*Intelligent Automation & Soft Computing*
DOI:10.32604/iasc.2021.014280
***Article***

Tech Science Press

# CMMI Compliant Modernization Framework to Transform Legacy Systems

**Musawwer Khan[1], Islam Ali[1], Waqar Mehmood[1], Wasif Nisar[1], Waqar Aslam[2], Muhammad Shafiq[3] and Jin-Ghoo Choi[3,*]**

[1]Department of Computer Science, COMSATS University Islamabad, Wah Cantt, 47040, Pakistan
[2]Department of Computer Science & IT, The Islamia University of Bahawalpur, Bahawalpur, 63100, Pakistan
[3]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, 38541, Korea
[*]Corresponding Author: Jin-Ghoo Choi. Email: jchoi@yu.ac.kr
Received: 10 September 2020; Accepted: 10 October 2020

**Abstract:** Legacy systems deteriorate due to changing business practices and recurrent problems such as high maintenance cost, lack of agility, and degraded performance. Modernization is highly desired especially for those organizations where key business processes are managed by legacy systems. During modernizing, two important aspects are usually ignored, i.e., the transformation of a legacy system to an enterprise solution, and considerations of quality concerns. This overlooking leads to a modernized information system that partially achieves the expected outcome. In this paper, we propose a Capability Maturity Model Integration (CMMI) Compliant Modernization Framework (CCMF) that addresses the problems of legacy systems by modernizing to an enterprise system in order to meet the expectations of stakeholders. CCMF is compatible with CMMI Level 2, which is a world-renowned software process improvement standard. Our proposed framework consists of eight distinct stages containing best practices with expected outcomes that are compliant with the specific goals of CMMI Level 2. The framework is based on various strategies that ensure the seamless modernization of legacy systems. Our contributions are twofold. First, it supports the modernization of outdated legacy systems to a more reliable enterprise solution. Second, it is compliant to CMMI standard with due consideration of quality and reliability concerns of the evolved modernized systems. The usability, scalability, and usefulness of the framework are validated by expert judgment on an industrial case study. Our proposed framework is also applicable for change-resistant companies with monolithic systems.

**Keywords:** Software engineering; legacy system; ERP; modernization; information System; CMMI

## 1 Introduction

Legacy systems (LSs) are outdated systems that are still valuable for the organizations. The term "legacy system" has been defined by Alkazemi et al. as "A running system that still meets a considerable percentage of an enterprise's business needs in terms of functional aspects, but does not comply with emerging

architectural standards" [1]. LSs refer to any business-critical software systems that are the backbone of enterprises. These systems resist modifications that are required due to changing requirements, thus risk daily operations [2,3]. Conteh et al. defines an LS as not upgradeable [4], hence they have to be replaced carefully. Other than those LSs become inefficient technically, they also lead to unmanageability at the enterprise level [5]. Business enterprises invest huge money in information systems that serve their key functions like Human Resources, Marketing, Finance, Sales, Transportation, etc. Due to ad-hoc decisions made by organizations, not all information systems integrate well with one another thus hinders the ability of upper management to make optimal decisions [4,6,7]. According to Shatat et al., switching from a traditional business process to a new way of conducting business through implementing a new information system is considered a difficult task [8]. With the software replacement, a significant amount of business knowledge gained over the period of time is lost [9]. LSs evolve in three ways: maintenance, modernization and replacement [10,11]. Maintenance is costly and many small changes are more cumbersome than one big change. Replacement is more resource consuming and requires rigorous testing. Its replacement with an ERP system allows preserving the data and functionality of the LS, hence ranks as the most suitable solution [10]. The term "modernization" is described as involving significant changes, such as implementation of new and relevant functional requirements, a modification on the software architecture, or a system migration to a new software platform [12]. White-box modernization means to know the internal details of the LS while black-box modernization is about knowing the external interfaces [11]. Though many efforts have been made in the modernization of LSs, but unfortunately we hardly find efforts that propose a quality framework that guides through a process approach. Nevertheless, ERP systems provide integrated and unified information solutions that facilitate businesses and automate operations for better reporting and informed decision making. Some common examples of enterprise systems are Financial Management, Human Resources, Inventory Management, Supply Chain Management, Business Intelligence, Customer Relationship Management, Distribution Management, Manufacturing Management, etc. Generally, an LS to a ERP system migration is not based on a quality-driven approach, one that is based on well-defined processes; hence fail to produce a complete solution that can address the expected outcome, even reliability is compromised. With this background, we specifically formulate three research questions:

RQ1: How can we modernize a legacy system to an enterprise solution?

RQ2: How much is our proposed modernization solution compliant to CMMI standard?

RQ3: How is the dependence of the quality of a modernized system on the quality of approach adopted?

We focus on the white-box modernization so that the entire LS along with its data and functionality, is modernized keeping the user's trust alive. We contribute by proposing a framework that provides a complete solution by transforming existing LS to an enterprise system. It is fully compliant to the Specific Goals (SGs) of all Process Areas (PAs) at CMMI Level 2. This feature augments on the quality and reliability concerns.

The paper is organized as follows. In Section 2, we discuss the related work. Section 3 introduces and elaborates the proposed CCMF along with its implementation details. Section 4 explains the validation process of the proposed framework including case study and results. In Section 5 we have presented the compliance of our framework to CMMI using compliance chart. Finally, Section 6 gives the conclusion and the future work.

## 2 Related Work

In 1999, Carnegie Mellon University published a report "Why Reengineering Projects Fail," in which the main reason of failure was pointed out to be "The organization inadvertently adopts a flawed or incomplete reengineering strategy" [13]. Thus, a lot of effort is required to upgrade or change an LS while keeping its main functionality undisturbed [14]. More than 70% of the research work in the recent

past pertains to the managerial aspects of the modernization activities [12]. The key challenges that hinder an LS modernization are time of transition, legacy data migration, complex system architecture, lack of system understanding and resistance from within the organizations [3]. Four recurrent reasons that trigger modernization of LS are integration of LSs, software flexibility requirement, little or no knowledge about the system, and the error proneness [12]. The ways to modernize LSs are forward engineering, reverse engineering and restructuring [2].

Various frameworks have been proposed for modernization, though complete solutions are still missing. A proposed framework named Legacy System Assessment Conceptual Framework (LSACF) discusses the possibility of reengineering the components of an old system for using in a new system [2]. LSACF has qualification, selection and transition phases to transform the LS into a new one. SMLC (Software Migration Life Cycle) is proposed by Althani et al., which is based on selecting a suitable pathway using Analytic Hierarchy Process (AHP) [15]. SMLC prioritizes phases/tasks in the life cycle, enabling consideration of quality concerns at each phase of the life cycle. A model driven reverse engineering framework can aim at identifying the functionalities, architecture and data of legacy applications [16]. Though, it has a generic support for various metamodels and formats such as UML, XML, SQL, COBOL, etc., its role is limited to only initial phase of modernization. Another model GEQUAMO proposes criteria to decide about feasibility of restructuring. It allows determining qualitatively and quantitatively, the improvements both in process and product [17]. Reengineering of LSs is possible with an aim to replace them, though it lacks consideration of the quality concern and elucidation of the process followed [18]. It also lacks comprehensive risk management at each each phase of the project life cycle therefore the risks associated are also being managed improperly. The identification of risk factors, avoidance and mitigation strategies have been discussed in [19], which according to the author can be useful for small and medium enterprises involved in software development.

