong, Y. S., & Lee, K. S. (2011). A manufacturing-oriented approach for multi-platforming product family design with modified genetic algorithm. *Journal of Intelligent Manufacturing*, *22*(6), 891–907. https://doi.org/10.1007/s10845-009-0365-8

Lund, A. (Arnie) M. (2008). Measuring Usability with the USE Questionnaire. Retrieved December 30, 2016, from http://hcibib.org/bs.cgi?searchtype=question&query=U.Lund.2001

Luo, W., & Strong, D. M. (2004). A framework for evaluating ERP implementation choices. *IEEE Transactions on Engineering Management*, *51*(3), 322–333. https://doi.org/10.1109/TEM.2004.830862

Maglyas, A., Nikula, U., & Smolander, K. (2011). What do we know about software product management? - a systematic mapping study. *Software Product Management (IWSPM)*, 26–35. https://doi.org/10.1109/IWSPM.2011.6046201

Maheshwari, S., & Sharma, C. (2015). Ten security practices to a formidable ERP system. *2014 IEEE International Conference on "Smart Structures and Systems", ICSSS 2014*, 41–50. https://doi.org/10.1109/ICSSS.2014.7006193

Maiani de Mello, R., Nogueira Teixeira, E., Schots, M., Lima Werner, C. M., & Travassos, G. H. (2014). Verification of Software Product Line artefacts: A checklist to support feature model inspections. *Journal of Universal Computer Science*, *20*(5), 720–745.

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, *15*(4), 251–266. https://doi.org/10.1016/0167-9236(94)00041-2

Matinlassi, M. (2004). Comparison of Software Product Line Architecture Design Methods : Mari Matinlassi. *Engineering*, 127–136. https://doi.org/10.1109/ICSE.2004.1317435

Mei, H., Zhang, L., & Yang, F. (2001). A software configuration management model for supporting component-based software development. *ACM SIGSOFT Software Engineering Notes*, *26*(2), 53. https://doi.org/10.1145/505776.505790

Mell, P., Grance, T., & NIST. (2011). The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology. *Nist Special Publication*, *145*, 7. https://doi.org/10.1136/emj.2010.096966

Menascé, D. A., Krishnamoorthy, M., & Brodsky, A. (2015). Autonomic smart manufacturing. *Journal of Decision Systems*, *24*(2), 206–224. https://doi.org/10.1080/12460125.2015.1046714

Mietzner, R., Leymann, F., & Papazoglou, M. P. (2008). Defining composite configurable SaaS application packages using SCA, variability descriptors and multi-tenancy patterns. *Proceedings - 3rd International Conference on Internet and Web Applications and Services, ICIW 2008*, 156–161. https://doi.org/10.1109/ICIW.2008.68

Mijač, M., Picek, R., & Stapić, Z. (2013). Cloud ERP System Customization Challenges. *Central European Conference on Information and Intelligent Systems*, 132–140. Retrieved from http://www.ceciis.foi.hr/app/public/conferences/1/papers2013/637.pdf

Moens, H., & De Turck, F. (2014). Feature-based application development and management of multi-tenant applications in clouds. *Proceedings of the 18th International Software Product Line Conference on - SPLC '14*, *1*, 72–81. https://doi.org/10.1145/2648511.2648519

Mohd Yusof, S., & Aspinwall, E. (2000). Total quality management implementation framework:comparison and review. *Total Quality Management*, *11*(3), 281–294.

Mondal, S. C. (2016). Process capability – a surrogate measure of process robustness: a case study. *International Journal of Quality & Reliability Management*, *33*(1), 90–106. https://doi.org/10.1108/IJQRM-12-2013-0202

Moody, D., & Kortink, M. A. . (2000). From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design. *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2000)*, *2000*, 5–16.

