SuccERP: The design science based integration of ECS and ERP in post-implementation stage

Yung-Yu Lin¹, Yukari Nagai¹, Tzu-Hang Chiang², and Hua-Ko Chiang¹

Abstract
The existing studies of Enterprise Resource Planning (ERP) are primarily focusing on the adoption and implementation stages, the post-implementation stage has received less attention in comparison. However, most enterprises have been struggling under the post-implementation stage. This work aims to review the state-of-the-art issues of ERP in the post-implementation stage, including communication, legacy system, collaboration, and the manager is hard to monitor the performance. Based on the Design Science (DS) method, we highlight how to offset the lack of an ERP system and ECS according to the guidelines of DS, and show the exhaustive steps for implementing the artifact—SuccERP. Our research is rigorous and interpretive by considering the steps of the DS and the functions of Software Engineering. Further, we explore multiple ERP systems to summarize the difference in authentication, initial data, and specific procedures aspects, after that, we consider the two most popular procedures (order creation and bill of purchase creation) as examples to demonstrate and evaluate the proposed artifact—SuccERP in the result. We propose the complete and practical research for solving the issues from previous theoretical results of an ERP, and to show experimentally that the proposed SuccERP is easy to maintain by applying the Cyclomatic Complexity and the Maintainability Index as metrics. This study is a milestone that allows ERP research to move from the theoretical stage to integrating, creating things that serve a human purpose, and dealing with the issues presented by previous works practically.

Keywords
Enterprise resource planning (ERP), enterprise collaboration systems (ECS), design science (DS), communication, post-implementation stage

Introduction
In today’s competitive business environment, an efficient and less expensive information management system is a key to maintain a competitive edge.¹ ERP system comprises multiple software modules to facilitate the flow of information in organizations, which improves the overall performance of manufacturing firms.² Further, prior literature indicates that strategic IT investment, such as ERP, can help sustain operational efficiencies in the long run.³ However, some scholars have highlighted the considerable disadvantages of ERP systems. Carton et al. (2003: 159) commented that “ERPs are good for storing, accessing and executing data used in daily transactions, but it is not good at providing information” and “Many [organizations]...
experience frustration when they attempt to use their ERP
to access information and knowledge. While we try to
improve these shortcomings, the task does not seem easy.
There are quite different characteristics from legacy sys-
tems to ERP systems, including complex, large-scale, inte-
grated, and interdependent. Accordingly, to an extent,
the difficulties have led to growing attention to the works
in the ERP field.

When it comes to the topic of ERP, most of the research-
ers will readily agree to divide the ERP project life cycle
into multiple implementation stages and focus on different
issues of each stage. On the one hand, Esteves and Bohor-
quex claim the life cycle can be classified into three imple-
mentation stages, including pre-implementation,
implementation, and post-implementation. On the other
hand, Swanson and Ramiller propose the life cycle com-
prises three stages: ERP adoption, implementation, and
post-implementation.

Our eyes were drawn to the post-implementation stage of
ERP system. Willis and Willis-Brown point out that once
an ERP system has been successfully built, it has a “go live”
date; however, implementation is not the end of the ERP
journey; the post-implementation or exploitation stage is
where the real challenges begin. Some studies have also
suggested that existing studies are primarily focusing on the
adoption and implementation stages. However, maintenance
and upgrading in the post-implementation have received less
attention in comparison with others. Esteves and Bohor-
quex further present the relevant statistics from the existing
research, with about 47% ERP implementation studies,
compared with only 15% of the studies on ERP post-
implementation. It is clear that research in the post-
implementation stage is necessary but lacking. To that end,
scholars have given different opinions on the issue of inade-
quacy of the post-implementation study.

Some research is to suggest that the post-implementation
study should toward exploring the Critical Success Factors
(CSFs). Hence, a portion of the study identified several
key factors critical to post-implementation, including support
from top management, user training, competency
of an internal ERP team, continuous process improvement,
collaboration, communication, continuous systems integra-
tion/extension. However, on the other hand, other
scholars have different views on exploring the CSFs. They
declare that the relevant literature remains limited to
explore the success factors and learning issues on the post-
implementation. More precisely, we should focus more
on improving performance to achieve real benefits as the
success of ERP post-implementation rather than the current
literature, which focuses on software selection and imple-
menting processes and CSFs.

Although ERP replaces some legacy systems (e.g.,
accounting, billing, order entry, etc.), it is challenging to
integrate ERP systems seamlessly with other resources. Under
highly dynamic requirements from the user and a
competitive market, organizations need to continuously
enhance and upgrade the ERP systems in the post-
implementation stage. It will help the organizations
differentiate themselves from the competitors. More spec-
ifically, an organization should advance the ERP system’s
information and communication and improve the business
processes to increase the informational effects. However,
about the issues on the management side, it suffers from a
lack of in-house experts and insufficient support from sys-
tem vendors and consultants; on the practical side, it
encounters technical pitfalls and deficient system design.
Furthermore, the previous CSFs research overlooks the
contextual factors of post-implementation maintenance,
knowledge management, and enhancements.

Motivated by these issues and suggestions, we attempt
to extract information from an ERP system and integrate it
with the Enterprise Collaboration Systems (ECS). The
integration aims to improve collaboration and communica-
tion and allow the managers to realize the effective way
to monitor and improve employees’ efficiency by sharing
and synchronizing ERP and ECS information. To that end,
this paper presents the artifact—SuccERP. Build upon the
Design Science (DS) methodology, we show the complete
development process and provide noteworthy schema and
rules that an organization should be aware of when con-
sidering enhancing/upgrading an ERP system in the post-
implementation stage by exploring multiple ERP systems.

More precisely, instead of operating in the original ERP
system, the user can carry out the procedures of the ERP in
the ECS directly based on the support of SuccERP, such as
the inventory inquiries, order processing, and bill process-
ing. We aim to overcome the issues that organizations
encounter during the post-implementation stage, such as
communication, collaboration, information sharing, and
technical pitfalls, by providing a complete development
process to implement ERP and ECS integration. As sug-
gested by Kwon and Lee, the development of SuccERP
will be along with the pseudo-code to present our pro-
posed system’s details. In short, our demonstration also
includes the pseudo-code to provide detailed steps and
functions.

The remainder of this paper is organized as follows. In
Section 2, we provide a literature review on ECS, ERP, and
DS. Section 3 presents how we design and develop our
proposed SuccERP system. In Section 4, we demonstrate
the results and evaluate the proposed artifact. In Section 5,
we discuss the results by answering two fundamental ques-
tions while considering DS to implement our artifact and
present the implications, contributions, limitations, and
future works.

Literature review

In this section, first, we review the ECS, since the Suc-
cERP implements the integration between ERP and ECS. Later, a brief overview of the ERP system mostly focuses
on the post-implementation stage. In the end, we review
the DS to interpret how we apply it to achieve the design, implementation, and evaluation of the proposed artifact—SuccERP.

**Enterprise collaboration systems (ECS)**

From the description of the 8C Model for Enterprise Information Management, we can explicitly get the ECS outline, where employees can achieve collaborative works such as information and content sharing, communication, cooperation, and coordination through all areas of collaboration covered by ECS. On the whole, ECS is a software system that supports the collaborative work of employees.\(^{29}\)

Previous studies think highly of ECS, as Schubert and Glitsch\(^{30}\) suggest that ECS is a crucial enabler of the modern digital workplace. It is a suitable tool for longitudinal work according to the identification of cases and scenarios. Using ECS, from the manager to the employees, the members of an organization can work effectively. However, research using ECS for further exploration is relatively fragmented and rarely provides in-depth empirical studies. Further, ECS challenges are multifaceted; they often require to be addressed in different ways.\(^{31}\)

Greeven and Williams\(^{32}\) identify the challenges and issues while organizations encounter during the introduction and use of an ECS. They propose five adoption challenge areas, including culture, business/operation, technology in use, benefits, and attitude/behavior according to academic literature and interviews. They are especially studying the lack of activity in the period of ECS adoption that ineffective content is the primary reason. While ineffective content contributes to this problem primarily, inferior quality and not enough usable content is also an essential factor that they reminded. As a result, it leads the employees to move to other alternatives (e.g., email, previous generations of groupware software.).