During reengineering, it might be desired to integrate LS to commercial software with flexible software architecture. There is a proposal to explore the multi-objective solution space using integer programming, thereupon selecting feasible services [20]. Generally, application code, information system and infrastructure are independent entities, thus they can be modernized separately [21]. This proposal is based on incremental transformation consisting of pre-processing, processing and post-processing phases. It suffers from limitations like human factors, vagueness on 'Return on Investment' and limited case study validity. With the advent of Service Oriented Architecture (SOA), the migration of LSs to SOA has become more challenging [22] due to implementation issues, selection of transformation tools and the change in the architectural tiers. Another model studies effectiveness using cause-effect relationships [23]. A Knowledge Discovery Metamodel is proposed for modeling various artifacts at different abstraction levels of LS [24]. It supports the Architecture Driven Modernization (ADM) which is a widely used solution for the evolution of LSs. Another ADM was adopted in the Eurocat Product-line comprising of around 1.6 million lines of code [25]. Improvements in both code and design quality were achieved.

There have been efforts to transform LSs by changing the code language, for instance, COBOL to Java [26] or introduction of graphical user interfaces, CRUD logic and PLSQL triggers [27]. Again, the focus is for particular cases, rather than following some modernization strategy. Another extended effort extracts business rules from the legacy code to form interconnected components [28].

In our research we have primarily focused on the managerial aspect of the modernization as this area has gained much attention. In context of the implementation of CMMI Framework, Niazi et al. implemented a specific practice 1.3 of Requirements Management process area at Level 2 of CMMI [29]. They proposed a model for managing requirements changes and conducted a case study to validate the practice. CMMI Level 2 process areas are largely supporting process areas relating to project management and engineering, which can be implemented in the software process improvement programs. CMMI Level 2 process areas such as Project Planning, Requirements Management and Project Monitoring and Control can be mapped to scrum model [30].

All the discussed works mainly focus on the modernization of LS to a new system which solves the problems that existed in the old system. Our framework overcomes the issues of LSs by modernizing them to enterprise system using quality practices and techniques.

## 3 CMMI Compliant Modernization Framework (CCMF)

An ERP system allows intercommunication between various business processes and flow of data between them. It collects an organization's shared transactional data from multiple sources, eliminates data duplication and provides data integrity with a single source of truth. Our proposed framework is a complete methodology that transforms ordinary and low scaled LS into a large scale enterprise system. Each stage of the framework consists of key practices and expected outcome, steering it compliant with CMMI Level 2. This approach is comparatively novel, and helps enterprises seamlessly modernize their LSs through quality best practices, thus ensuring the development of reliable enterprise level software systems.

Our proposed framework has four main components Fig. 1: Modernization Requirements, Modernization Roadmap, CMMI Framework and the Modernized Enterprise System. The modernization roadmap is the main component of the framework containing eight stages, with output of one taken as input of next. The framework is discussed in detail in the next section.

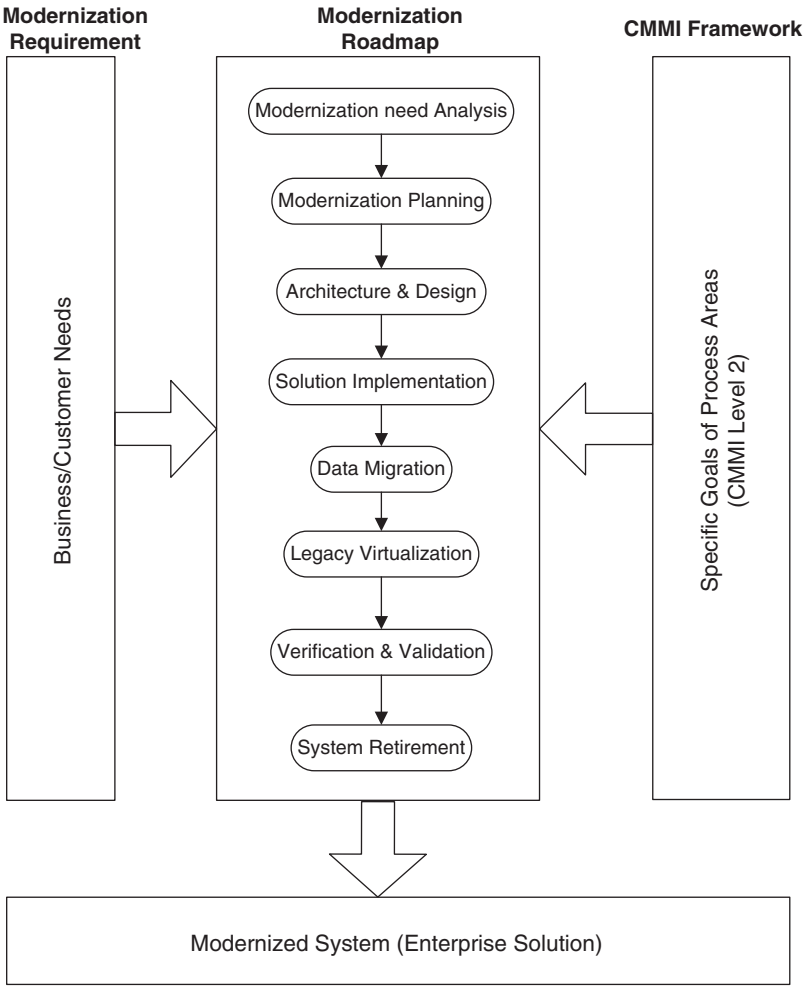


**Figure 1:** The layout of CCMF

### *3.1 Modernization Requirements*

Any modernization project must satisfy the business needs of organization, while its outcomes should be according to management expectations at all levels. Many modernization projects fail due to ignoring these aspects. It is very important to focus on the exact vision of the organization, and understand the management's expectations. The management may express their needs using business language such as "We want to enhance the carrying capacity of the enterprise (that is the total number of clients/customers served per day) and also to reduce the expenses by 10% by February. We also need to cut down the registration process from 25 to 10 minutes on average by February and reduce client wait time from 20 to 5 minutes on average by Mar 2020." These types of statements express extra-functional requirements per business needs. In the real world, many kinds of statements and scenarios may arise that must be analyzed judiciously and dug deeply so that the desired outcome is achieved successfully.

### *3.2 Modernization Roadmap*

Modernization roadmap is based on stages, with each one consisting of three main elements i.e., Modernization Goal (MG), Key Practices (KPs) and Expected Outcome (OE). Adopting KPs contributes to the achievement of the respective Modernization Goal (MG) and implementations of all MGs consequently lead to achievement of the relevant OE.

#### *3.2.1 Modernization Need Analysis (MNA)*

Identification of business needs necessitates analysis. This include defining project scope, understanding all stakeholders requirements, detailed analysis of all LSs (within the scope), identification of associated issues and problems, identifying user's expectations from modernized system and new features required by the business.

As shown in Fig. 2, MNA is a three steps process, viz, Understand, Analyze and Document. The purpose of the first step is to clearly define and understand the organization's vision and the user's expectations from the new system. Once the needs and requirements are understood clearly, the next step is to analyze the requirements in detail. Analysis is about investigating the operational aspect of the system. The operational execution means to examine the development and maintenance of product and product components with respect to the intended environment. We find answers to the questions like "How operations are being performed currently?", "What the client feels about the enterprise or company?", "How effective is the current process serving the clients/customers?" and "What issues and problems are being faced by the end users while using the LS?" Finally, all the requirements are documented. They are sufficient to provide strong basis for preparing a comprehensive system design. The EO and KPs performed at this stage are given next.

