Logical Framework of Information Technology: Systematization of Software Development Research

Dian Sa'adillah Maylawati and Muhammad Ali Ramdhani

Abstract—This article aims to present a comprehensive Logical Framework for Information Technology (IT) Research, specifically for developing customized IT applications or software. The methodology of writing this article uses a content analysis with the main source of literature review, Focus Discussion Group, and also based on the experience and knowledge of the authors as lecturers of Software Engineering and Software Project Management. This article shows that although current IT development approaches or methodologies (especially software development methodology) continue to develop, good IT design is carried out through six main stages, namely planning, analysis, design, construction, implementation, and maintenance. The success of IT implementation depends on the good process of all stages of IT design. The involvement of all actors/stakeholders in IT design is essential to be accommodated at all stages of IT design. Quality also becomes the main goal and controls every process of IT development.

Keywords—information technology, logical framework, software development, software engineering, software measurement, software metrics, software project, software quality.

I. INTRODUCTION

The development of human civilization has led most people to feel that information is one of the basic needs besides the main need such as clothing, food, and shelter. Changes in the rapid, dynamic, and broad environment are triggered by the advancement of information technology (IT) in all fields. This has encouraged the transformation of civilization into an information society [1]. It has become necessary for the information community to utilize IT as a medium of communication and information exchange to obtain fast and accurate information [2].

Utilization of IT is one of the primary keys if individuals or organizations want to improve the quality of service, control the utilization of resources efficiently and effectively, and correct in making strategic and operational decisions [3]–[5]. Proper IT use in organizations will lead to competitive and comparative advantages [6]–[8]. The information generated from data processing by IT must not only be accurate and fast but also consider its relevance to the ability of user needs.

A mismatch between IT and users' capabilities and needs makes the technology unable to be utilized properly. The facts show that in 2015, the evaluation results of 1,800 software, 37% of the software was wasted [9]. Even the highest percentage of unused software is in education, which is 47%. Wasted software is caused by several factors, such as not meeting the needs of users; there are software errors, faults, and failures; software quality is not fulfilled; no innovation; does not apply the concept of Human and Computer Interaction properly; difficult to use; not according to market needs (not up-to-date); the lack of understanding of the use of technology due to its rapid development so that trends cannot be followed.

There are many studies in information technology, most of them focus on one case study that implements IT in various fields, such as in education [10], [11], government [12], [13], health [14], tourism [15], finance [16], and so on. On the other hand, there are several research purposes for possible future research on IT [17]. However, the failure of IT utilization indicates the need to involve user preferences as part of technology design from the planning stage. Creating an IT design framework can provide intense communication between developers and users (prospective users). Therefore, there is a need for a comprehensive framework as an essential foundation that can guide IT research. This article aims to create a logical framework for IT design research based on an agreement between the developer and the user/prospective user.

II. METHODOLOGY

This article uses the main source of analysis from the literature review. Logical framework testing is carried out in the learning process, which involves discussions with students taking courses in Software Engineering, Software Project Management, and Software Projects at the Department of Informatics, Universitas Islam Negeri (UIN) Sunan Gunung Djati Bandung, Indonesia. The validity of the material presented in this article was carried out through the Focus Group Discussion (FGD) as one qualitative research technique with a structured discussion of a small group of people to generate qualitative data on a precise topic of interest, using a set of open-ended questions [18], [19]. This study conducts FGD of 15 experts in the study of Information Systems and Software Engineering scientific group. They worked as lecturers at the Department of Informatics, UIN Sunan Gunung Djati Bandung, Indonesia.
III. RESULT AND DISCUSSION

The subject of this article is the logical framework on information technology research. In some literature, sometimes information technology (IT) and information systems (IS) are exchanged similarly. This article defines IS as a combination of business processes, work procedures, information, people, and technology that facilitates activities to achieve organizational goals [20]. IS is a collection of components in an organization related to the process of creating and flowing information [21]. Whereas IT is defined as the technology used to process data, including processing, obtaining, compiling, storing, and manipulating data in various ways to produce quality information that is relevant, accurate, and timely information, which can be used for personal, business, and governance, and the strategic information for decision making [22].

In general, IT development and design are divided into three major groups. The first group is the development and design of IT infrastructure, including stand-alone computer installation development of LAN (Local Area Network) and WAN (Wide Area Network) network infrastructure. The second group is designing IT applications with general uses, in the form of application packages that can be purchased on the market, such as Microsoft retail products to integrated applications based on Enterprise Resource Planning (ERP) such as SAP (System Analysis and Program Networking), Oracle, Baan, and PeopleSoft. The third group is the planning and development of specifically designed (customized software) applications for the unique needs of individuals or organizations, with the developer carried out by internal organizations or by working with outside parties such as consultants and software houses.