Moura, L. F., Lima, E. P. De, Deschamps, F., & Gouvea, S. E. (2015). Manufacturing Strategy Design Process Improvement Through Enterprise Engineering Recommendations. *Proceedings of the 2015 Industrial and Systems Engineering Research Conference*, (2002), 2118–2127.

Mtebe, J. S. (2015). Learning Management System success: Increasing Learning Management System usage in higher education in sub-Saharan Africa. *International Journal of Education and Development Using Information and Communication Technology*, *11*(2), 51–64.

Müller, A.-L., & Pfleger, R. (2014). Business transformation towards sustainability. *Business Research*, *7*(2), 313–350. https://doi.org/10.1007/s40685-014-0011-y

N. Ghauri, P., & Grønhaug, K. (2005). *Research Methods in Business Studies: A Practical Guide*.

Nagpal, S., Khatri, S. K., & Kumar, A. (2015). Comparative study of ERP implementation strategies. *Systems, Applications and Technology Conference (LISAT), 2015 IEEE Long Island*, 1–9. https://doi.org/10.1109/LISAT.2015.7160177

Nepal, B. (2014). An integrated approach to optimize design, marketing, and manufacturing objectives during product development. *2014 International Annual Conference of the American Society for Engineering Management - Entrepreneurship Engineering: Harnessing Innovation, ASEM 2014*, (2000). Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-84925625060&partnerID=40&md5=fdeb8e1635d259557cf0a5f89763fc88

Nepal, B., Natarajarathinam, M., & Balla, K. (2011). Improving manufacturing process for biomedical products: a case study. *Journal of Manufacturing Technology Management*, *22*(4), 527–540. https://doi.org/10.1108/17410381111126436

NIST. (2011). The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology. *Nist Special Publication*, *145*, 7. https://doi.org/10.1136/emj.2010.096966

Nomura, L., Tonini, A. C., Hikage, O. K., & Tonini, A. C. (2007). A model for defining software factory processes.

Northrop, L. (2008). Software product lines essentials. *Software Engineering Institute, Carnegie Mel-Ion University*. Retrieved from http://resources.sei.cmu.edu/asset_files/Presentation/2008_017_001_24246.pdf%5Cnhttp://www.sei.cmu.edu/library/assets/spl-essentials.pdf

Offermann, P., & Platz, E. R. (2009). Outline of a Design Science Research Process.

Pal, S., Mandal, A. K., & Sarkar, A. (2015). Application Multi-Tenancy for Software as a Service. *ACM SIGSOFT Software Engineering Notes*, *40*(2), 1–8. https://doi.org/10.1145/2735399.2735412

Panorama Consulting. (2015). Key Findings from the 2015 ERP Report.

Panorama Consulting Solutions. (2016). 2016 Report on Erp Systems and Enterprise Software, 1–32.

Park, J., & Lee, N. (2006). A Conceptual Model of ERP for Small and Medium-Size Companies Based on UML, *6*(5), 42–49.

Parthiban, P., & Goh, M. (2011). An integrated model for performance management of manufacturing units. *Benchmarking: An International Journal*, *18*(2), 261–281. https://doi.org/10.1108/14635771111121702

Peloza, J., & Yachnin, R. (2008). Valuing Business Sustainability : A Systematic Review. *Research Network For Business Sustainability*, (November), 1–34.

Phillipo, J., & Krongard, S. (2012). Learning Management System ( LMS ): The Missing Link and Great Enabler. *Massachusetts ASCD Perspectives*, *Spring*, 1–7.

Piho, G., Tepandi, J., Roost, M., Parman, M., & Puusep, V. (2011). From archetypes based domain model via requirements to software: Exemplified by LIMS software factory. *34th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2011*, 570–575. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-80052280525&partnerID=40&md5=84e563673ead8fbdad83e08a60deb3bf

Pine, B. J. (1993). Making mass customization happen: Strategies for the new competitive realities. *Planning Review, Vol. 21 Issue:5*, 23-24.