EO-I: Well documented requirements approved by the relevant stakeholders

MG 1: Understand Business Needs

- KP 1.1 Comprehend the scope of the project
- KP 1.2 Understand problems in the existing legacy environment
- KP 1.3 Clearly understand the requirements of all the relevant stakeholders including external entities
- KP 1.4 Identify third party tools, agreements and other statutory requirements
- KP 1.5 Discuss the project's outcome with the higher management

MG 2: Analyze Requirements

- KP 1.1 Analyze the existing business processes and the LS
- KP 1.2 Identify issues and discuss these with the top management for resolution

- KP 1.3 Ensure resolution of issues

MG 3: Document and Maintain Requirements

- KP 1.1 Prepare Requirements Traceability Matrix to keep track of requirements changes
- KP 1.2 Take assurance on roles and responsibilities of all the participants
- KP 1.3 Discuss the requirements with the relevant stakeholders, and resolve issues
- KP 1.4 Document requirements signed by all the relevant stakeholders

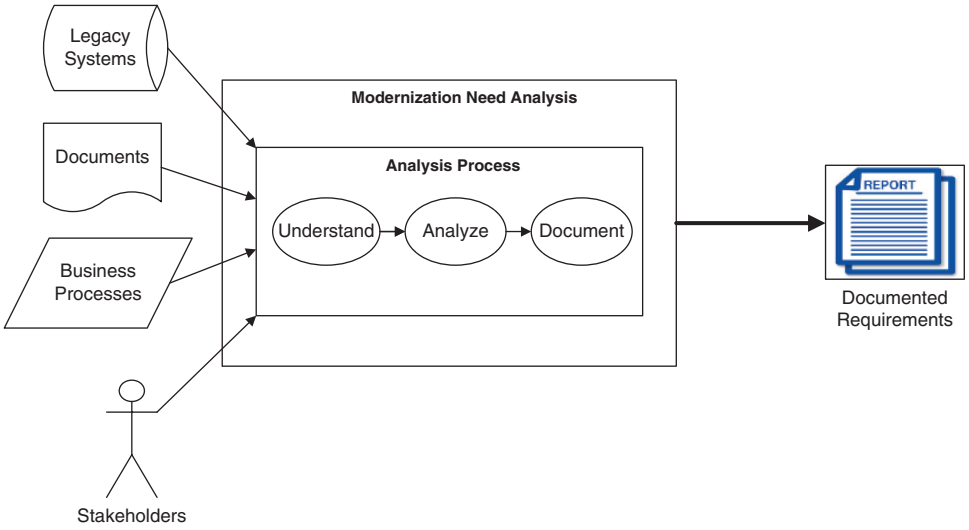


**Figure 2:** A modernization need analysis process

*3.2.2 Modernization Planning (MP)*

Planning is a very important stage in the modernization roadmap of CCMF. It defines how to complete the project within the stipulated time and using the allocated resources. The sub-activities of planning include defining objectives, identifying deliverables, schedule planning and preparing the supporting plans, i.e., SQA plan, Test Plan, Project Monitoring Plan, etc. Modernization planning consists of the following EO and KPs.

EO-II: A detailed Project Plan containing all the necessary details required for project execution

Mg 1: Prepare Project Plan

- KP 1.1 Estimate project scope, and obtain commitments of all the relevant stakeholders including external suppliers
- KP 1.2 Estimate project scope, resources, efforts, cost and time
- KP 1.3 Prepare work breakdown structure and define responsibilities
- KP 1.4 Prepare and maintain a comprehensive project plan and communicate that to the relevant stakeholders

*3.2.3 Architecture and Design (A&D)*

Architecture and Design is another crucial stage of the modernization framework that builds foundation for the new system. A&D is all about defining and designing a collection of software components and their interfaces to establish the architecture for the development of enterprise system. Poor architecture and designs cause massive development and maintenance cost, which leads to project failure.

A&D stage has been elaborated in six distinct steps that are executed in the order <Architecture Decision, Design and Development Standards, Foundation System, Core module (s), System Interfaces, Design Verification and Validation >. All the steps are interdependent, that is output of one step is input for the next step.

*Architecture Decision*

Architecture decision is related to selection of appropriate architecture style for solution. The commonly available architecture styles are Data Centric Architecture, Data Flow Architecture, Call and Return Architecture, Layered Architecture and Object Oriented Architecture. This framework does not recommend or enforce the selection of a particular style. The selection of the appropriate architecture style depends on the requirement and nature of the projects.

*Design and Development Standards*

The design and development standards are prepared before developing any artifact of the proposed system. These standards are collected and documented after reviewed by the design team. All the artifacts being prepared follow the rules and guidelines defined in the design and development standard document.

*Foundation System*

For almost every ERP system, there has to be a base or foundation software module to control and manage the functions of all other modules. This system is usually configured and managed by the ERP management team. Any adjustment or configuration required in the operation sequence or execution is performed via Foundation System. Without this controlling system, seamless operation of integrated software systems cannot be ensured.

*Core Module (s)*

The core module, selected during the MNA phase, is designed to facilitate the integration of enterprise system components. The core module is integrated or referenced by almost every software module running in the legacy environment. In an enterprise system, it is the most commonly used software module that must be designed and developed early so that the integration is managed properly. The most common example of such module is Human Resource System because almost every information system requires integration with it.

*System Interfaces*

In an enterprise system all the software modules are well integrated, sharing information through standard interfaces. Various functions/procedures are defined along with its pseudo code to provide a transparent layer among systems independent to any underlying design changes.

*Design Verification and Validation*

Once the design of the core module and its interfaces are developed, it is carefully reviewed by the team of experts in context of requirements specifications and design standards. The changes, if any, are made to the design and it is handed over to the development team. Before embarking upon the next stage, the design must be carefully verified and validated because any change to the design at a later stage is more costly. The EO and the KPs to be performed at this stage are given next.

EO-III: To establish and maintain a detailed design document based on requirements specifications

MG 1: Decide a Suitable Architecture

- KP 1.1 Choose the suitable Architecture and define its components
- KP 1.2 Evaluate the selected Architectural decisions

MG 2: Develop Standards

- KP 1.1 Prepare and maintain database design standards
- KP 1.2 Identify and design reusable components
- KP 1.3 Develop interface design standards
- KP 1.4 Prepare and maintain coding standards

MG 3: Design Foundation System

- KP 1.1 Devise method for configuring users and access profiles
- KP 1.2 Design an Enterprise Access Control
- KP 1.3 Plan and design the document workflow
- KP 1.4 Design conformant entities for uniformity

MG 4: Design Base Software Module

- KP 1.1 Identify issues in the existing core module
- KP 1.2 Design the proposed core module and address all issues
- KP 1.3 Identify integrations with associated software modules
- KP 1.4 Transform requirements into design specifications

MG 5: Design Integration

- KP 1.1 Devise standard transparent layer for interfacing among software system to share data
- KP 1.2 Methodically review the interface mechanism

MG 6: Validate Design

- KP 1.1 Validate the Architecture and design document
- KP 1.2 Resolve issues and readjust the design

*3.2.4 Solution Implementation (SI)*

After detailed verification and validation of design and architecture, the implementation of core modules is started. Typically the Human Resource System, Work Flow Management and foundation systems are kind of software that are referenced by majority of software modules in an enterprise software system. The implementation of core modules uses the design specifications already prepared in the MP stage of modernization roadmap.

The core module is the most integrated software system in an enterprise system as shown in Fig. 3. SI stage is not only related to the implementation of core software modules but also for the implementation of all other LSs. All the stages are executed for the rest of LSs in the same way as it is executed for core systems. It is important to mention that core software systems are selected for modernization before selection of other LSs, so that a platform is set for modernization of rest of the systems. The solution implementation stage contains the following EO and KPs.