Apart from the broader categorization of developer groups, IT development and design generally go through six stages which are used as stages in carrying out IT development activities, namely: planning, analysis, design, construction, implementation, and maintenance. This article is directed at planning and developing customized IT applications. This stage does not refer to any particular software development model or methodology, such as Waterfall [23], Iterative [24], Spiral [25], Prototype [26], Rational Unified Process [27], Rapid Application Development [28], Incremental [29], and Agile methodologies like Scrum [30], [31], Lean [32], [33], Crystal [34], Extreme Programming [35], and so on. Because, in general, both the "conventional" and Agile methodologies have the six main processes: Planning, Analysis, Design, Construction, Implementation, and Maintenance.

The logical framework proposed in this study (provided in Fig. 1) is an analysis of the system design requirements based on the needs of the problems that will be solved by IT implementation. The logical framework is prepared based on an agreement between the developer and the prospective user. For general IT design, the needs of prospective users can be defined with a survey to understand the business processes that will be accommodated in IT.

IT research is the process of identifying a portfolio of computer-based IT applications that will support organizations in implementing business plans to realize their business goals. An important factor in the IT planning process is an analysis to minimize the risk of failure, ensure the involvement of all interested parties, and emphasize the desired goals. The input in preparing this strategic plan are internal business environment, external environment, internal IT environment, and external IT environment [36]. The authors propose an analysis of IT development planning, beginning with a study of the needs of IT applications by paying attention to aspects of the instrumental and strategic environment based on a systems approach. There are found three essential mindsets in designing and building solutions to problems based on a systems approach, namely: 1) cybernetic, goal-oriented; 2) holistic, which is a complete perspective on the system and 3) effective, namely, principles that are more concerned with operational results and can be implemented rather than a theoretical view to achieving decision efficiency [37].

A. The Framework of the Planning Stage

The development of the knowledge economy era has made IT a competitive advantage and is an effective way to help organizations survive in modern society and gain more significant benefits. However, the success rate of IT implementation is not always exemplary. In some organizations that have failed in implementing IT, most of them did not make a systematic and scientific IT strategy planning at an early stage. The absence of IT strategic planning makes IT utilization unclear and valuation uncertain in the technical selection criteria. Conversely, successful organizations have good planning [38]. Two parties must be directly involved in IT planning: those in need (demand side) and IT developers (supply side). The first party is an individual or organization that has a problem that requires the existence of IT to improve its performance. The second party is an individual or organization that answers the first party's issues, which is realized by IT design.

Based on an IT development project management review, the output from the planning stage is the use of resources and design procedures for the other five stages in
IT design (analysis, design, construction, implementation, and maintenance). Planning that is compiled at a minimum must contain elements of drafting schedules, standards and procedures, utilization of resources, personnel needs and assignments, and financial aspects (Fig. 2).

Fig. 2. Planning on IT Development.

B. The Framework of the Analysis Stage

Analysis of IT research is an activity to define problems and formulate solutions to problems through IT implementation. In general, the focus of analysis on IT research considers two main aspects, namely aspects of management and technology (Fig. 3). Analysis of management aspects includes historical aspects, vision, mission, critical success factors, performance measurements, strategies, activities, and other matters relating to aspects of management or business processes. Analysis of technological aspects of activities to evaluate IT assets owned by organizations, study IT infrastructure, the effectiveness and reliability of IT use and analyze the possibility of adding systems (system upgrading) as a consequence of the proposed IT implementation.

Fig. 3. Analysis of IT Development.

The output of the analysis process on management and technology aspects in the form of problem statements containing important issues that must be dealt with IT. It contains an analysis of the causes of the problem, its impact on management aspects, several possible scenarios for solving risks (cost and benefit) and trade risk, and the recommended solution choices. Proposed problem solving can be done by reviewing the existing system and its weaknesses, opportunities for the development of existing systems, and the overall concept related to the design of the system to be built. In academic culture, this stage is usually called state of the art.

C. The Framework of the Design Stage

The design stage is the stage of making a design (sketch) of the system requirements as a clear picture of the IT application that will be made to the user (such as provided in Fig. 4) [39]. The design stage aims to describe the design of IT applications so that the structures and operations created can be easily understood and the procedures are easy to follow. In addition, to meet users' needs, a clear picture of the system should be made and its implementation [40].