Diehl et al.\(^{33}\) suggest that the laisseez-faire approach cannot get the full potential to project success. Besides avoiding the laissez-faire approach, ECS is social software that social software presents cultural rather than technical challenges. These cultural challenges predictable and should be managed beforehand, not ad hoc. Most importantly, the adoption of trading systems such as ERP is almost mandatory, and the use of ECS is usually voluntary.

From the above literature reviews, we highlight the short conclusion below:

1. Although ECS effectively supports collaborative work and a suitable tool for longitudinal work, the ECS content determines the user’s attitude, either animated in using ECS or moving to alternatives. Hence, how to make the ECS content more useful is an important issue.
2. In the attitude’s discussion of the users between ERP and ECS, ERP is almost mandatory, and ECS is usually voluntary. These outcomes open the door to studies that the possibility of combination or integration with ERP and ECS.

**Enterprise resource planning (ERP)**

Research on the integration of ERP and external resources is gaining more and more attention.\(^{34-38}\) Ruivo et al.\(^{37}\) stated that ERP systems support day-to-day business operations; however, it is worth considering that integrated ERP and CRM can communicate with customers and deliver process-related information to each other. The study discusses whether ERP and CRM systems’ integration significantly impacts the systems and process integration within an organization. From the findings that system integration has a positive and significant impact on business value. They conclude there might be other resources can contribute to the business value that are worth exploring, and both IT and functional managers need to take into consideration processes integration between systems.

Ha and Ahn\(^{19}\) identify six critical factors that influence ERP performance in the post-implementation stage, including top management support, competency of an ERP team, user training, inter-department collaboration and communication, continuous process improvement, and continuous systems integration/extension. They also emphasize that continuous improvement in the post-implementation stage has a positive impact on ERP performance.

We confirm that ERP integration is a worthwhile research direction from the review mentioned above; further, continuous process improvement and continuous systems integration/extension are the CSFs in the post-implementation stage. Although the increasing studies involved in the post-implementation stage, most studies are still limited to explore CSFs. The following studies explicitly point out the issues in the actual world. Ali and Miller\(^{14}\) show critical gaps in current research that only a few works concentrate on the post-implementation stage. Besides, there is no standard methodology devised, and the cost is also a significant limitation. Willis and Willis-Brown\(^{9}\) argue that once the ERP system is set up, it has a “go-live” date. However, this is not the end; instead, the post-implementation stage’s development process is the big challenge. Such proposals are analogous to other studies. Peng and Nunes\(^{26}\) figure out the organizations often encounter various risks such as technical pitfalls, emergent business needs, inadequate user behavior, and deficient system design when developing, maintaining, and enhancing the ERP system.

In the discussion of the literature review on post-implementation, one controversial issue has been what kind of studies should give more emphasis. On the one hand, Osses et al.\(^{16}\) argues a need for more research on CSFs in the ERP post-implementation stage. On the other hand, Amid et al.\(^{39}\) contends current research has focused on software selection, implementing process, and CSFs rather than...
the success, actual benefits, and performance improvements of the ERP post-implementation stage. Others even directly claim the relevant literature still limited to explore CSFs and learning issues in the post-implementation stage.17

Our view aims to improve ERP performance by data integration and the information sharing between the organization’s systems and business processes and proposing complete development steps of the artifact. Hence, before that, we make a brief conclusion to extract the keywords of the suggestion from previous studies in the following Table 1.

**Table 1. Arranged suggestion from previous research in post-implementation stage of ERP.**

| Keywords/Source | Short conclusion |
|-----------------|------------------|
| Internal hosting. | From this report, the ratio of hosting options between the external and internal is 22.6% and 77.4%. The reasons why organizations do not select cloud ERP, including (i) Risk of security branch (27.27%). (ii) Risk of data loss (31.82%). (iii) Lack of information/ knowledge about offerings (40.91%). Another point is that the more employees use cloud ERP, the more licenses must purchase. |
| Functional upgrade, Enhancement. | There are two functional roles of an ERP in organizations: automating and informing. Automating is considered as an essential function and has been thoroughly used to date. Informating role is defined as all the information generated during the work process using an ERP system and translating the description of activities, events, and objects into data to support work integration and decision support. The potential of an ERP depends on the informing role and data integration throughout the enterprise. |
| Information sharing, Communication, Monitoring. | To solve unexpected projects, we need to develop an ERP system for demands continuously. Meanwhile, both the continuous system development and ongoing business process monitoring were identified as the ultimate success factor. Part of these developments involves the achievement of inter-department information sharing because we consider poor communication, which is a reason for the post-implementation stage failure. Besides, ERP systems are complex, and the extension of such complex systems often necessary to rely on external IS services. Therefore, managers should mindfully select metrics and information to monitor employees’ use of the ERP system to manage the ERP system and external services. Moreover, keeping the metrics and information accurate, updated, consistent, relevant, complete, and format is easy to understand. |

**Design science (DS)**

As stated above, most ERP researchers have focused on the subject of study in behavioral science research. We consider the theory-oriented cumulative knowledge base as a set of statements that summarized the critical factors and lessons learned from ERP research. The theoretical basis is the logical concern and product of both design and behavioral science scholars. Aboulafia47 specifies the relationship between theory and artifact. The truth (justified theory) and utility (artifacts that are effective) are two sides of the one coin, and the scientific research should consider its practical implications for evaluation. Lee48 elaborates that technology and behavior are not dichotomous in an information system; they are inseparable. To some degree, it supports the perspective that the theory and artifacts are equally important. Simon49 advocated the natural sciences and social sciences attempt to understand reality. Design science attempts to create things that serve a human purpose. The DS paradigm has its roots in engineering and the sciences of the artificial.

Hevner et al.50 describe the DS research in information systems by proposed a conceptual framework that comprises seven guidelines: Design as an Artifact, Problem Relevance, Design and Evaluation, Research Contribution, Research Rigor, Design as a Search Process and Communication of Research. They also summarized two fundamental questions for DS research: “What utility does the new artifact provide?” and “What demonstrates that utility?”.

Peffers et al. 51 propose and develop a design science research methodology (DSRM) for implementing the Information System. There are six activities in the DSRM include: Problem identification and motivation, Define the objectives for a solution, Design and development, Demonstration, Evaluation, and Communication. Also, there are multiple possible entry points for DS research, and they show several cases start from different activities to carry out.

In the next section, we will refer to the six activities of the DSRM51 and seven guidelines.50

**The design and development of proposed SuccERP system**

Davenport52 explains that an ERP system’s heart is a central database used to draw the data and feed them into a series of applications to support the organization. The flow of information could flow throughout a business in an organization based on a single database. For providing more substantive guidelines of how we built our artifact based...
on the activities and guidelines to integrate with an ERP database, this section begins with the problem identification and motivation to the evaluation and demonstration. Also, some activities and guidelines that will moderately incorporate software engineering.

**Problem identification and motivation**

Software engineering is a widely used discipline, especially in the improvement and development of information systems. The purpose of the function of the software specification and the problem identification and motivation of DSRM is similar; therefore, we apply the user requirements definition and system requirements specification to systematically achieve this work, which is present in Table 2 and Table 3.

The items of Table 2 mainly refer to the section of the literature review where we outline the items one-by-one below. For item 1, we first consider the ratio of hosting options between the external and internal is 22.6% and 77.4% from the ERP report.\textsuperscript{40} Hence, the post-implementation stage studies are impractical if without considering the issues of the ratio of internal hosting still high. Further, although Liu and Liu\textsuperscript{33} identified the necessity for integrating an ERP system with other external resources, we have no choice but to face a growing challenge for developing and enhancing an ERP system. When integrating an ERP system with other external resources, applying to both internal and external hosting is a more practical and complete solution.