EO-IV: Modernized Software Systems aligned with the design specifications

MG 1: Develop Core/legacy Software System(s)

- KP 1.1 Implement core/legacy software system according to the design specifications
- KP 1.2 Monitor the progress according to the planned parameters and resolve deviations
- KP 1.3 Ensure all the features of a legacy system and the additional user requirements are modernized

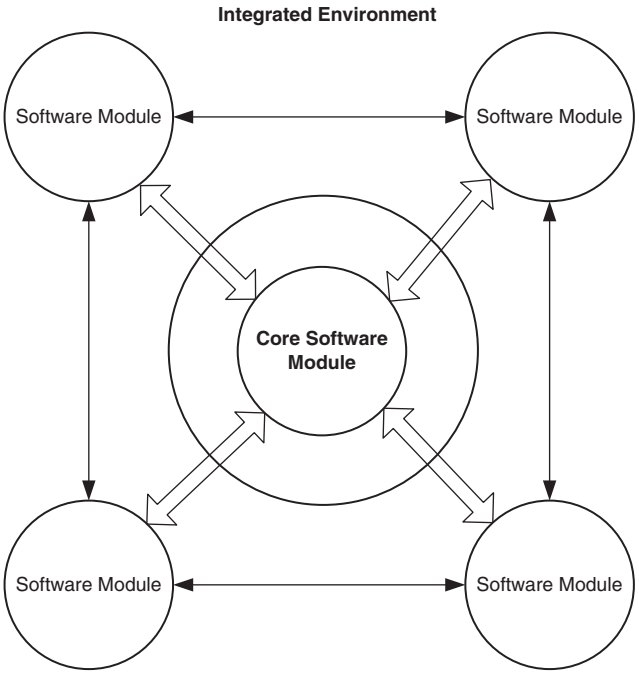


**Figure 3:** An illustration of core modules in the integrated environment

*3.2.5 Data Migration (DM)*

Data migration is the most critical, challenging and time consuming activity of modernization project. In the data migration stage, the data from LSs is selected and loaded into the newly developed core system(s). This is a cumbersome activity that needs much attention and requires accuracy. It is pertinent to mention that wrong or incomplete data migration activity could be catastrophic, so this activity must be completed carefully. The legacy data is an asset for organizations, and especially large organizations never want to lose it. It plays a key role in day-to-day operations and utilized for future predictions. The KPs along with the EO to be achieved at this stage are given next.

EO-V: Quality Data Migration

MG 1: Migrate Legacy Data

- KP 1.1 Analyze the legacy data and identify data quality issues
- KP 1.2 Set-aside the data quality issues that need to be resolved in the LS
- KP 1.3 Devise a strategy for improving data quality in the existing LS
- KP 1.4 Prepare a mapping sheet for tracking source to target migration
- KP 1.5 Create data migration script
- KP 1.6 Test data migration script
- KP 1.7 Migrate data
- KP 1.8 Monitor the progress regularly, and ensure the data migration completes well in time
- KP 1.9 Validate data

*3.2.6 Legacy Virtualization (LV)*

After successful data migration, we are ready to deploy our first modernized software module. For that purpose we need to setup a virtual environment. The virtual legacy environment is a layer built upon the

newly developed modernized software to provide a virtual view of the LS to the rest of software operational in the legacy environment in order to keep these systems undisturbed. The EO and KPs performed at this stage are given next.

EO-VI: Seamlessly Operational Legacy Environment

MG 1: Prepare Virtual Legacy Environment

- KP 1.1 Identify all the components being accessed by the legacy environment
- KP 1.2 Ensure the legacy environment is undisturbed and make quick corrective actions to resolve issues
- KP 1.3 Create a virtual layer upon the modernized system to behave like the LS
- KP 1.4 Validate if the virtual layer is providing the desired output

#### *3.2.7 Verification and Validation (V&V)*

Verification and validation stage is all about confirmation and testing of the modernized system after implementation and data migration. The testing of system is done by both the developers and users. When the newly modernized system gets verified, it is handed over to the users to validate if it fulfills all the requirements and provide the desired outcome. The EO and KPs at this stage are given next.

EO-VII: A Ready-to-Use Core Modernized System

MG 1: Validate the System

- KP 1.1 Deploy the Core Modules
- KP 1.2 Create the virtual legacy environment
- KP 1.2 Verify all inputs/outputs of the modernized system
- KP 1.3 Create a virtual operational environment where the modernized system is intended to be used
- KP 1.4 Verify if all features of LS exist in the modernized system
- KP 1.5 Test the system in user environment
- KP 1.6 Ensure seamless functionality of third party items (if any)
- KP 1.7 Ensure all issues have been tracked to closure

#### *3.2.8 System Retirement (SR)*

Finally access to the LS is gradually discontinued and the legacy virtualization layer is progressively dismantled. The integration links are redirected to the new enterprise system as and when a system gets modernized. All the stages mentioned are repeated for modernizing of each LS. In this way, the entire LS is seamlessly reengineered into a new enterprise system, i.e., when the modernization of one system takes place all other integrated systems remains undisturbed. The key EO and KPs performed at this stage are given next.

EO-VIII: A Reliable Modernized Enterprise Software Solution

MG 1: Retire the legacy system

- KP 1.1 Discontinue access to all objects of the LS and ensure that no legacy component is in use
- KP 1.2 Train the user on the new system and ensure user's trust
- KP 1.3 Backup the LS for future reference

### *3.3 CMMI Framework*

It is a collection of best practices to improve business processes in organizations. These models are developed by Software Engineering Institute (SEI). For software products and services, the CMMI

framework offers a set of guidelines that are comprehensively designed and integrated. Whenever EOs at each stage of our modernization roadmap are well aligned with the SGs of each Process Area of CMMI Level 2 framework, it justifies the quality and the reliability of our proposed framework. In the later part of this paper, we shall discuss the compliance to CMMI framework in details.

### *3.4 Modernized System (Enterprise Solution)*

After successfully going through all the stages of modernization roadmap, the expected enterprise system is achieved. The newly developed system serves as a complete enterprise solution for organizations. It is a quality system integrating all sub-systems into one unified solution that fulfills the needs of all users at different levels in an enterprise. The requirements related to operational, managerial and informed decision making are successfully achieved through this reliable enterprise system.

## 4 Validation of CCMF

We validated CCMF both quantitatively and qualitatively. For the former, CCMF containing modernization roadmap along with EOs, MGs and its associated KPs were thoroughly reviewed by a panel of experts. Using Delphi method, rating based approach was used for analysis and compilation of results. For the latter, the proposed framework was applied on all six types of LSs. The results achieved after executing the case study have been discussed in details in the relevant section.

### *4.1 Expert Judgment*

The proposed CCMF was validated by the industry experts. Total 18 experts were engaged from 7 different software development companies. The Delphi method was adopted and "Likert Scale" was used to rate the opinions.

The names of experts are not shown for anonymity; however other relevant details are shown in Tab. 1. All the experts involved in the process, had diversified experience of more than 10 years in their relevant field of work. Multiple brainstorming sessions were conducted to review the proposed framework. At the end, a common consensus was developed to select the optimal choices. The process of validating quantitative method is shown in Fig. 4, which shows the process of selecting most suitable components for CCMF.

**Table 1:** The details of experts

| Domain | No. of Experts | Experience | Company |
|---|---|---|---|
| SEPG Members | 2 | 15 years | A |
| CMMI Auditors | 1 | 12 years | B |
| Project Managers | 4 | 13 years | C |
| QA Managers | 3 | 15 years | D |
| Software Team Leads | 3 | 13 years | E |
| ERP Consultants | 4 | 11 years | F |
| ERP Implementation Partner | 1 | 12 years | G |

In order to develop the framework structure we followed the design science research guidelines [31]. According to the guidelines, the model or artifact is designed first, followed by the problem relevance, evaluation, contribution, selection of optimal solution and application. After finalizing the proposed framework it was presented to the panel of experts. The experts rated each component in terms of

inclusion/exclusion, title, location and its logical order, etc. Each component of the framework was rated in five categories of importance in the respective domain. The given choices were less important, marginally important, moderately important, very important and highly important with associated weights of 1, 2, 3, 4 and 5 respectively. The weighted averages were absolutely rounded while excluding the values below 3 in order to get more quality results.