Determination of system architecture design includes hardware and software that will be used in system development, interface design (navigation methods and menus provided), and equations to be used by the system. The success of IT applications is influenced by the way the technology presents itself to the user, so the user interface needs to be considered [41].

Fig. 4. IT Development Design.

Database system design includes the data to be used and the storage and data recall system in question. An IT application, like any other equipment, is a subject that will experience failures from various causes, such as disk crashes, power failures, software errors, computer room fires, and sabotage. In the design stage, system recovery needs to be prepared. Every failure can cause data loss. The database system must ensure that transactions' atomicity and durability can be maintained. System recovery must have a high capability to minimize data loss due to crashes [42]. Program (algorithm) design that defines the functions of the program that must be made, the logic of data processing, system control, as well as the input-output activities of the system.

At the software design stage, there are two modeling approaches: structured modeling and object-oriented modeling [43], [44]. Structured modeling usually uses Context Diagrams and Data Flow Diagrams (DFD), complete with Process Specifications and Data Dictionary. Whereas object-oriented design usually uses UML (Unified Modeling Language) modeling [45], where there are eight basic diagrams that should exist in UML modeling, including use case diagrams, class diagrams, sequence diagrams, activity diagrams, state diagrams, collaboration diagrams, state diagrams, and component diagrams.

Both structured and object modeling are better equipped with data modeling using Conceptual Data Model (CDM), Logical Data Model (LDM), and Physical Data Model.
(PDM). CDM only contains relations between entities complete with their cardinalities. LDM contains relationships between entities complete with their cardinalities and has defined every attribute owned by these entities, including attributes marked as primary keys. LDM is also better known as Entity Relationship Diagram (ERD). Whereas PDM, better known as a relational scheme or relational table, contains an LDM mapping ready to be implemented as a database, where the PDM is already equipped with a foreign key that appears due to cardinality rules [46].

DFD describes the system as a relation between functions that relate to each other with the data flow and data storage. As an analytical device, DFD can only model the system from one perspective, namely the function represented by the bubble process [40]. At the same time, ERD is a relation diagram that uses an abstract arrangement of data stored in the system. The purpose of ERD is to show data objects and relationships that exist on the objects or entities. In addition, this ERD model is one of the tools for database design [40].

For UML modeling, an activity diagram is done on a system and describes the running system activity [47]. The use case diagram is a series or description of a group of interrelated functions and forms a system regularly carried out or supervised by an actor. Use case diagrams are graphical depictions of some or all actors, use cases, and interactions between them that introduce a system. Use case diagrams do not explain the use of use cases in detail but only briefly describe the relationship between use cases, actors, and systems. In this use case will be known functions on the system created [47]. Sequence diagrams are a description of interactions between a number of objects in a time sequence. Sequence Diagrams depict dynamic collaboration between a number of objects. Its purpose is to indicate the sequence of messages sent between objects as well as interactions between objects. Something that happens at a certain point in the system execution [40]. Class diagrams are collections of objects with a general structure, behavior, relations, and semantic/common words. Classes are determined by finding objects in sequence diagrams and collaboration diagrams [48].

When using different design methodologies, different explanatory tools can design IT applications. The main output from the design stage is a technical and management system design blueprint that is used as a guide in the construction process and implementation of components in IT applications to be developed.

D. The framework of the Construction Stage

The construction phase transforms system design into an IT application system that computer devices can run. Construction consists of two parts, namely coding and testing (to measure the software quality described in Fig. 5) [49]. Coding is the writing or translation of a set of computer instructions and data definitions as outlined in a form understandable from the human and computer sides [50]. The next stage is testing. Testing is a stage to ensure that the system is designed to function properly according to user needs.

One mechanism for testing IT applications can be done with the Software Quality Assurance model. Software Quality Assurance is a systematic approach to evaluating quality, product standards, processes, and procedures in software. One of the standards relating to software quality is ISO/IEC 9126. ISO/IEC 9126 is a standard made by the International Standardization Organization regarding the quality of software products. Characteristics related to software quality include six categories: functionality, efficiency, reliability, usability, maintainability, and portability [51].