Next, we consider both item 2 and item 3 at the same time. Most organizations are struggling to share information effectively from an ERP for communication. We found ECS and ERP can support the weak points of each other from the previous studies below. Oseni et al.\textsuperscript{25} show that for attaining ERP-based operational effectiveness, it needs to rely on the present advances in information and communication ability. Al-Mashari\textsuperscript{45} presents to considering the monitoring of an ongoing business process in an ERP is a significant stage for completing a full process of ERP implementation. Meanwhile, Hsu et al.\textsuperscript{13} also suggest that managers should consider meaningful metrics from the ERP system to monitor and test the performance from employees’ use of an ERP system. As a result, selecting effective information for communication and monitoring is a critical priority in the post-implementation stage.

Although the ECS can achieve collaborative works such as communication, longitudinal work, and monitoring, the ineffective content is the primary reason for the lack of activity in ECS adoption.\textsuperscript{30–32} For item 4, Weston Jr\textsuperscript{54,55} present that the cost matters in determining whether the SMEs should carry out upgrade/enhancement or not. Peng and Nunes\textsuperscript{26} further figure out that organizations often encounter various risks of techniques when developing or enhancing an ERP system in the post-implementation stage. Hence, for small and medium enterprises (SMEs), both enhancements and functional upgrades require development costs and extra license costs with the increase in the number of users. Therefore, an explicit guideline of upgrades and enhancements can prevent encountering the risks of techniques; also, the cost of self-exploring and consultant’s charges can significantly decline.

After proposing the user requirements definition, we summarize the system requirements specification to show how we define the proposed artifact’s functions in Table 3.

**Define the objectives for a solution**

This research aims to create integration with external resources to strengthen the communication and monitoring of an ERP. We select ECS as the external resource and

### Table 2. User requirements definition.

| Item | Descriptions |
|------|--------------|
| 1.   | How can we enable an ERP system to integrate with external resources regardless of internal or external hosting? |
| 2.   | How can we provide an efficient way for managers to monitor their employees’ performance in operating an ERP system? |
| 3.   | How can we make content in the ECS system more effective? |
| 4.   | For SMEs, how can they strike a balance between the enhancements and costs? |

### Table 3. System requirements specification.

| Item | Descriptions and specifications |
|------|---------------------------------|
| 1.   | Regardless of internal or external hosting, the artifact needs able to access an ERP database. |
| 2.   | The artifact needs to collect the database schema and information of tables from the corresponding procedure of an ERP system. |
| 3.   | The artifact needs to provide the user interface in the artifact to allow the users to carry out the ERP procedures and deliver the data to external resources that allow the managers to monitor and evaluate employees’ performance by the external resources. |
| 4.   | The artifact needs to synchronize the data between the external service and an ERP system to let all the members get information timely, whether the members access external service or an ERP system. |
interpret the entire development process for creating the integration between ECS and ERP system by the proposed artifact. The reason to select ECS as describe below.

1. ECS is a suitable tool for longitudinal work\textsuperscript{30}, such a characteristic is ideal when an organization manager needs to monitor employees’ performance.

2. The adoption of an ERP system is almost mandatory, and the use of ECS is usually voluntary.\textsuperscript{33} To our knowledge, with voluntary, the resulting content is relatively casual, that it might be the primary reason an ECS with the issues of lower quality and not enough usable content.\textsuperscript{32} However, ERP data sources are mandatory and valid that can improve these shortcomings of an ECS.

We illustrated the architecture diagram in Figure 1. There are four kinds of roles in the architecture diagram, including the ECS external user, ERP external user, ERP internal user, and developer. It is worth mentioning that there are two primary actions in this architecture, and we used the arrow with a distinct style for recognizing.

The solid lines in the blue color show an action that grasps the database schema and corresponding table info then inserts into the SuccERP database for achieving the actions representing with the dotted lines in green color. We describe it step by step below.\textit{Step 1 and Step 2}: The developer initializes the Reverse Engineering Service (RES) by invoking the SuccERP Application Programmers Interface (API).\textit{Step 3}: The RES ready to grasp and get the schema and corresponding tables info with the Database Profiler after initialized. We applied the Database Profiler to grasp the information and rules from the Database and Application server after the ERP user carried out the ERP procedures in \textit{Step 4}.\textit{Step 4}: The client user can carry out the specific procedure of an ERP based on the external or internal hosting.\textit{Step 5 and Step 6}: After receiving the requests of procedures from the ERP users, there is a series of data processing between the ERP database and server.\textit{Step 7}: Then, RES grasps and receives those necessary schema and tables info.\textit{Step 8}: The RES insert the schema and tables info of the ERP database into the SuccERP database.\textit{Step 9 and Step 10}: The RES returns the confirmation message to the SuccERP API and the developer.

Switch to the part of dotted lines in the green color of Figure 1, where we describe how the ECS users with the proposed artifact-SuccERP to carry out the ERP procedures directly.\textit{Step 1 and Step 2}: The ECS users will deliver the request for executing the specific ERP procedures to the SuccERP API via the ECS API.\textit{Step 3 and Step 4}: While the SuccERP API receives the request from the ECS API, then the SuccERP API will request the database schema and tables info from the SuccERP database to execute the corresponding procedures in the ERP database.\textit{Step 5}: After completing the data processing from the SuccERP to an ERP database, sending a confirmation message to the

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{architecture.png}
\caption{The architecture of proposed artifact-SuccERP.}
\end{figure}
ECS API. Step 6: Create a corresponding message in the ECS system for further communication and the example, as shown in Figure 12. Step 7: Upload the relevant report files to the stored files server. Step 8: Return the confirmation message to the ECS users.

The part of blue/solid lines and green/dotted lines are two independent actions. Typically, the blue part executed at the beginning only, and we describe the subsequent operations from the ECS users as the green part, such as order creation, inventory inquiries, and bill of purchase.

**Design and development: The solution for the hosting issues**

Regarding the keywords of Internal hosting from Table 1, and the item 1 of Table 3 for defining the system requirements, we apply Figure 2 to interpret the productive solution. First, the limitation on the ERP for internal hosting is without a physical address for identification. In other words, SuccERP API must be able to move to the ERP local environment and hold the ability to connect with the ECS API.

With internal hosting, users are only allowed to access the ERP with a dynamic address. As mentioned earlier, the action constructed by the solid lines with a blue color in Figure 1 only execute at the beginning to recognize the database schema and table info of the ERP database. Hence, in Figure 2 we only describe the dotted lines’ action with green color. The servers and services relevant to the blue lines turn to the semi-transparent color in the diagram; they have no connection with this solution.

Here we describe the steps that differ from Figure 1, and the users will access ECS and SuccERP API via the web browser as a local client. Step 1: This step is divided into two parts, especially, Step 1.B which is used to request the initial data from a client user of ECS for keeping those data in the local environment. Step 2 to Step 4 are used to describing the requests from the ECS user to carry out the ERP specific procedure in ECS. Step 5 to Step 7 are used to creating the corresponding message in the ECS after the data-processing in the ERP database completed. In the end, Step 8 will send a confirmation message to the ECS user after completing the ECS message creation.

In summary, the critical point is that the SuccERP API can change the role according to the environment. The SuccERP API’s role in Figure 1 is as a bridge that ECS API delivers the data to the ERP database by the SuccERP API. On the other hand, under the local environment’s limitation, it can treat as the only pathway for communicating with external services, as we describe in Figure 2.
**Design and development: Collect the outline and schema of ERP database**

The Reverse Engineering System (RES) is an initial point under the SuccERP architecture; it is the heart of the action, constructed by blue lines in Figure 1. By invoking the Database Profiler, the RES is used to request and grasp the statements executed in the ERP database that from the ERP users login into the ERP system to carry out the specific procedures.