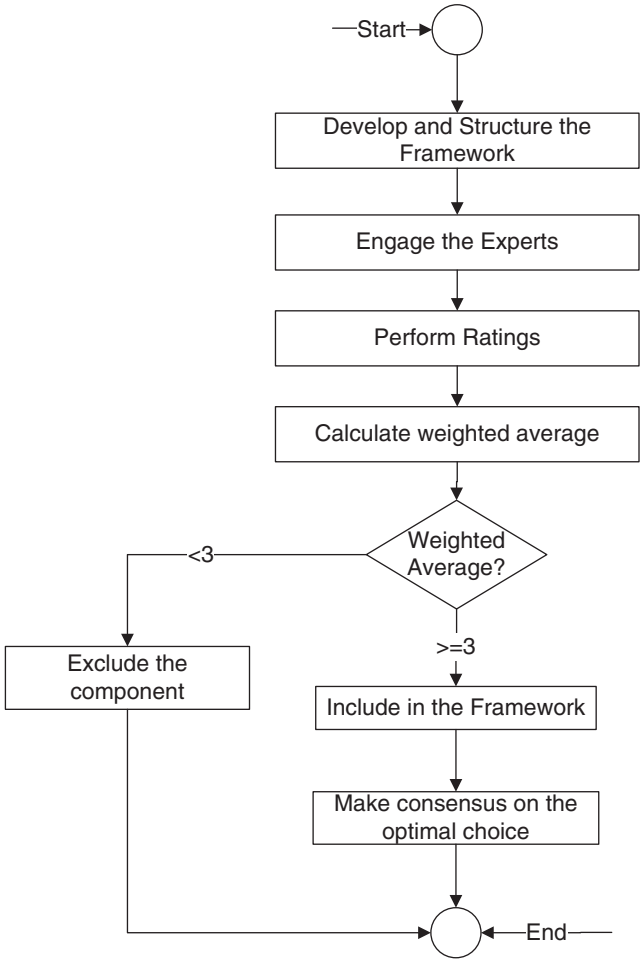


**Figure 4:** The components selection process of CCMF

### *4.2 Case Study*

CCMF is validated through an industrial case study (multiple systems) of a medium sized manufacturing organization. The organization started its operation in 1995 with about 450 employees and grew quickly to around 7000 employees. Since its inception the organization is using legacy software applications for performing different business functions. For our purpose, CCMF is applied on six different projects within the organization: Human Capital Management, Production Management, Inventory Control, Financial & Accounting, Purchase Management and Transportation Management. Before discussing the case study in detail, a glimpse of the LS based operational environment is given next.

*4.2.1 Overview of Existing Applications*

To fulfill the needs of end users working in different functional domains, 12 small LSs were handling the tasks of six projects running in the organization Fig. 5. These were sharing duplicate copies of information with tightly bound integrations. Their problems are listed next.

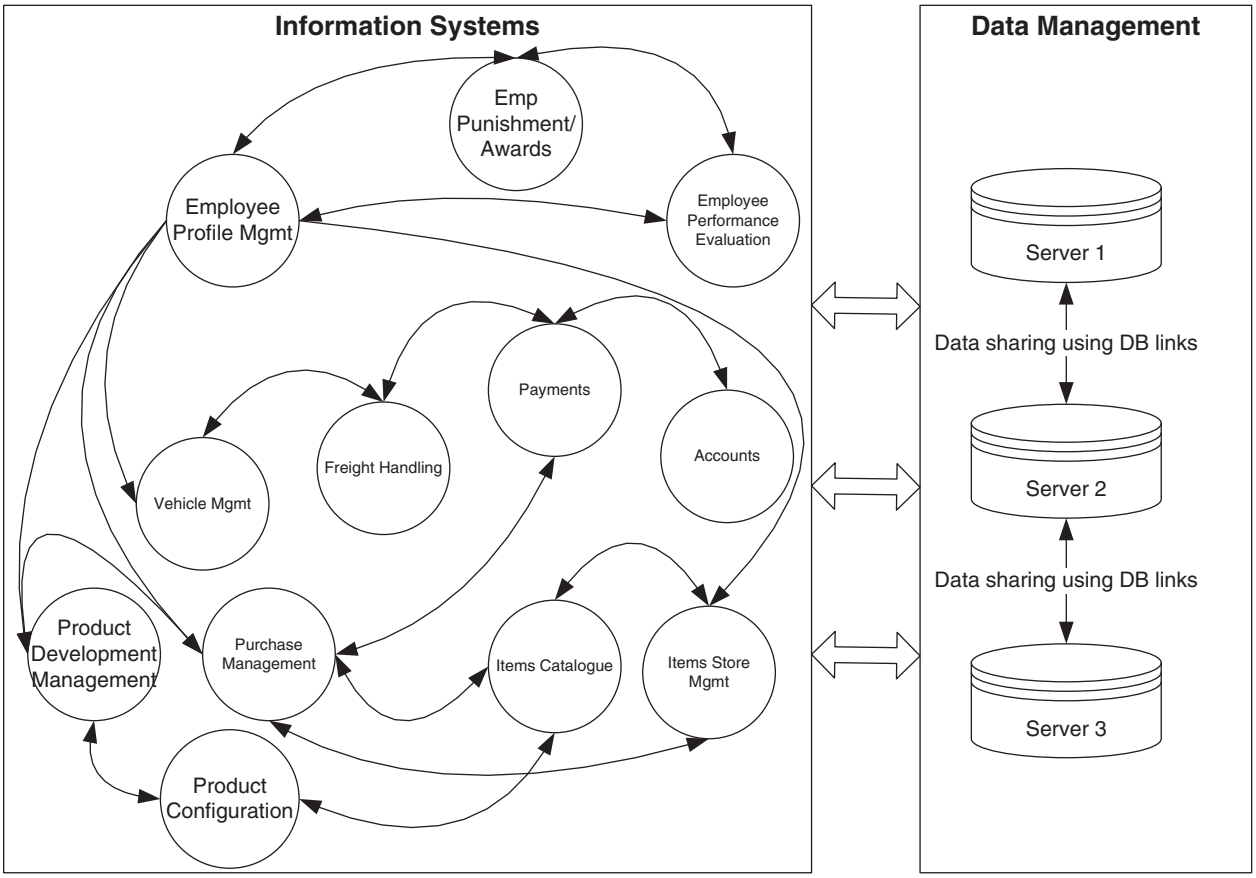


**Figure 5:** The structure of the LS before modernization

i) No centralized control
ii) Multiple servers for data management, causing data duplication and data inconsistencies
iii) No single entry point
iv) Worse security of data and software
v) Rigid design structure
vi) Low performance
vii) Incompatibility
viii) Lack of standardization
ix) High maintenance cost

*4.2.2 Implementation of CCMF*

Before applying CCMF, a detailed meeting was conducted with higher management of the enterprise to discuss the desired outcome and project's objectives. The legacy applications related to Human Capital Management were chosen as pilot candidates (core module) for modernization because these were

referenced by almost every LS in the legacy environment. Workflow management and enterprise control systems were also designed and developed in parallel. After HCM, five remaining projects were modernized. The modernization is completed in six months. The architecture of the modernized system is shown in Fig. 6. The new system has six enterprise level applications controlled by a centralized module with single server for data management.