Pipino, L. L., Lee, Y. W., & Wang, R. Y. (2002). Data quality assessment. *Communications of the ACM*, *45*(4), 211. https://doi.org/10.1145/505248.506010

Pohl, K., Metzger, A., & Pohl, K. (2006). Variability Management in Software Product Line Engineering. *29th International Conference on Software Engineering {(ICSE} 2007), Minneapolis, MN, USA, May 20-26, 2007, Companion Volume*, (November 2016), 186–187. https://doi.org/10.1109/ICSECOMPANION.2007.83

Pollock, N., Comford, J., Neil, P., & Comfordnclacuk, J. (2002). Fitting Standard Software to Non-Standard Organisations, 721–725.

Psomas, E. L., Fotopoulos, C. V., & Kafetzopoulos, D. P. (2011). Core process management practices, quality tools and quality improvement in ISO 9001 certified manufacturing companies. *Business Process Management Journal*, *17*(3), 437–460. https://doi.org/10.1108/14637151111136360

Purohit, G. N., Jaiswal, M. P., & Pandey, M. S. (2012). Challenges Involved in Implementation of ERP on Demand Solution : Cloud Computing. *International Journal of Computer Science Issues*, *9*(4), 481–489.

Rabelo, R. J., Noran, O., & Bernus, P. (2015). Towards the next generation service oriented enterprise architecture. *Proceedings of the 2015 IEEE 19th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, EDOCW 2015*, 91–100. https://doi.org/10.1109/EDOCW.2015.34

Raihana, G. F. H. (2012). Cloud erp – a solution model, *2*(1), 76–79.

Ravn, B. T., & Høedholt, M. (2008). Measuring and Monitoring the Quality of Master Data, (November), 1–7.

Reingruber, M. C., & W. Gregory, W. (1994). *The Data Modeling Handbook : A Best-Practice Approach to Building Quality Data Models*.

Rimal, B. P., Jukan, A., Katsaros, D., & Goeleven, Y. (2011). Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach. *Journal of Grid Computing*, *9*(1), 3–26. https://doi.org/10.1007/s10723-010-9171-y

Rippel, D., Ltjen, M., & Scholz-Reiter, B. (2014). A framework for the quality-oriented design of micro manufacturing process chains. *Journal of Manufacturing Technology Management*, *25*(7), 1028–1048. https://doi.org/10.1108/JMTM-11-2012-0110

Rittik, G., & Ghosh, R. (2012). A Comprehensive Study on ERP Failures Stressing on Reluctance to Change as a Cause of Failure. *Journal of Marketing and Management*, *3*(May), 123–134.

Robinson, S. (2010). Conceptual Modelling : Who Needs It? *SCS M&S Magazine*, *n2*(April), 1–7.

Rockwell, R., & Gera, M. H. (1993). The Eureka Software Factory CoRe: A Conceptual Reference Model for Software Factories. *Proceedings of the Software Engineering Environments Conference (SEE)*, 240. https://doi.org/10.1109/SEE.1993.388419

Rosenbaum, S., & Kantner, L. (2007). Field usability testing: Method, not compromise. *IEEE International Professional Communication Conference*, (November 2007). https://doi.org/10.1109/IPCC.2007.4464060

Rothenberger, M. A., & Srite, M. (2009). An investigation of customization in ERP system implementations. *IEEE Transactions on Engineering Management, 56*(4), 663–676. https://doi.org/10.1109/TEM.2009.2028319

Rubin, J., & Chisnell, D. (2008). *Handbook of usability testing [electronic resource] : How to plan, design, and conduct effective tests (2nd ed.). Indianapolis, IN: Wiley Pub.* https://doi.org/10.1007/s13398-014-0173-7.2

Sahin, N. Y. (2013). Cloud ERP Security : Guidelines for Evaluation, 11–38. Retrieved from http://www.diva-portal.org/smash/get/diva2:614895/ATTACHMENT01