We make an illustration in Figure 3, which comprise two kinds of swim lane for the comparison. The bottom part of Figure 3 is the general ERP process without SuccERP. On the contrary, the upper part is in the case with the SuccERP. The most crucial part is verifying the collected schema and tables to select the relative data from the ERP database according to the two decision processes in the swim lane. We consider multiple ERP systems to verify whether the RES is working and mention those crucial parts below.

1. Before the user logs into an ERP system, the developer initializes the Database Profiler to wait to get the statements and corresponding authentication schema by invoking SuccERP API. After that, collect the authentication schema and relevant table info into the SuccERP database during user login into an ERP system. The schema and table info is the basis for subsequent integration between SuccERP API and ERP system. There are two common scenarios of authentication schema.

   We describe the first scenario in Figure 4; it collects the authentication data in an independent database Auth*. The login data of each user preserve on a table of the database Auth*. The point is, they summarize all the authentication data in a database and determine which companies that a user allows accessing according to the specific column (i.e., MA003) of the table dbo.Table N. Also, each company has an independent database. The second scenario is described in Figure 5. It conserves the authentication data at the pre-defined table (Table P) in each independent company database.

2. Then, we make a brief conclusion of the initialize data schema in Table 4. We can categorize the initial data into 10 categories, including user department, company info, warehouse, items type, items collection,
Figure 4. The schema of authentication in an ERP system.

Figure 5. The schema of authentication in an ERP system.
sheet rule, invoice, currency, customer & supplier, and tax. To our knowledge, most ERP systems are following this category to initialize the necessary data. Therefore, we summarize it to let enterprises have these rules in hand for upgrading or enhancing the post-implementation stage.

3. Finally, we considered the four most frequently used procedures, including order creation, bill of purchase creation, bill of sale creation, and inventory inquiries regarding the execution for the specific procedures. Equally, we arrange the relative tables associated with these four procedures in Table 5.

Based on RES, the researcher and developer can get the scope of an ERP database. Especially in the authentication part, there is a significant difference between different ERP systems. Next, we describe how ECS integrates with ERP while carried out ERP procedures.

**Table 4. The relative tables and descriptions in initialize data step.**

| Category         | Short descriptions                                                                 |
|------------------|------------------------------------------------------------------------------------|
| User department  | An ERP system goes through the department of a user to govern the permission.     |
| Company info     | The company info typically includes the company name, unique identifier code of the company, telephone, fax, address, and the unique identifier code of administrator columns. |
| Warehouse        | There are two dependent tables relevant to the warehouse. One is the collection of warehouse includes a unique identifier code of warehouse and name. The other is used to collect the inventory of each item in a warehouse. While we try to carry out the procedure of inventory inquiries, we must identify how many warehouses in a company first confirm the inventory of each warehouse. |
| Items type       | The items typically comprise finished inventory, semi-finished products, and defective products, etc. Besides, a single warehouse might comprise multiple types of items. |
| Items collection | The items collection table typically includes a unique identifier code, name, specification, items unit, items type, and unit price columns. In some cases, they summarize all the inventory from each warehouse and save it as on-hand inventory in this table. |
| Sheet rule       | This table is used to indicate how to generate the unique identifier code of each sheet. |
| Invoice          | This table is used to preserve invoice records of purchases and sales, including invoice type, tax rate structure, and tax name columns. |
| Currency         | Each company might have multiple options on the currency. This table typically includes the currency name, rate, and a unique identifier code column. |
| Customer, Supplier| These two data are similar; the customer and supplier data are preserved in a single table in many cases. It typically comprises a unique identifier code, name, liaison, tax ID number, telephone number, fax number, and address columns. |
| Tax              | This table collects a list of tax types, and it comprises the value of tax rate, name of the tax, and a unique identifier code column. |

**Table 5. The relative tables and descriptions in the step of four specific ERP procedures.**

| Category           | Short description                                                                 |
|--------------------|------------------------------------------------------------------------------------|
| Header of an order, Header of a bill | Each sheet with a single header-only and the important columns include order serial number, date, identifier code of customer or supplier, currency, delivery address, tax type, tax fee, sum total, and sum quantity. |
| Content of an order, Content of a bill | At least one record per sheet in the content and the important columns include order serial number, index number, relevant data of items, relevant data of warehouse, quantity, unit, unit price, and subtotal. |
| Log data of an order, Log data of a bill | While the user inserts or updates a sheet, there is a record insert into this table. Typically, the important columns are similar to the tables of header category as described above. There are two kinds of the scenario for recording the inventory of each item. One only records the inventory of items in each warehouse. The other is beside the inventory of items in each warehouse and record all the inventory of each item on-hand in another table. |

**Design and development: The integration between ECS and ERP**

Let us briefly look back on why we want to integrate ECS and ERP from the conclusion and suggestion of previous works. First, the ERP system contains much information effective for management. However, collaborate and communicate with the ERP system is partly ineffective. While ECS has the advantages of facilitating communication and collaboration, it usually contains ineffective and useless content. We believe that ECS and ERP offset the lack of each other and settle ERP’s serious issues in the post-implementation stage.
Figure 6. Use case diagram and the illustration of integration between ECS and ERP.
We present the detailed integration between ECS and ERP system via proposed artifact in Figure 6, which is the more interpretive version of Figure 1, especially in terms of processes and data flow. From the previous work, Lin et al. 56 presented SuccMail (https://succmail.com) as an ECS to achieve the research purpose of integrating ECS and ERP system as an initial exploration. Regarding the use of ECS, we can discover some interesting features. The usage of an ECS is hard to prescribe, and it is challenging to develop manuals or guidelines for its utilization. From the theory on the social construction of technology (SCOT), we can explain that the evidence of the ECS is open to interpretative flexibility.57 The practical cases define part of the features. For example, the software IBM Connections provides users with multiple options for creating templates of activities with a list of tasks. Creating such a template is simple, but the areas of use are endless. Equally, during our research, we found evidence for such a task by integration with the ERP system and SuccMail. We identify these areas for the use case to implement the specific purposes in an enterprise. As illustrated in Figure 6, the ECS software SuccMail provides managers and employees with the possibility to communicate/collaborate based on lists of enterprises they took part in.

With the SuccMail, the managers will pre-define several discussion groups that give an initial name to reveal the group’s expected topics. Each discussion group with different task orientations comprises members according to what they are in charge of in an enterprise or organization. Each member can assign the receiver while they create a message within a discussion. Each message comprises a subject (title and content) and multiple replies similar to the structure of an email. With the support from the SuccERP, the relevant data of the specific ERP procedures will fully synchronize to a message of a discussion in the SuccMail. For communication, it allows the managers and employees can create replies to do further discussion and collaboration. Considering the suggestions from Kwon and Lee,28 we present the pseudo-code (Algorithm 1 and 2) below and the functions of each table of the ERP database.

The scenario of Algorithm 1 is while a user accesses a created message in the SuccMail, the ECS API will send multiple requests to the SuccERP API to determine whether it needs to update the message. Although we conduct the SuccMail as the example of the ECS system in our research, the Algorithm 1 we present is not limited to specific systems and programming languages. It begins at the user accesses a created message; the ECS API will apply the function $\text{CONN\_SERVER}$ to confirm whether it can create a connection with the ERP database and ensure the ECS user has permission to access the company of an ERP. It is worth mentioning that considering internal hosting, we apply the function $\text{GENERATE\_HOST\_URL}$ to determine the way of connection with the ERP database. Next, the ECS API delivers an ECS user’s data as a parameter to the function $\text{USER\_BINDING}$ for requesting the user binding, typically depends on the email address. Each member can assign the receiver while they create a message within a discussion. Each message comprises a subject (title and content) and multiple replies similar to the structure of an email. With the support from the SuccERP, the relevant data of the specific ERP procedures will fully synchronize to a message of a discussion in the SuccMail. For communication, it allows the managers and employees can create replies to do further discussion and collaboration. Considering the suggestions from Kwon and Lee,28 we present the pseudo-code (Algorithm 1 and 2) below and the functions of each table of the ERP database.