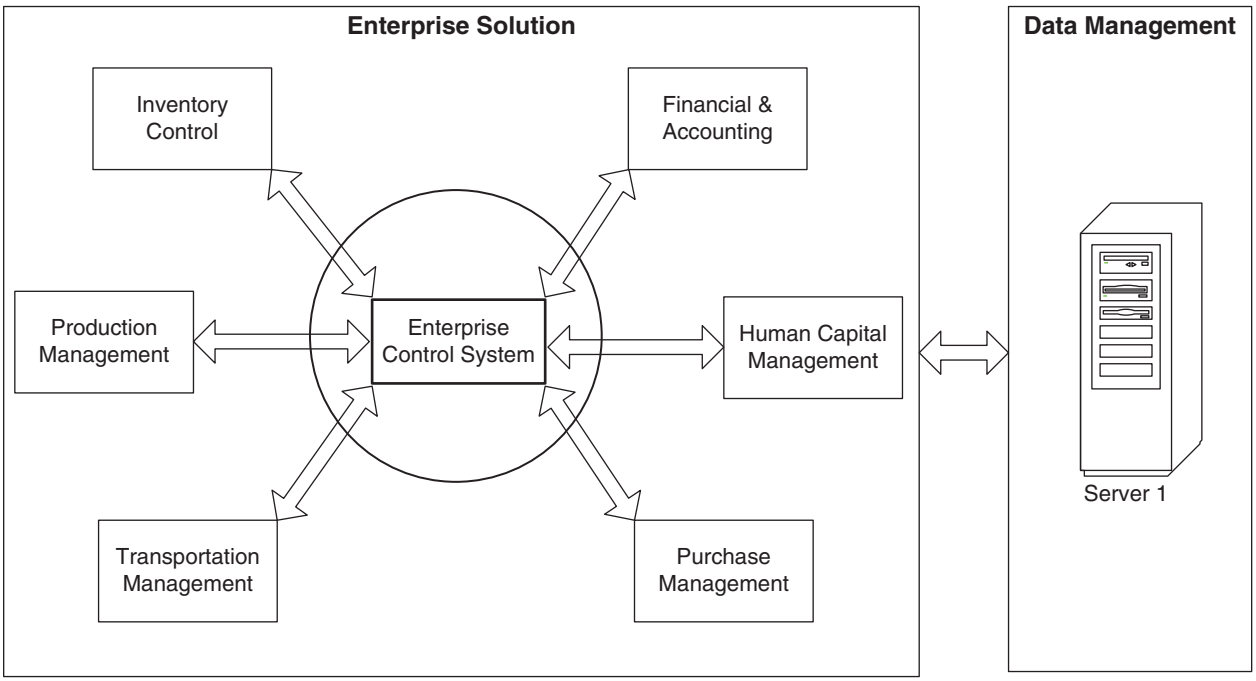


**Figure 6:** The system architecture of the proposed enterprise solution

*4.2.3 Case Study Results*

The case study was a combination of six different types of LSs, resulting in six different modernized systems. In order to validate our results more affectively we decided to consider key features of an ERP system as our criteria parameters. The features were discussed in a detailed brainstorming sessions with the same panel of experts as mentioned earlier. At the end of the sessions, four key features were selected as criteria for evaluation of our results.

As depicted in Fig. 6, the modernized system manages the main business process and integrates all important parts of business. The Enterprise Control System manages the administration and security aspects of the modernized system. The new enterprise solution fulfills the operational needs of the end users effectively and serving the top management to take better and informed decisions.

*4.2.4 Validation of Results*

An ERP system has many features like tracking and visibility, accounting, reporting and data analysis, etc. Almost any contemporary enterprise solution has the following key features:

- Usability
- Administration and Security
- Integration
- Business Intelligence

These four features are our main criteria parameters for validation. We shall validate our results by analyzing the ERP system in context of these important key features. Next we discuss validation of these features.

*Usability*

Usability is one of the key quality attribute of an ERP solution. Software usability raises the satisfaction of users [32,33,34]. Software Usability Scale (SUS) [35] is a popular method for measuring usability, with values between 0 (negative) and 100 (positive). A survey based on 10 opinions was conducted among 50 users before and after modernization of the system. Each opinion had to be judged as 1: Strongly Disagree, 2: Disagree, 3: Neutral, 4: Agree, 5: Strongly Agree. The average scores per opinion are given in Tab. 2 for both the modernized enterprise system and the LS (usage at least 5 years).

**Table 2:** The case study average scores of opinions for the Modernized System (MS), i.e., enterprise system and the LS. Text within quotes is taken from [35]

| SUS Opinion | MS score | LS score |
|---|---|---|
| 1. "I think that I would like to use this system frequently." | 4 | 2 |
| 2. "I found the system unnecessarily complex." | 3 | 4 |
| 3. "I thought the system was easy to use." | 5 | 3 |
| 4. "I would need the support of a technical person to be able to use this system." | 2 | 3 |
| 5. "I found the various functions in this system were well integrated." | 4 | 3 |
| 6. "I thought there was too much inconsistency in this system." | 3 | 5 |
| 7. "I would imagine that most people would learn to use this system quickly." | 5 | 3 |
| 8. "I found the system very cumbersome to use." | 2 | 4 |
| 9. "I felt very confident using the system." | 2 | 1 |
| 10. "I needed to learn a lot of things before I could get going with this system." | 1 | 5 |

After collecting the responses, we applied the SUS method by subtracting 1 from scores of all odd numbered opinions and subtracted from 5 the scores of all even numbered opinions. All the values were summed up and multiplied by 2.5 to get the desired usability score (US).

$$\mathrm{US} = 2.5 \times \sum_{i=0}^{10} x_i.$$

Fig. 7 shows the plots of results. Typically, a score of 68 is considered an acceptable average for usability of a product. A value below 68 is considered a usability problem. In Fig. 7, it is quite evident that the usability i.e., a key quality attribute of an ERP system was highly appreciated by the users.

*Administration and Security*

Administration and security is another important feature of ERP systems. Each ERP system is a collection of well managed and finely controlled software products ensuring accessibility and integrity of system data using enhanced security features. Tab. 3 shows a brief comparison of administration and security related features of the modernized enterprise system and the LS.

It is evident that the newly proposed enterprise solution is more secure and administered affectively. The access control in the LS was not centralized rather each legacy software module had its own access control

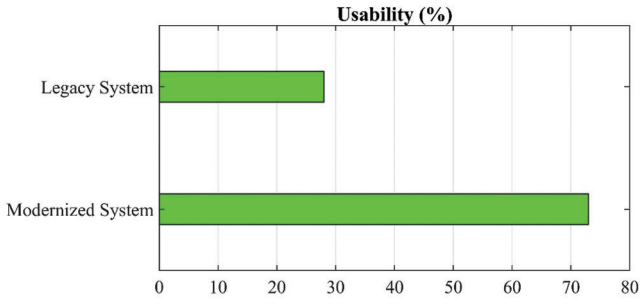


**Figure 7:** Comparison of the usability score between the legacy and modernized systems

**Table 3:** A comparison of administration and security features between the Modernized System (MS) and the LS

| Feature | Availability in the System | |
|---|---|---|
| | LS | MS |
| System Administration Control | ✗ | ✓ |
| Centralized Access Management | ✗ | ✓ |
| Role-based Access Control | ✓ | ✓ |
| Two Layer Data Audit Trail | ✗ | ✓ |
| Single Server for Data Management | ✗ | ✓ |

mechanism. Due to lack of a centralized access control, it was hard to maintain the LS and ensure its security. There was no data audit trail mechanism in the LS, while the proposed system had a comprehensive audit trail system to keep complete track of transactions taking place over time. The LS database was maintained using three database servers, and snapshots had been created using database links for sharing among various modules. Due to such architecture, there were performance issues and database administration was inefficient, time consuming and less trust worthy. The enterprise system comprises of single server to maintain the data. The integration among systems is quite manageable easily. Database schemas were managed through role based data sharing mechanism among various departments such as HRM, Purchase, etc.

*Integration*

An ERP system is a well-integrated system that can share data among various interrelated business functions and manages these functions effectively. The integration attribute was analyzed with two different aspects, i.e., programmer's view and management's view using "information independence" and "Enterprise Business Reporting" features.