Salim, S. A. (2013). Cloud ERP adoption-A process view approach. *Proceedings - Pacific Asia Conference on Information Systems, PACIS 2013,* (November), n. pag. https://doi.org/10.13140/RG.2.1.4900.8240

Satam, N., Taslim, J., Adnan, W. A. W., & Manaf, N. A. (2017). Usability testing of e-learning system: A case study on CeL in TARUC, Johor Branch Campus. *Proceedings - 2016 4th International Conference on User Science and Engineering, I-USEr 2016,* 63–68. https://doi.org/10.1109/IUSER.2016.7857935

Scheer, A.-W., & Habermann, F. (2000). Making ERP A Success, *43*(4), 57–61.

Schilling, A., Laumer, S., & Weitzel, T. (2012). Train and retain: The impact of mentoring on the retention of FLOSS developers. *SIGMIS-CPR'12 - Proceedings of the 2012 Computers and People Research Conference*, 79–83. https://doi.org/10.1145/2214091.2214112

Schroeter, J., Mucha, P., Muth, M., Jugel, K., & Lochau, M. (2012). Dynamic configuration management of cloud-based applications. *Proceedings of the 16th International Software Product Line Conference on - SPLC '12 -Volume 1, 2*, 171. https://doi.org/10.1145/2364412.2364441

Schubert, P., & Adisa, F. (2011). Cloud Computing for Standard ERP Systems : Reference Framework and Research Agenda. *Arbeitsberichte Aus Dem Fachbereich Informatik,* (16), 29. https://doi.org/10.1108/13598540510612721

Sekaran, U. (2013). *Research methods for business*. *Research methods for business* (Vol. 65). https://doi.org/10.1017/CBO9781107415324.004

Sellami, W., Kacem, H. H., & Kacem, A. H. (2014). Elastic Multi-tenant Business Process Based Service Pattern in Cloud Computing. *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*, 154–161. https://doi.org/10.1109/CloudCom.2014.151

Shahidul, M. I., Shazali, S. T. S., & Syed Shazali, S. T. (2011). Dynamics of manufacturing productivity: lesson learnt from labor intensive industries. *Journal of Manufacturing Technology Management*, *22*(5), 664–678. https://doi.org/10.1108/17410381111134491

Shaul, L., & Tauber, D. (2013). Critical success factors in enterprise resource planning systems. *ACM Computing Surveys (CSUR), 45*(4), 1–39. https://doi.org/10.1145/2501654.2501669

Shi, J., Dong, F., Zhang, J., Jin, J., & Luo, J. (n.d.). Resource Provisioning Optimization for Service Hosting on Cloud Platform, 1–6.

Shim, J., Han, J., Kim, J., Lee, B., Oh, J., & Wu, C. (2011). Patterns for configuration requirements of Software-as-a-Service. *Proceedings of the 2011 ACM Symposium on Applied Computing - SAC '11*, 155. https://doi.org/10.1145/1982185.1982222

Shneiderman, B. (1992). Designing the user interface. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, *3*(2), 193–200. https://doi.org/10.1016/0166-3615(93)90066-A

Singh, G., & Ahuja, I. S. (2014). An evaluation of just in time (JIT) implementation on manufacturing performance in Indian industry. *Journal of Asia Business Studies*, *8*(3), 278–294. https://doi.org/10.1108/JABS-09-2013-0051

Siqueira, F. L., Barbarán, G. M. C., & Becerra, J. L. R. (2008). A software factory for education in software engineering. *Software Engineering Education Conference, Proceedings*, (158), 215–222. https://doi.org/10.1109/CSEET.2008.10

Siy, H. P., Herbsleb, J. D., Mockus, A., Tucker, G. T., Krishnan, M., Labs, B., & Technologies, L. (2001). Making the Software Factory Work: Lessons from a Decade of Experience. *Proceedings of the 7th International Symposium on Software Metrics*, 317--. https://doi.org/10.1109/METRIC.2001.915539