The scenario of Algorithm 1 is while a user accesses a created message in the SuccMail, the ECS API will send multiple requests to the SuccERP API to determine whether it needs to update the message. Although we conduct the SuccMail as the example of the ECS system in our research, the Algorithm 1 we present is not limited to specific systems and programming languages. It begins at the user accesses a created message; the ECS API will apply the function $\text{CONN\_SERVER}$ to confirm whether it can create a connection with the ERP database and ensure the ECS user has permission to access the company of an ERP. It is worth mentioning that considering internal hosting, we apply the function $\text{GENERATE\_HOST\_URL}$ to determine the way of connection with the ERP database. Next, the ECS API delivers an ECS user’s data as a parameter to the function $\text{USER\_BINDING}$ for requesting the user binding, typically depends on the email address. Each

```plaintext
Algorithm 1. Synchronize data while accessing a message in ECS.

procedure $\text{Synchronize\_data\_with\_ERP\_via\_SuccERP}(\text{sheet})$
    $C \leftarrow \{\text{compId}, \text{user_id}, \text{ip\_address}, \text{db\_name}, \text{user\_name}, \text{password}, \text{host\_type}\}$
    $D \leftarrow \{\text{order\_id}, \text{bill\_id}, \text{flag}, \text{message\_id}\}$
    $\text{target\_Host} \leftarrow \text{GENERATE\_HOST\_URL}(\text{host\_type})$
    $\text{connRes} \leftarrow \text{CONN\_SERVER}(C, \text{target\_Host})$
    if $\text{connRes.IsSuccess} == \text{FALSE}$ then
        return $\text{FAIL}$;
    end if
    $\text{bindingUserRes} \leftarrow \text{USER\_BINDING}(\text{user\_id}, \text{user\_email}, \text{target\_Host})$
    if $\text{bindingUserRes.IsSuccess} == \text{FALSE}$ then
        return $\text{FAIL}$;
    end if
    $\text{bindingCompRes} \leftarrow \text{COMP\_BINDING}(\text{comp\_id}, \text{target\_Host})$
    if $\text{bindingCompRes.IsSuccess} == \text{FALSE}$ then
        return $\text{FAIL}$;
    end if
    $\text{sheet\_data} \leftarrow \text{GET\_SHEET\_DATA}(D, \text{target\_Host})$
    if $\text{sheet} == \text{ORDER}$ then
        $\text{UPDATE\_MESSAGE\_DATA}(\text{sheet\_data})$
    end if
    if $\text{sheet} == \text{BILL}$ then
        $\text{UPDATE\_MSG\_BINDING\_INFO}(\text{order\_id}, \text{target\_Host})$
    end if
end procedure
```

Each member can assign the receiver while they create a message within a discussion. Each message comprises a subject (title and content) and multiple replies similar to the structure of an email. With the support from the SuccERP, the relevant data of the specific ERP procedures will fully synchronize to a message of a discussion in the SuccMail. For communication, it allows the managers and employees can create replies to do further discussion and collaboration. Considering the suggestions from Kwon and Lee,28 we present the pseudo-code (Algorithm 1 and 2) below and the functions of each table of the ERP database.
created message in the ECS will record the unique, identical
code of ERP company, which is the parameter for applying
the function \texttt{COMP\_BINDING()} for identifying which ERP
company is the target and accessing it for following
requirements. Then, according to the result of the request
from the function \texttt{GET\_SHEET\_DATA()} to update ECS’s
Corresponding message and the binding info. Finally, each
sheet (bill or order) report will remove and generate a new
one by the function \texttt{REFRESH\_REPORT\_DATA()} if the
sheet has changed. The format of the report is in PDF file.

Compared with Algorithm 1 that the Algorithm 2 is
more complex, hence, we will specifically explain the func-
tion with corresponding categories of table as we summar-
ized in Table 4 and Table 5. In addition, our description
will begin from function \texttt{SELECT\_DISC()} because we skip
the duplicate parts in the Algorithm 1. Algorithm 2 mainly
describes sheet creation, including order creation, bill of
purchase, and bill of sales creation.

+ function \texttt{SELECT\_DISC()} is used to provide a list of
discussion groups for the ECS user, and the user
assigns a discussion group to the creating message
associated with the ERP sheet in the ECS.
+ function \texttt{INITIALIZE()} is used to read all the neces-
sary initial data from the ERP database via the API of
SucCERP. We summarize these data into categories,
as shown in Table 4 and we discuss the critical differ-
dences from various ERP systems while initial the data.
  + \textit{Company info:} It is used to show the information
    of current ERP company connected. In our case,
    reading the information of ERP company is not
    only at a particular point in this time (initializing)
    but also beginning to generate the report data
    after the sheet creation. Other users might modify
    the ERP system between invoking the
    \texttt{INITIALIZE()} function until submitting the sheet
    creation request. Therefore, our artifact will send
    the request to get the company info again to
    ensure data consistency between the ERP system
    and ECS.
  + \textit{Warehouse:} Regardless of order, bill of purchase,
or bill of sales creation, the content items must
    associate with a warehouse. Hence, a sheet might
    relate to multiple warehouses. While we do the
    inventory, inquiries, or adjustments need to refer
to the warehouse data. As describe in Table 5,
    there are two kinds of scenarios. While an ERP
    system is with recording the all inventory of each
    item on-hand in another table, that the ECS API
    will sum the inventory from each warehouse and
    update the corresponding record in that table.
  + \textit{Itemtype:} The items type contain finished inven-
tory, semi-finished product, and defective prod-
uct, Etc. A single warehouse might comprise
multiple types of items.
  + \textit{Sheerule:} These are the rules of an identical code
    of an order or bill. Generally, the code comprises
    4-digits for the year, 2-digits for the month, 2-
digits for the date, and 4-digits indicating how
    many orders or bills create in a day. In most ERP
    systems, they record their rule on a table. Some
defined the rules in the ERP system instead of the
    table. In that case, the developer needs to check
    the rule manually.
  + \textit{Customer,Supplier:} The object of an order is a
customer, and the object of a bill is a supplier. In
some ERP systems, they used different tables to
collect the customer and supplier data separately.
However, others apply an additional column as a
flag to identify the collected data from both the
customer and the supplier within a single table.
Hence, we need to deliver one more parameter
while an ERP system collects both customer and
supplier into a single table.
+ function \texttt{GENERATE\_SHEET\_CODE()} is applied to
    generate the latest unique identifier code according
to the sheet rule and the number of sheets created on
that day. According to the sheet type (order or bill),
consider the corresponding table to estimate the
number of the created sheet on that day.
+ function \texttt{SELECT\_OBJECT()} according to the sheet
type (order or bill) to provide the customer or sup-
plier list options. While the user selects an object that
the artifact will load the relevant data, including
address, phone number, and liaison, Etc.
+ function \texttt{SELECT\_TAX()} most ERP systems provide
several taxable, exemption, and untaxed options.
These tax options need to take into account while
calculating the total of a sheet.
+ function \texttt{INIT\_EMPTY\_SHEET\_ITEM()} is used to generate
a new item in the sheet’s content after initial-
izing the data in our artifact’s user interface, which
allows the user to add a new item in the sheet’s
content by themselves.
+ both functions \texttt{GENERATE\_TITLE()} and
\texttt{GENERATE\_CONTENT()} are used to generate the
message of ECS, and the generated message looks
like the example we illustrate in the right-bottom cor-
er of Figure 6. We organize the sheet from the ERP
database into completed message information via
these two functions. These two functions provide
more possibilities to the managers and developers in
the post-implementation stage, such as transferring
the ERP data into other languages, building evaluation up
with more metrics/indicators, or keeping the prompt-
ing messages for further communication ECS.
+ function \texttt{CALCULATE\_TAX()} is according to the
user’s selected tax option to calculate the total and the
Value-added Tax (VAT). The tax rate value is a record
in a table as the category tax we describe in Table 4.
Algorithm 2. Sheet creation.

```plaintext
1: procedure CREATE_SHEET()
2:     \( C = \{ \text{comp.id, user.id, ip.address, db.name, user.name, password, host.type} \} \)
3:     \( D = \{ \text{flag, message.id} \} \)
4:     \( \text{connHost} \leftarrow \text{GENERATE_HOST_URL(host.type)} \)
5:     \( \text{connRes} \leftarrow \text{CONN_SERVER(C, connHost)} \)
6:     if connRes.IsSuccess == FALSE then
7:         return FAIL;
8:     end if
9:     bindingUserRes \leftarrow \text{USER_BINDING(user.id, user.email, targetHost)}
10:    if bindingUserRes.IsSuccess == FALSE then
11:        return FAIL;
12:   end if
13:    bindingCompRes \leftarrow \text{COMP.Binding(comp.id, targetHost)}
14:    if bindingCompRes.IsSuccess == FALSE then
15:        return FAIL;
16:   end if
17:    discGroup \leftarrow \text{SELECT_DISC()}
18:    initData \leftarrow \text{INITIALIZE(C)}
19:    sheetCode \leftarrow \text{GENERATE_SHEET_CODE(flag, initData)}
20:    sheetDate \leftarrow \text{ASSIGN_DATE()}
21:    sheetObject \leftarrow \text{SELECT_OBJECT(initData)}
22:    sheetCurrency \leftarrow \text{SELECT.CURRENCY(initData)}
23:    sheetTax \leftarrow \text{SELECT.TAX(initData)}
24:    INIT_EMPTY_SHEET_ITEM()
25:    sheetData \leftarrow \text{FILL_SHEET()}
26:    while CREATION_SUBMIT do
27:        MessageTitle \leftarrow \text{GENERATE.TITLE(sheetData)}
28:        MessageContent \leftarrow \text{GENERATE.CONTENT(sheetData)}
29:        TaxInfo \leftarrow \text{CALCULATE.TAX(sheetData, sheetTax)}
30:        TotalQuantity \leftarrow \text{CALCULATE.QUANTITY(sheetData, sheetTax)}
31:        ReportInfo \leftarrow \text{GENERATE_REPORT(sheetData, sheetCode, sheetDate)}
32:        CREATE_SHEET(sheetData, sheetCode, sheetDate, TaxInfo, TotalQuantity)
33:        CREATE_MESSAGE(discGroup, sheetCode, sheetData, MessageTitle, MessageContent)
34:        UPDATE_INVENTORY(TotalQuantity)
35:        CREATE_SHEET_LOG(sheetData, sheetCode, sheetDate, TaxInfo, TotalQuantity)
36:        UPLOAD_REPORT(ReportInfo)
37:    end while
38: end procedure
```