*Information Independence*

In the legacy system all the systems were tightly integrated and it was very hard to make changes in the integrated information or implementation scheme. Contrary to this, in the new system the programmers found it very simple to manage the sharing of data among various software systems using standard transparent layers (utility functions). These transparent functions were so useful that even changes in the users requirement could rarely affected the underlying code being written for integration.

*Enterprise Business Reporting*

It was a common complaint by the management people that they could not receive information timely and accurately related to different business functions. The reason behind this problem was that in the LS it was quite a tough job to pick the related data from different domains. LSs had no scheme for various software modules to share data for reporting purpose. Similarly the data being received was also cumbersome to be translated into useful information as there was not standard coding scheme for data format. In the newly proposed enterprise system, the data was translated to standard format during data migration stage. The sharing of this data among different software was also made simple through standard utility functions for better reporting. With the modernized system, the management people were happy to timely receive different types of reports containing information related to different business domains.

*Business Intelligence*

An enterprise software system also provides business insight to the management that facilitates them in informed decision making. To ensure the new system provide the business intelligence feature, user's feedback was required. For this purpose a "feedback form" was circulated among the representatives from the middle and the higher management. The form contained five attributes related to the business intelligence. The Likert scale method was used to rate each attribute as 1: Poor, 2: Satisfactory, 3: Good, 4: Very Good, 5: Excellent. Results of the survey among 20 officials in both the higher and middle management are summarized in Tab. 4.

The average feedback value for the modernized enterprise system was 4.4, which is equivalent to "Very Good" rating. The average rating for LS was 2, which is "Satisfactory." Thus the average business gain of the MS is clearly better than that of the LS.

**Table 4:** A comparison of MS and the LS in context of business intelligence feature

| Business Gain | LS | MS |
|---|---|---|
| Business insight | 2 | 4 |
| Decision Making | 1 | 5 |
| ROI | 2 | 4 |
| Reporting | 3 | 5 |
| Data Quality | 2 | 4 |
| Average Rating | 2 | 4.4 |

## 5 Compliance to CMMI Framework

Our proposed framework has very close compliance to CMMI. In order to prove our claim and ensure the quality and reliability of our proposed framework, we compared our work with the CMMI framework (CMMI-DEV, V1.3). CMMI is a process improvement framework for software industry developed by Carnegie Mellon University. CMMI constitutes of five levels of maturity, naming Initial (Level 1), Managed (Level 2), Defined (Level 3), Quantitatively Managed (Level 4) and Optimizing (Level 5) as well as contains 22 process areas. CMMI framework is extensively being adopted by large scale software businesses. Our proposed framework is well compliant with the Specific Goals (SGs) mentioned in the Process Areas (PAs) of CMMI Level 2. The KPs at each stage of our modernization roadmap have close compliance with the specific practices (SPs) mentioned under the relevant SGs.

Tab. 5 displays complete details of our framework's compliance with CMMI Level 2. Column names used in Tab. 5 are explained as:

MNA (EO-I): Well documented requirements approved by the relevant stakeholders

MP (EO-II): A detailed Project Plan containing all the necessary details required for project execution

A&D (EO-III): To establish and maintain a detailed design document based on requirements specifications

**Table 5:** A compliance chart for mapping CCMF components to CMMI

| CMMI Framework (Level-2) | | CCMF | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Process Area | Specific Goal (SG) | MNA (EO-I) | MP (EO-II) | A&D (EO-III) | SI (EO-IV) | DM (EO-V) | LV (EO-VI) | V&V (EO-VII) | SR (EO-VIII) |
| Requirements Management (RM) | SG 1 Manage requirements | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ |
| Project Planning (PP) | SG 1 Establish estimates | | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| | SG 2 Develop a project plan | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| | SG 3 Obtain plan commitment | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Configuration Management | SG 1 Establish baselines | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| | SG 2 Track and control changes | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| | SG 3 Establish integrity | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |
| Measurement and Analysis | SG 1 Align measurement and analysis activities | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |
| | SG 2 Provide measurement results | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Process and Product Quality Assurance | SG 1 Objectively evaluate processes and work products | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | SG 2 Provide objective insight | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ | |
| Project Monitoring and Control | SG 1 Monitor project against plan | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ |
| | SG 2 Manage corrective action to closure | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ |
| Supplier Agreement Management | SG 1 Establish supplier agreement | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |
| | SG 2 Satisfy supplier agreement | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |

SI (EO-IV): Core modernized software systems align with the design specifications

DM (EO-V): Quality Data Migration

LV (EO-VI): Seamlessly operational legacy environment

V&V (EO-VII): A Ready-To-Use Core Modernized System

SR (EO-VIII): A reliable Modernized Enterprise Software Solution

We have compared each EO of CCMF with Specific Goals (SG) of each Key Process Area (KPA) of CMMI maturity Level 2. Results show that the EO at each stage of CCMF is well compliant with the SG of each KPA of CMMI Level 2. Tab. 5 was again presented to the same panel of experts for review. All the experts analyzed and objectively reviewed the framework's compliance in detail, along with the KPs mentioned under MGs at each stage of the framework. The panel of experts found that CCMF fully compliant with the CMMI framework.

## 6 Conclusion

Modernization of LS has a tall order. It involves the transformation of architecture, design, data and the user experience to a new system paradigm. This modernization process becomes more challenging when the expected outcome is an enterprise solution. This work is about modernization of LSs to new enterprise system. We tackled RQ1 by proposing a modernization roadmap of CCMF. It consists of eight distinct stages namely Modernization Need Analysis, Modernization Planning, Architecture and Design, Solution Implementation, Data Migration, Legacy Virtualization, Verification and Validation and System Retirement. By following these stages, LS can be transformed to an enterprise solution. CCMF caters to all issues of LSs, leading to a system solution that best fits the desired expectations of stakeholders. Effectiveness of CCMF is established by the fact that it has successfully transformed six legacy projects to their modernized counterparts of different nature. RQ2 is addressed by making CCMF compliant to CMMI framework. CCMF is enriched with quality methods and techniques, so that the modernized system ensures improved quality and enhanced reliability. CMMI framework Level 2 benchmark is used to match the EOs at each stage of the modernization roadmap with SGs of respective PA at Level 2 using compliance chart. All the EOs are found compliant to SGs, which clearly indicating that CCMF is fully compliant to CMMI framework. To the best of our knowledge, no modernization approach is based on quality practices for compliance to CMMI framework. RQ3 is probed by conducting six projects of different nature in our case study. We have observed that the modernized systems achieved high values for key quality attributes, usability, security, administration and integration. Repetitively achieving enhanced quality attribute values for all the modernized systems, testifies the quality of our approach. Basically adhering to quality standards as specified by CMMI has led to our promising results. The key aspect of CCMF is the strategy based on quality KPs that leverages effective implementation of the modernization framework. The reliability concern of the resultant software is tested and verified by applying CCMF to multiple LSs in our case study. CCMF primarily focuses on the adoption of process driven approach. During the transition, the system is kept operationally available. The uninterrupted continuation of user's operations during the transition has been an important objective of this framework. This holds even for the implementation phase. As compared to other available frameworks, the compliance feature of CCMF to a quality process improvement framework (CMMI Framework Level-2) makes not only it a viable solution for enterprises but also ensures the delivery of reliable system.

**Funding Statement:** This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2016-0-00313) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] B. Y. Alkazemi, M. K. Nour and A. Q. Meelud, "Towards a framework to assess legacy systems," in *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics, SMC 2013*, Manchester, pp. 924–928, 2013.

[2] S. M. Hussain, S. N. Bhatti and M. F. U. Rasool, "Legacy system and ways of its evolution," in *Proc. Int. Conf. on Communication Technologies 2017*, Rawalpindi, pp. 56–59, 2017.