Skok, W., Legge, M., & Hill, K. (2002). Evaluating enterprise resource planning (ERP) systems using an interpretive approach. *Knowledge and Process Management*, 189–197. https://doi.org/10.1002/kpm.138

Smith, J. (2006). Getting past knowledge management. *IEEE International Engineering Management Conference*, 66–71. https://doi.org/10.1109/UTEMC.2006.5236197

Smith, P. a. C. (2012). The importance of organizational learning for organizational sustainability. *The Learning Organization*, *19*(1), 4–10. https://doi.org/10.1108/09696471211199285

Sommerville, I. (2004). *Software Engineering (7th Edition)*.

Sridharan, B., & Kinshuk. (2003). Reusable active learning system for improving the knowledge retention and better knowledge management. *Proceedings - 3rd IEEE International Conference on Advanced Learning Technologies, ICALT 2003*, 72–75. https://doi.org/10.1109/ICALT.2003.1215029

Stoilov, T., & Stoilova, K. (2008). Functional analysis of enterprise resource planning systems. *CompSysTech'08: Proceedings of the 9th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, Gabrovo, 12-13 Jun 2008*, Art. 43. https://doi.org/10.1145/1500879.1500927

Stojanova, T., Suzic, N., & Orcik, A. (2012). Implementation of Mass Customization Tools in Small and Medium Enterprises. *International Journal of Industrial Engineering and Management*, *3*(4), 253–260.

Talebi, K., & Khaksar, S. M. S. (2011). The process of formation of manufacturing in small and medium-sized enterprises in Iran. *World Applied Sciences Journal*, *14*(1), 157–166. https://doi.org/10.5897/AJBM11.1363

Thoreson, S. A., Chief, S., Company, M. A., Corporation, M. D., & Louis, S. (1989). (asd) (irad), *4*(13), 1576–1580.

Timm Seitz. (2010). SAP ERP in the Cloud. *Oracle White Paper*, (April), 19–37. Retrieved from http://www.oracle.com/us/solutions/sap/database/sap-erp-cloud-352626.pdf

Torkashvan, M. (2009). A Service Oriented Framework for Cloud Computing, *2*.

Tracy, K. (2016). Cloud application sprawl in enterprise applications, 26–29.

Trochim, W., & Donnelly, J. P. (2006). *Research Methods Knowledge Base*. https://doi.org/10.1515/9783110858372.1

Truong, H. (2016). SINC – An Information-Centric Approach for End-to-End IoT Cloud Resource Provisioning. https://doi.org/10.1109/ICCCRI.2016.12

Tsai, W.-T., Shao, Q., Huang, Y., & Bai, X. (2010). Towards a scalable and robust multi-tenancy SaaS. *Proceedings of the Second Asia-Pacific Symposium on Internetware - Internetware '10*, 1–15. https://doi.org/10.1145/2020723.2020731

Tsai, W., Li, W., & Esmaeili, B. (2012). 2012 IEEE 4th International Conference on Cloud Computing Technology and Science Model-Driven Tenant Development for PaaS-Based SaaS, 821–826.

Tsai, W. T., Huang, Y., & Shao, Q. (2011). EasySaaS: A SaaS development framework. *Proceedings - 2011 IEEE International Conference on Service-Oriented Computing and Applications, SOCA 2011*. https://doi.org/10.1109/SOCA.2011.6166262

Tullis, T. (Thomas), & Albert, B. (William). (2008). *Measuring the user experience : collecting, analyzing, and presenting usability metrics*. Elsevier/Morgan Kaufmann.