- function **CALCULATE QUANTITY**() output a pair of data with the item’s quantity and the corresponding warehouse. It is the necessary parameter for the subsequent function **UPDATE INVENTORY**().
- function **GENERATE REPORT**() is used to generate the corresponding report for each sheet, usually in PDF format.
- after the report generated, with function **UPLOAD REPORT**() to upload the report to the stored file server of an ECS side.
- function **CREATE SHEET**() is used for the sheet creation. The parameters will separate into two parts, one is the header, and the other is the content. As the categories of the table we mentioned in Table 5, the SuccERP API will submit the parameters for insert, update or query the relevant table.
- function **CREATE MESSAGE**() is used to create an ECS message for further communication and monitoring. The two most important parameters(title and content) are generated by functions **GENERATE TITLE**() and **GENERATE CONTENT**().
- after completing the message creation and sheet creation, the artifact will insert the log data to the ERP system according to the type of sheet via function **CREATE_SHEET_LOG**().

Results of the demonstration and evaluation

In this section, it begins by using the sequence diagram to describe how those functions of the Algorithm 2 working between API services, servers, and database sequentially as Figure 7. Besides, the demonstration also arranges snapshots in pairs with each function.

In Figure 8, we first provide the physical and dynamic IP options that users can decide how to operate ERP
depending on their current environment. We deploy our proposed artifact—SuccERP in the public server; however, as we discussed earlier, some enterprises are struggling with internal hosting issues that might be unable to build the connection with the public server. Hence, we also provide the guideline about installing and deploying SuccERP within a server in the local environment instead of the case by accessing the public server. As shown in Figure 8, we provide a red label for each solution to let the user download the SuccERP installation files and guidelines.

After confirming the connection between each service and server works well, the ECS API will send a request to

Figure 7. The sequence diagram for the case of sheet creation.
the SuccERP API for the user binding that the purpose is to create and assign the association between these two user data. As we mentioned earlier, typically, the association is according to the email of the user. The snapshots as shown in Figure 9, in this example, there are three companies exist in an ERP system, the user will choose one of them to connect,
and binding their user information with existed user in the ERP system by email address afterward. Then provide users with the options of ERP procedures, bill creation, and order creation. Finally, return the ViewModel of sheet creation as a user interface and related initial data to the ECS user for subsequent processes according to the selected option.

Figure 10 is the user interface for the sheet creation, allowing the user to carry out the specific ERP system procedure within our artifact. In this example, the user completes most of the content of this sheet. There are three items in the sheet’s content, and it will create the message in the BILL OF SALE group of the Enterprise A+ company. The user can create more or fewer items by clicking the add button or remove button. While the user fills all the necessary data of the sheet, after they click the submit button, the ECS API will send several requests to the SuccERP API and ERP database, including creating a sheet, generating a report, creating the logs, and update the inventory.

As shown in Figure 11, the ERP system can correctly read the data from its database after the sheet data submit from ECS API to the ERP database. It ensures the artifact SuccERP can deliver the data to the ERP database and owns the ability to provide a User Interface to carry out ERP procedures within SuccMail.

At the same time, the related message will be created in the SuccMail, as described in Figure 12. On the left-hand side, it is the discussion list of the Enterprise A+, and the created message is on the right-hand side. While user click the submit button as illustrates in Figure 11 that our artifact will complete the data-processing in both ERP and ECS, and the results present as Figure 11 and Figure 12. As we illustrate an example of the right-bottom corner in Figure 6 that Figure 12 is the practical case.
In this example, the message comprises a single subject, and two replies allow the employees and managers to do further discussion and communication after creating a sheet. The subject’s content is equal to the corresponding sheet data in the ERP system, and the members (employees or managers) can create the reply. The generated report data is right on the bottom of the subject, which is in PDF format. Each subject and reply will show the read or unread status of all the receipts. The username with a red color shows the user has not read this subject or reply yet; on the contrary, the username with a black color shows the user read the subject or reply. Based on the integration between ECS and SuccERP, each sheet has expanded as information in an enterprise instead of only stored in the database. Next, we get into the evaluation part. To verify that the proposed artifact-SuccERP is workable and straightforward to design, we apply the metrics to discuss it further and note the relevant notation in Table 6.

\[ n = n_1 + n_2 \]  
\[ N = N_1 + N_2 \]  
\[ D = N \times \log_2 n \]  
\[ V = \frac{n_1}{2} \times \frac{N_2}{n_2} \]  
\[ \text{MAX}(0,(171 - 5.2 \times \ln(V) - 0.23 \times CC - 16.2 \times \ln(Lines \ of \ Code)) \times 100/171) \]
We apply both the Cyclomatic Complexity and the Maintainability Index to evaluate the performance of the proposed artifact. The Cyclomatic Complexity is used to count the number of independent paths through the source code, and McCabe\textsuperscript{58} recommends developers consider each module’s Cyclomatic Complexity and divide module with the value higher than 10 into the smaller modules. Moreover, there is another research to suggest the value of Cyclomatic Complexity could accept as 15. However, it should provide a written explanation for exceeding limitations.\textsuperscript{59} The Maintainability Index is an index value between 0 and 100. Welker et al.\textsuperscript{60} suggests the value of Maintainability Index between 20 and 100 that shows the program has good maintainability, between 10 and 19 that shows the program is moderately maintainable, and between 0 and 9 that means low maintainability. We divide the evaluation of SuccERP into two parts, that is the front end and the back end. The back end is used to process the database operation, and the front end is the ViewModel, in our case, which is used to process the user interface and the relative logic.