[3] R. Khadka, B. V. Batlajery, A. M. Saeidi, S. Jansen and J. Hage, "How do professionals perceive legacy systems and software modernization?," in *Proc. Int. Conf. on Software Engineering*, Hyderabad, pp. 36–47, 2014.

[4] N. Y. Conteh and M. Jalil Akhtar, "Implementation challenges of an enterprise system and its advantages over legacy systems," *International Journal on Computer Science and Engineering*, vol. 7, no. 11, pp. 120, 2015.

[5] R. Khadka, A. Saeidi, S. Jansen, J. Hage and G. P. Haas, "Migrating a large scale legacy application to SOA: Challenges and lessons learned," in *Proc. Working Conf. on Reverse Engineering*, Koblenz, pp. 425–432, 2013.

[6] K. Bennett, "Legacy systems: Coping with success," *IEEE Software*, vol. 12, no. 1, pp. 19–23, 1995.

[7] J. Bisbal, D. Lawless, B. Wu and J. Grimson, "Legacy information systems: Issues and directions," *IEEE Software*, vol. 16, pp. 103–111, 1999.

[8] F. Fui, H. Nah, S. Delgado and F. Fui-Hoon Nah, "Critical success factors for enterprise resource planning implementation and upgrade," *Journal of Computer Information Systems*, vol. 46, no. 5, pp. 99–113, 2006.

[9] B. Paradauskas and A. Laurikaitis, "Business knowledge extraction from legacy information systems," *Information Technology and Control*, vol. 35, no. 3, pp. 214–221, 2006.

[10] N. H. Weiderman, J. K. Bergey, D. B. Smith and S. R. Tilley, "Approaches to legacy system evolution," Technical Report: Carnegie-Mellon University, 1–42, 1997. [Online]. Available: https://resources.sei.cmu.edu/asset_files/TechnicalReport/1998_005_001_16601.pdf.

[11] S. Comella-Dorda, K. Wallnau, R. Seacord and J. Robert, "A survey of legacy system modernization approaches," *Technical Report: Carnegie-Mellon University*, 2000. [Online]. Available: https://apps.dtic.mil/sti/pdfs/ADA377453.pdf.

[12] E. De Vargas Agilar, R. B. De Almeida and E. D. Canedo, "A systematic mapping study on legacy system modernization," In *Proc. of the Int. Conf. on Software Engineering and Knowledge Engineering 2016*, Redwood, pp. 345–350, 2016.

[13] J. Bergey, J. D. Smith, S. Tilley, N. Weiderman and S. Woods, "Why Reengineering Projects Fail," (No. CMU/SEI-99-TR-010). CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST. [Online]. Available: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=13405.

[14] A. Vijaya and N. Venkataraman, "Modernizing legacy systems: A re-engineering approach," *International Journal of Web Portals*, vol. 10, no. 2, pp. 50–60, 2018.

[15] B. Althani and S. Khaddaj, "The applicability of system migration life cycle (SMLC) framework," in *Proc. Int. Sym. on Distributed Computing and Applications to Business*, Anyang, pp. 141–144, 2017.

[16] H. Bruneliere, J. Cabot, F. Jouault and F. Madiot, "MoDisco: A generic and extensible framework for model driven reverse engineering," in *Proc. IEEE/ACM Int. Conf. on Automated Software Engineering*, Antwerp, pp. 173–174, 2010.

[17] E. Georgiadou, "A holistic method for improving software product and process quality," Ph.D. dissertation. Middlesex University, 2019.

[18] M. Khan, W. Nisar, E. U. Munir, W. Anwar and I. Ali, "Deployment strategies for a reengineered information system in context of legacy system," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4, no. 3, pp. 178–185, 2012.

[19] B. Shahzad and Y. Al-Ohali, "Trivial model for mitigation of risks in software development life cycle," *International Journal of Physical Sciences*, vol. 6, no. 8, pp. 2072–2082, 2011.

[20] L. Mu and C. K. Kwong, "A multi-objective optimization model of component selection in enterprise information system integration," *Computers & Industrial Engineering*, vol. 115, no. 2018, pp. 278–289, 2018.

[21] T. C. Fanelli, S. C. Scott and S. Banerjee, "A Systematic Framework for Modernizing Legacy Application Systems," in *Proc. IEEE Int. Conf. on Software Analysis, Evolution, and Re-engineering*, Suita, pp. 678–682, 2016.

[22] R. Heckel, R. Correia, C. Matos, M. El-Ramly, G. Koutsoukos *et al., Architectural Transformations: From Legacy to Three-tier and Services*. *Software Evolution*, Berlin Heidelberg: Springer, pp. 139–170, 2008.

[23] A. Bianchi, D. Caivano, V. Marengo and G. Visaggio, "Iterative reengineering of legacy systems," *IEEE Transactions on Software Engineering*, vol. 29, no. 3, pp. 225–241, 2003.

[24] R. Pérez-Castillo, I. G. De Guzman and M. Piattini, "Knowledge discovery metamodel-ISO/IEC 19506: A standard to modernize legacy systems," *Computer Standards & Interfaces*, vol. 33, no. 6, pp. 519–532, 2011.

[25] J. Delapeyronnie, P. H. Newcomb, V. Morillo, F. Trimech, L. Nguyen *et al.,* "Modernization of the EuroCAT air traffic management system (EATMS)," in *Proc. Information Systems Transformation*, Morgan Kaufmann, Amsterdam, pp. 91–131, 2010.

[26] H. Sneed and C. Verhoef, "Re-implementing a legacy system," *Journal of Systems and Software*, vol. 155, pp. 162–184, 2019.

[27] K. Garcés, R. Casallas, C. Álvarez, E. Sandoval, A. Salamanca *et al.,* "White-box modernization of legacy applications: The oracle forms case study," *Computer Standards & Interfaces*, vol. 57, pp. 110–122, 2018.

[28] C. C. Chiang and C. Bayrak, "Legacy software modernization," in *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics*, Taipei, vol. 2, pp. 1304–1309, 2006.

[29] J. C. Deprez and S. Alexandre, "Comparing Assessment Methodologies for Free/Open Comparing Assessment Methodologies for Free/Open Source Software : OpenBRR and QSOS," in *Proc. Int. Conf. on Product Focused Software Process Improvement*, Berlin: Springer, pp. 189–203, 2008.

[30] J. J. Diaz, J. Garbajosa and J. A. Calvo-Manzano, "Mapping CMMI Level 2 to Scrum Practices: An Experience Report," in *Proc. European Conference on Software Process Improvement*, Alcala (Madrid), pp. 93–104, 2009.

[31] A. R. Hevner and S. Chatterjee, "Design Research in Information Systems, Intergrated Series," in *The Design Research in Information Systems, Intergrated Series*, Boston, MA: Springer, pp. 9–22, 2010.

[32] B. Shahzad, B. I. Ullah and N. Khan, "Software risk identification and mitigation in incremental model," in *Proc. IEEE Int. Conf. on Information and Multimedia Technology*, Jeju Island, pp. 366–370, 2009.

[33] M. Shafiq, M. Ahmad and J. G. Choi, "Public system usability analysis for the valuation of cognitive burden and interface standardization: A case study of cross-ATM design," *Journal of Organizational Computing and Electronic Commerce*, vol. 27, no. 2, pp. 162–196, 2017.

[34] I. H. Mathkour, B. Shahzad and S. Al-Wakeel, "Software risk management and avoidance strategy," in *Proc. Int. Conf. on Machine Learning and Computing*, Singapore, pp. 477–481, 2011.

[35] J. Brooke, *SUS-A Quick and Dirty Usability Scale*. *The Usability Evaluation in Industry*, Redhatch Consulting Ltd., Earley, Reading, Uk, 189, 1996.