Uflacker, M., & Busse, D. (2007). Complexity in Enterprise Applications vs. Simplicity in User Experience. *Human-Computer Interaction. HCI Applications and Services, Lecture Notes in Computer Science*, 778–787. Retrieved from http://www.springerlink.com/index/14U5G341H0256075.pdf

Uppström, E., Hoffsten, M., Uppstrom, E., Lonn, C. M., Hoffsten, M., & Thorstrom, J. (2015). New implications for customization of ERP systems. *Proceedings of the Annual Hawaii International Conference on System Sciences*, *2015–March*, 4220–4229. https://doi.org/10.1109/HICSS.2015.505

Vaishnavi, V., & Kuechler, B. (2004). Design Science Research in Information Systems Overview of Design Science Research. Retrieved from http://www.desrist.org/design-research-in-information-systems/

Valmohammadi, C. (2011). The impact of TQM implementation on the organizational performance of Iranian manufacturing SMEs. *The TQM Journal*, *23*(5), 496–509. https://doi.org/10.1108/17542731111157608

Vaniea, K. E., Rader, E., & Wash, R. (2014). Betrayed by updates. *Security and Human Behavior 2014*, 2671–2674. https://doi.org/10.1145/2556288.2557275

Vaniea, K., & Rashidi, Y. (2016). Tales of Software Updates : The process of updating software. *Chi*, 3215–3226. https://doi.org/10.1145/2858036.2858303

Vassiliadis, P., Bouzeghoub, M., & Quix, C. (1999). Towards quality-oriented data warehouse usage and evolution. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *1626*, 164–179. https://doi.org/10.1007/3-540-48738-7_13

Wade, M., & Hulland, J. (2004). Review : The Resource-Based View and Information Systems Research : Review, Extension, and Suggestions for Future Research. *MIS Quarterly Vol.*, *28*(1), 1–36.

Wang, R. Y., & Strong, D. M. (1996). Beyond Accuracy: What Data Quality Means to Data Consumers. *Journal of Management Information Systems*, *12*(4), 5–33. https://doi.org/10.1080/07421222.1996.11518099

Wazed, M. A., Ahmed, S., & Nukman, Y. (2010). Component and process commonalities in production system under various uncertain factors, *4*(17), 3697–3707.

Weber, F., Dauphin, E., Fuschini, R., Haarmann, J., Katzung, A., & Wunram, M. (2007). Expertise Transfer : A Case Study about Knowledge Retention at Airbus. *13th International Conference on Concurrent Enterprising (ICE 2007)*, (June), 329–338.

Wernerfelt, B. (1984). A Resource based view of the firm. *Strategic Management Journal*, *5*(2), 171–180. https://doi.org/10.1002/smj.4250050207

Whittaker, B. (1999). What went wrong? Unsuccessful information technology projects. *Information Management & Computer Security*, *7*(1), 23–30. https://doi.org/10.1108/09685229910255160

Wiegers, K. E. (2002). Seven truths about peer reviews. *Cutter IT Journal*, *15*(7), 31–37.

Wright, H. K. (2009). Release Engineering Processes, Models, and Metrics. *Proceedings of the Doctoral Symposium for ESEC/FSE on Doctoral Symposium*, 27–28. https://doi.org/10.1145/1595782.1595793

Wuest, T., Irgens, C., Thoben, K.-D., Wuest, T., & Thoben, K.-D. (2014). An approach to monitoring quality in manufacturing using supervised machine learning on product state data. *J Intell Manuf*, *25*, 1167–1180. https://doi.org/10.1007/s10845-013-0761-y

Xu, L. Da. (2011). Enterprise systems: State-of-the-art and future trends. *IEEE Transactions on Industrial Informatics*, *7*(4), 630–640. https://doi.org/10.1109/TII.2011.2167156

Zaidman, A. (2010). Multi-Tenant SaaS Applications : Maintenance Dream or Nightmare ? Position paper. *Iwpse-Evol '10*, 88–92. https://doi.org/10.1145/1862372.1862393

Zhu, X., & Wang, S. (2009). Software Customization Based on Model-Driven Architecture Over SaaS Platforms. *2009 International Conference on Management and Service Science*, 1–4. https://doi.org/10.1109/ICMSS.2009.5300967

# APPENDICES

## A - EXPERT REVIEW FORM

**VALIDATION FORM**

**Purpose of expert review:**

The objective of this document is to get assumptions and expert judgement with regards to a new proposed model entitled "**Cloud ERP Factory**". This is a new proposed model that can be used as a blueprint for Cloud ERP Production environment to reduce the complexity of managing Cloud ERP deployment, manage the workload of developers and improve software quality inconsistency with knowledge retention capabilities. The input and feedback from this exercise will be used to enhance and validate the proposed model.