In the back-end part’s evaluation, as shown in Table 7, all the functions have good maintainability by considering the MI metrics. However, both functions InsertOrderSheet() and InsertBillSheet() are lower than 30 on the value of MI, the reason is the function also includes inventory adjustment, log creation parts. We recommend the two functions for future research into smaller modules if there are further requirements. Then, about the Cyclomatic complexity, the function GetItemsList() is higher than 10 but lower than 15. The primary reason is this function scrutinizes each item’s value process those invalid values, including null value, invalid date-time string, and extra space. Next, the evaluation of the front-end part shown in Table 8, all the functions have good maintainability by considering the MI metrics and the value of Cyclomatic complexity is lower than 10 as well. The result shows that the ECS is simply to maintain and low complexity for expanding applications via the support from SuccERP.

## Conclusion and discussion

In conclusion, We first answer two fundamental questions proposed by Hevner et al.\textsuperscript{50} below:

- What utility does the new artifact provide?: First, our artifact-SuccERP enables the architecture of both internal and external hosting. Meanwhile, allow the enterprise to expand the applications with legacy ERP systems via our artifact. The SuccERP keeps the consistency between the ERP system and the ECS, not only implement the enhancement of communication but also meticulously sustain all the data in the legacy ERP systems. In short, we can regard SuccERP as a bridge to connect with the external

| Table 6. The notation of this paper. |
|-------------------------------------|
| Notation  | Description                  |
|\(n_1\)   | The number of distinct operators.  |
|\(n_2\)   | The number of distinct operands.  |
|\(n\)     | Program vocabulary.           |
|\(N_1\)   | The total number of operators. |
|\(N_2\)   | The total number of operands.  |
|\(N\)     | Program length.               |
|\(D\)     | Halstead difficulty.          |
|\(V\)     | Halstead volume.              |
|\(CC\)    | Cyclomatic complexity.        |
|\(CCD\)   | Cyclomatic complexity density.|
|\(MI\)    | Maintainability Index.        |
|\(LOC\)   | Lines of Code.                |

| Table 7. The metrics of functions in the back end part of SuccERP. |
|-------------------------------------|
| Function               | MI  | CC  | LOC |
| InitConnection()       | 43  | 8   | 37  |
| GetWarehouseList()     | 44  | 8   | 35  |
| GetTermsList()         | 44  | 8   | 35  |
| GetItemsList()         | 36  | 13  | 55  |
| GetItemsTypeList()     | 44  | 8   | 33  |
| GetMemberList()        | 42  | 8   | 39  |
| GetIdentityCode()      | 44  | 9   | 34  |
| GetMaxSheetCount()     | 44  | 9   | 34  |
| GetInvoiceInfo()       | 41  | 8   | 43  |
| GetDepartmentList()    | 44  | 8   | 35  |
| GetObjectList()        | 36  | 8   | 62  |
| GetCurrencyList()      | 41  | 8   | 43  |
| GetCompDeailInfo()     | 33  | 5   | 73  |
| GetAddressList()       | 40  | 8   | 43  |
| CalculateTotal()       | 63  | 1   | 8   |
| InsertOrderSheet()     | 29  | 9   | 98  |
| InsertBillSheet()      | 29  | 9   | 97  |
| UpdateUserBinding()    | 49  | 5   | 22  |
| SynchronizeSheetInfo() | 63  | 2   | 11  |
| GenerateSheetReport()  | 47  | 4   | 22  |
| GenerateReportHeader() | 82  | 1   | 3   |
| GenerateSheetPageContent() | 54 | 6   | 13  |
| GenerateSheetPageContext() | 50 | 4   | 20  |

| Table 8. The metrics of functions in the front end part of SuccERP. |
|-------------------------------------|
| Function                     | MI  | CC  | LOC  | CCD | D  | V   |
| onSheetHeaderView()          | 47.46 | 1   | 30   | 5.88% | 7.57 | 762.25 |
| handleDiscChange()           | 62.76 | 1   | 10   | 50%   | 2   | 152.29 |
| handleObjectChange()         | 37.79 | 4   | 62   | 16.67% | 19.72 | 1673.73 |
| handleDateChange()           | 70.5  | 1   | 6    | 33.33% | 2.85 | 58.81 |
| handleAddressChange()        | 66.43 | 1   | 8    | 33.33% | 2.8  | 91.37 |
| handleItemTaxChange()        | 78.82 | 1   | 3    | 100%   | 1.14 | 38.03 |
| onSettingItemTaxChange()     | 78.39 | 1   | 3    | 100%   | 1.14 | 38.03 |
| onSheetContentView()         | 57.87 | 1   | 15   | 14.29% | 5   | 215.22 |
| onAddItem()                  | 67.41 | 1   | 7    | 33.33% | 2.16 | 71.69 |
| onRemoveItem()               | 62.78 | 1   | 10   | 40%   | 7.5  | 128.00 |
| handleWarehouseChange()      | 69.29 | 1   | 6    | 33.33% | 2.8  | 87.56 |
| onQuantityChange()           | 68.3  | 2   | 7    | 33.33% | 2.16 | 71.69 |
| onPriceChange()              | 58.57 | 3   | 16   | 75%   | 6    | 128.00 |
| onGenerateECSMsg()            | 33.34 | 2   | 84   | 4.35% | 10.4 | 3066.71 |
systems. We proposed the complete architecture and implementation process in this research.

- What demonstrates that utility?: an exploration of the CSFs and learning issues are popular research topics still in the post-implementation stage. However, most of the results are standing on the academic aspect. For an enterprise, they need a more practical case as a guideline. Hence, we present the complete steps and architecture to construct the artifact and present the snapshots and complexity metrics to show how it works and integrates the legacy ERP system and external systems.

we also highlight the contributions below:

- We believe that this is the first research to construct a post-implementation and show how to achieve the integration between ERP system and ECS by following the guidelines of DS.
- Instead of providing case studies, we present a complete process for investigating the structure of various ERP systems. It will make our research results widely available to the development of communication between different ERP systems and ECS.
- Truth and utility (artifacts that are effective) are two sides of the one coin. We list the weaknesses of both ECS and ERP system from the literature review, to make the purpose of this research is based on the theoretical foundation. Also, we consider the guidelines of DS to demonstrate how to develop and design the artifact completely.

Implications and contributions

Considering the way to reduce business costs is by eliminating poor communication and coordination between the organization and its suppliers and customers. Langenwalter\textsuperscript{21} created the term “Total Enterprise Integration (TEI)” to describe the process of integrating all the information and operations required to support an organization and its supply chain fully. The concept involves integrating information across the organization and throughout the supply chain to customers and suppliers, which is in communication, letting everyone know the full ramifications of current demands and tasks. We believe that this concept is a more pragmatic way of presenting the needs of organizations. To the extend, the SuccERP implements such a concept through achieving the integration between the ERP system and the ECS system. In addition to the CSFs identified from the literature review, we have also recognized issues such as integration, communication, collaboration, and technical pitfalls. Thus, we present how to develop ERP and ECS integration upon the DS methodology, and proposing artifact-SuccERP, our results are more pragmatic and complete the lack of findings on post-implementation research’s practicality.