Besides ISO, several quality factors can be achieved in software development. These quality factors include the McCall Software quality factor, which is the basis of software quality development. The McCall software quality factor has three main groups, with 11 quality factors in it, among others [52]: product operations (correctness, reliability, usability, integrity, and efficiency), product revision (maintainability, flexibility, and testability), and product transition (portability, reusability, and interoperability). Besides McCall, there are also software quality factors such as Deutsch and Willis, which have 12-15 quality factors [53], Evans and Marciniaik [54], Boehm [55], IBM Software Quality, FURPS [56], Dromey [57], and many more. Even today, there is the development of quality factor software that is tailored to Agile or rapid software development [58]–[60].

To achieve software quality factors, appropriate measurements and metrics are needed. Measurements and metrics can be used to measure products, people, and processes in software development [61], [62]. The simplest metric to use is to calculate the Line of Code (LoC) from the program code, but this metric is very dependent on the programming language used and the programmer's programming style. For example, the software that is built wants to achieve the quality of flexibility, so in achieving flexibility, the modularity sub-factor must be met. Modularity can be measured by measuring software module/class cohesion and coupling. The metric that can be used is Lack Cohesion on Method (LCOM) from the program code, but this metric is very dependent on the programming language used and the programmer's programming style. For example, the software that is built wants to achieve the quality of flexibility, so in achieving flexibility, the modularity sub-factor must be met. Modularity can be measured by measuring software module/class cohesion and coupling. The metric that can be used is Lack Cohesion on Method (LCOM) to measure cohesion [63], and Coupling between Object (CBO) to measure object coupling [64]. The McCabe cyclometric complexity metrics developed from graph theory can also be used to measure software complexity [65].
E. The Framework of the Implementation Stage

The system implementation phase is a procedure performed to complete the design of existing systems in the new system design document. In general, the purpose of the implementation phase is to implement the IT development that has been made [66], where an IT application is ready to operate [67]. Activities undertaken in the implementation include (available in Fig. 6): Socialization is an activity to introduce applications to end-users; Training is carried out by providing experience in using IT applications to increase knowledge and skills for end-users with manual books and training module [68]; Supporting system in the form of help-desk for the initial implementation of IT applications; Evaluation is helpful for improving the work patterns of end-users; and Control by the user is to create rules/policies that bind end-users to use IT platforms.

![Fig. 6. IT Implementation.](image)

F. The Framework of the Maintenance Stage

The maintenance phase is a series of activities to keep IT applications running and properly help business processes. Activities undertaken during the maintenance phase include (available in Fig. 7) maintenance, recovery, modification, and upgrading.

![Fig. 7. IT Maintenance.](image)

G. Evaluation of the Proposed Framework

The evaluation process of IT technology to systematization of software engineering research as the proposed framework was conducted by involving 27 experts and researchers in the field of information technology, information system, or software engineering. There are eight main questions to evaluate completeness of the proposed framework components in general and each part of the proposed framework. The completeness of the proposed framework is evaluated with a Likert scale (value 1 to 5), with gradations from Very Incomplete (VI), Incomplete (I), Quite Complete (QC, in borderline), Complete (C), and Very Complete (VC). Further analysis was conducted in the form of a qualitative evaluation that provides comments and suggestions on the proposed framework.

In general, the proposed framework has good completeness of components. Fig. 8 shows the evaluation result for the proposed framework in general. Based on the evaluation result, evaluators think that the proposed framework components 7.4% are quite complete, 51.9% are complete, and 40.7% are complete. Most evaluators comment that the proposed frameworks are good, easy to understand, easy to follow, effective, and comprehensive to guide IT research systematically, especially in software engineering research. The components or elements in the proposed framework are quite complete, including a feedback mechanism and solution improvement. However, it is also necessary to consider several aspects such as the point of view in IT design, such as technology-driven and human-driven, measuring effectiveness and efficiency of IT research, and there needs to be synchronized the aspect of all IT strategies used.

![Fig. 8. The general evaluation result of the proposed framework.](image)

The planning stage of the proposed framework has an evaluation result of 3.7%, 55.6%, and 40.7% of quite complete, complete, and very complete components. Most evaluators think that planning stage has contained requirements, criteria or components that must be planned in IT research, such as time, costs, and procedures that must be carefully planned. It is also necessary to consider the scopes, limits and qualities that must be achieved for the IT research.

In the analysis stage, 7.4% of respondents think that the analysis stage of the proposed framework is quite complete, 37% are complete, and 55.6% are very complete. Most of
research can be tested and evaluated in a real case so that its function and usefulness can be measured in information technology research.

REFERENCES


D. Sa’adillah Maylawati and M. A. Ramdhani: Logical Framework of IT: Software Development Research 31