**Expert/Reviewer Information**

Name : ______________________________________________

Age : 30 – 39 years [ ]   40 – 49 years [ ]   50 – 59 years [ ]

Gender : Male [ ]   Female [ ]

Affiliation : ______________________________________________

Working Experience : ________ years

Position: : ______________________________________________

**Items to Review**

Enclosed with this validation form is the proposed model entitled "Cloud ERP Factory" (Refer to Figure 1. In Page 3). Please validate based on the Likert Scale provided below:

1 – Strongly Disagree
2 – Disagree
3 – Somewhat disagree
4 – Undecided
5 – Somewhat agree
6 – Agree
7 – Strongly Agree

If the identified factors and elements is inappropriate or unclear, please comment, suggest, and edit accordingly. Your kind help and feedback is greatly appreciated.

Thank you.

**Dzulkafli Jalil**

1. **The CEF Model is easy to understand. (Understandability)** [ ]

2. **The CEF Model provided a clear steps and procedures to follow. (Clear steps)** [ ]

3. **The CEF Model is relevant to the software production environment of Cloud ERP. (Relevancy)** [ ]

4. **The CEF Model is able to support the needs for future Cloud ERP production environment. (Flexibility)** [ ]

5. **The CEF Model is able to be implemented based on the needs of an organization without having to add additional resources. (Scalability)** [ ]

6. **The data flow of the CEF Model is reliable and accurate. (Accuracy)** [ ]

7. **The CEF Model describes a complete Cloud ERP production lifecycle which can be applied by other Cloud ERP providers. (Completeness)** [ ]

8. **The CEF Model is able to act as a consistent information source to allow for decision making support. (Consistency)** [ ]

9. **The CEF Model is able to be applied in the future and in different context. (Timeless)** [ ]

10. Please provide any other comments or suggestions to improve the proposed model.

## B – VERIFICATION CHECKLIST

CEF Model Verification Checklist

| Component | No | Checklist Description | Tick Box |
|---|---|---|---|
| **CEF Product Line** | 1 | The product line grouping should cover intended product family and its sub product as well as future product can be fit into of the product family. | |
| | 2 | Important and common ERP modules should be defined and grouped as Core Module. | |
| | 3 | Model can be deployed to customer individually or in integrated mode (loosely coupled). | |
| | 4 | Core system, industry solutions and legacy can be integrated with standard interface protocols. | |
| | 5 | Future upgrades for one particular product should not affect the other system through backwards compatibility. | |
| | 6 | Each product has its own configuration parameter. | |
| **CEF Platform** | 7 | All product should fit in a common platform which has built-in product components. | |
| | 8 | All product information is systematically through a system. | |
| | 9 | Audit process and enforcement model are available. | |
| | 10 | The product line grouping should cover intended product family and its sub product as well as future product can be fit into of the product family. | |
| | 11 | It should be able to integrate with existing legacy product through API. | |
| | 12 | Model can be deployed to customer individually or in integrated mode (loosely coupled). | |
| | 13 | Future upgrades for one particular product should not affect the other system through backwards compatibility. | |
| | 14 | Each product has its own configuration parameter. | |
| **CEF Workflow** | 15 | The jobs can be across the department depending on the responsibility the task performed. | |
| | 16 | Each department need to have their own department specific job complete with checklists. | |
| | 17 | Each job can be used for other department's Work Template. | |
| | 18 | Upon confirmation of a Work Order, work order template will be used to distribute the jobs assigned to the designated staff within the system. | |