Limitations and future works

The empirical results reported herein should be considered in light of some limitations. The main objective of this study was to show how we proposed a manual piece to implement ERP system integration by reviewing the post-implementation issues and suggestions. However, in terms of issues and suggestions, we focused more on collaboration, communication, and information sharing, making the ECS system our target in implementing an ERP integration. Lanchaoen et al.\textsuperscript{62} consider the integration of information to improve healthcare organizations’ business processes; motivated by that, the future work could direct on evaluating the usability and potential of the system in various industries by targeting specific industries.

Acknowledgments

We would like to thank two anonymous reviewers, the editor Dr. Gupta, Annapurna, and Editor-in-Chief Prof. W.H. IP, for their comments. We are grateful to Prof. Tzu-Hang Chiang for letting us make use of his developed system-SuccMail and his helpful suggestions on the first draft of this paper. It would not have been possible without their help.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Yung-Yu Lin https://orcid.org/0000-0002-1928-714X

References

1. Babaei M, Gholami Z and Altafi S. Challenges of enterprise resource planning implementation in Iran large organizations. Inf Syst 2015; 54: 15–27.
2. Mraz SJ. Keeping up with ERP. Machine Design 2000; 72(14): S56–S56.
3. Nicolaou AI. Firm performance effects in relation to the implementation and use of enterprise resource planning systems. J Inf Syst 2004; 18(2): 79–105.
4. Carton F, Sammon D and Adam F. Benefit realisation through ERP: the re-emergence of data warehousing. Electron J Inf Syst Eval 2003; 6(2): 155–164.
5. Yi MY and Davis FD. Developing and validating an observational learning model of computer software training and skill acquisition. Inf Syst Res 2003; 14(2): 146–169.
6. Chou HW, Chang HH, Lin YH, et al. Drivers and effects of post-implementation learning on ERP usage. Comput Human Behav 2014; 35: 267–277.
7. Esteves J and Bohorquez V. An updated ERP systems annotated bibliography: 2001-2005. Commun Assoc Inf Syst 2007; 19: 18.
8. Swanson EB and Ramiller NC. Innovating mindfully with information technology. MIS Q 2004; 28(4): 553–583.
9. Willis TH and Willis-Brown AH. Extending the value of ERP. Ind Manage Data Syst 2002; 102: 35–38.
10. Moon YB. Enterprise resource planning (ERP): a review of the literature. Int J Manag Enterpr Dev 2007; 4(3): 235–264.
11. Law CC, Chen CC and Wu BJ. Managing the full ERP lifecycle: considerations of maintenance and support requirements and IT governance practice as integral elements of the formula for successful ERP adoption. Comput Ind 2010; 61(3): 297–308.
12. Grabski SV, Leech SA and Schmidt PJ. A review of ERP research: a future agenda for accounting information systems. J Inf Syst 2011; 25(1): 37–78.
13. Hsu PF, Yen HR and Chung JC. Assessing ERP post-implementation success at the individual level: revisiting the role of service quality. Inf Manage 2015; 52(8): 925–942.
14. Ali M and Miller L. ERP system implementation in large enterprises—a systematic literature review. J Enterp Inf Manag 2017; 30(4): 666–692.
15. McGinnis TC and Huang Z. Rethinking ERP success: a new perspective from knowledge management and continuous improvement. Inf Manage 2007; 44(7): 626–634.
16. Oses KB, Olsen JR, Vassilakopoulou P, et al. ERP systems in multinational enterprises: a literature review of post-implementation challenges. Procedia Comput Sci 2018; 138: 541–548.
17. Hasan N, Miah SJ, Bao Y, et al. Factors affecting post-implementation success of enterprise resource planning systems: a perspective of business process performance. Enterprise Inf Syst 2019; 13(9): 1217–1244.
18. Hämäläinen L and Hilmola OP. Life after ERP implementation. J Enterp Inf Manag 2008; 21(3): 285–310.
19. Ha YM and Ahn HJ. Factors affecting the performance of enterprise resource planning (ERP) systems in the post-implementation stage. Behav Inf Technol 2014; 33(10): 1065–1081.
20. Yu CS. Causes influencing the effectiveness of the post-implementation ERP system. Ind Manag Data Syst 2005; 105(1): 115–132.
21. Ram J, Corkindale D and Wu ML. Implementation critical success factors (CSFs) for ERP: Do they contribute to implementation success and post-implementation performance? Int J Prod Econ 2013; 144(1): 157–174.
22. Giachetti RE. A framework to review the information integration of the enterprise. Int J Prod Res 2004; 42(6): 1147–1166.
23. Peng GC and Nunes MB. Surfacing ERP exploitation risks through a risk ontology. Ind Manag Data Syst 2009; 109(7): 926–942.
24. Kremers M and Van Dissel H. Enterprise resource planning: ERP system migrations. Commun ACM 2000; 43(4): 53–56.
25. Oseni T, Rahim MM, Smith S, et al. Exploring ERP post-implementation modifications and their influence on business process outcomes: a theory driven model. In: ACIS 2013 proceedings, 2013, p. 23.
44. Boza A, Cuenca L, Poler R, et al. The interoperability force in the ERP field. *Enterp Inf Syst* 2015; 9(3): 257–278.
45. Al-Mashari M. Constructs of process change management in ERP context: a focus on SAP R/3. In: *AMCIS 2000 Proceedings*, 2000, p. 113.
46. Ross JW and Vitale MR. The ERP revolution: surviving vs. thriving. *Inf Syst Front* 2000; 2(2): 233–241.
47. Aboulafia M. *Philosophy, social theory, and the thought of George Herbert Mead* (SUNY Series in Philosophy of the Social Sciences), State University of New York Press, Albany, NY, 1991.
48. Lee A. Systems thinking, design science, and paradigms: heed- ing three lessons from the past to resolve three dilemmas in the present to direct a trajectory for future research in the information systems field, keynote address. In: *Eleventh international conference on information management*, Taiwan, 2000.
49. Simon HA. *The sciences of the artificial*. Cambridge, MIT press, 2019.
50. Hevner AR, March ST, Park J, et al. Design science in information systems research. *MIS Quart* 2004; 28(1): 75–105.
51. Peffers K, Tuunanen T, Rothenberger MA, et al. A design science research methodology for information systems research. *J Manage Inf Syst* 2007; 24(3): 45–77.
52. Davenport TH. Putting the enterprise into the enterprise system. *Harvard Business Rev* 1998; 76(4): 121–132.
53. Liu Q and Liu G. Research on the framework of decision support system based on ERP systems. In: *2010 second international workshop on education technology and computer science*, 2010, pp. 704–707. IEEE.
54. Weston FC, Jr. ERP implementation and project management. *Prod Invent Manage J* 2001; 42(3/4): 75.
55. Elragal A and Hassanien HE. Augmenting advanced analytics into enterprise systems: a focus on post-implementation activities. *Systems* 2019; 7(2): 31.
56. Lin Y, Nagai Y, Chiang T, et al. Design and develop artifact for integrating with ERP and ECS based on design science. In: *Proceedings of the 2020: The 3rd international conference on information science and system*, New York, 2020, pp. 218–223.
57. Donald M and Judy W. *The social shaping of technology*. Maidenhead, Open University Press, 1999.
58. McCabe TJ. A complexity measure. *IEEE Trans soft Eng* 1976; 4: 308–320.
59. Watson AH, Wallace DR and McCabe TJ. *Structured testing: a testing methodology using the cyclomatic complexity metric*, vol. 500. Maryland: US Department of Commerce, Technology Administration, National Institute of Standards and Technology, 1996.
60. Welker KD, Oman PW and Atkinson GG. Development and application of an automated source code maintainability index. *J Soft Main Res Pract* 1997; 9: 127–159.
61. Langenwalter GA. *Enterprise resources planning and beyond: integrating your entire organization*, vol. 12. Boca Raton: CRC Press, 2020.
62. Lanchareon S, Sukswang P and Naenna T. Readiness assessment of information integration in a hospital using an analytic network process method for decision-making in a healthcare network. *Int J Eng Bus Manag* 2020; 12: 1847979019899318.