| | | | |
|---|---|---|---|
| | 19 | Each task should carry its own time frame to allow for tracking of progress and at which task. | |
| **CEF Product Control** | 20 | Part number generation must be done with the event of new product released. (Concurrent only with a sale of a new product). | |
| | 21 | Product database should contain all documentation which includes product specification, release notes, system algorithm and actual coding files. | |
| | 22 | Each product development owner is responsible for the release of the developed product. | |
| | 23 | Serial number activation is required to allow for keeping tabs on the number of products being sold and/or deployed to clients. | |
| **CEF Knowledge Management** | 24 | The knowledge creator has full power over the content of the knowledge provided. | |
| | 25 | The updates of contents, training materials, guides, and related documentation are conducted by the administrator, through the main knowledge management engine. | |
| | 26 | Able to ensure knowledge retrieval and logging through procedures in the system | |
| | 27 | The system ensures that all knowledge can be referred to from a single focal point i.e. dashboard. | |

# C - SURVEY INSTRUMENT

**CLOUD EFP FACTORY MODEL**

**FIELD DEMONSTRATION SURVEY FORM**

**UNIVERSITI UTARA MALAYSIA**

**NAME:**

**DESIGNATION:** **DATE:**

**CASE STUDY NO:**

*Please circle your answer in the scale of 1 to 7 according to the below questions.*

| No | PERCEIVED USEFULLNESS | | | | Scale Rating | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Using the CEF system environment in my job would enable me to accomplish tasks more quickly | Unlikely | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Likely |
| 2 | Using the CEF system environment would improve my job performance | Unlikely | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Likely |
| 3 | Using the CEF system environment in my job would increase my productivity | Unlikely | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Likely |
| 4 | Using the CEF system environment would enhance my effectiveness on the job | Unlikely | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Likely |
| 5 | Using the CEF system environment would make it easier to do my job | Unlikely | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Likely |
| 6 | I would find the CEF system environment useful in my job | Unlikely | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Likely |

| | PERCEIVED EASE OF USE | | | | Scale Rating | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 7 | Learning to operate the CEF system environment would be easy for me | Unlikely | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Likely |
| 8 | I would find it easy to get the CEF system environment to do what I want it to do | Unlikely | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Likely |
| 9 | My interaction with the CEF system environment would be clear and understandable | Unlikely | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Likely |
| 10 | I would find the CEF system environment to be flexible to interact with | Unlikely | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Likely |
| 11 | It would be easy for me to become skilful at using the CEF system environment | Unlikely | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Likely |
| 12 | I would find the CEF system environment easy to use | Unlikely | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Likely |

| No | ADDITIONAL QUESTIONS | | | | Scale Rating | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | The CEF system environment was able to reduce my daily workload compared to previous method | Strongly Disagree | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Strongly Agree |
| 2 | The CEF system environment was able to improve the overall management of workload in the company compared to the previous method | Strongly Disagree | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Strongly Agree |
| 3 | The CEF system environment was able to maintain the software quality being developed compared to previous method | Strongly Disagree | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Strongly Agree |
| 4 | The CEF system environment was able provides better step by step procedures to ensure consistency compared to previous method. | Strongly Disagree | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Strongly Agree |
| 5 | Managing Cloud ERP solution is easier compared to the previous method. | Strongly Disagree | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Strongly Agree |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 6 | The CEF system environment has provided me with easier management tools for my work compared to the previous method. | Strongly Disagree | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Strongly Agree |
| 7 | I am able to execute my task with the help of the CEF knowledge management system compared to previous method. | Strongly Disagree | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Strongly Agree |
| 8 | The CEF system environment was able to provide better knowledge retention compared to previous method. | Strongly Disagree | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Strongly Agree |

**GROUP:**

**NEGATIVE COMMENTS**

**POSITIVE COMMENTS**