

Lean IT Transformation Plan for Information Systems Development

Muhammad K. A. Kiram¹

¹TNB Research Sdn. Bhd.
43000 Kajang, Selangor, Malaysia

Maryati Mohd Yusof²

²Faculty of Information Science and Technology,
Universiti Kebangsaan Malaysia
Bangi, Malaysia

Abstract—Information systems development (ISD) is prone to failure, which can be defined as a time-consuming and costly phenomenon that provides value that is not directly appealing to clients. While ISD can be enhanced using various tools, models, and frameworks, failures related to ISD remain to be evident and costly. These failures are related to human, organizational, and technological factors and waste in ISD. This study identifies the information system (IS) success criteria and factors that contribute to ISD waste. A qualitative case study was conducted for an ICT research unit by using interview, observation, and document analysis techniques as a means of analyzing the IS success criteria, leanness level, and waste. Findings show that lean IT approaches and IS success criteria can be combined to develop a holistic transformation plan for organizational ISD. This transformation plan can potentially assist IS developers deliver high-value IS while driving organizational growth towards the fourth industrial revolution.

Keywords—Failure; lean IT; information systems; information systems development; socio-technical; waste

I. INTRODUCTION

Information systems development (ISD) has emerged with cutting edge technologies and dynamic business processes as a means of supporting organizational performance. However, an ISD is prone to failures in terms of project overruns, over-budgeting, and unfulfilled user requirements [1]. An ISD is more likely to generate waste if it is not planned and managed appropriately [2]. In addition, the collaborative, complex, and subjective nature of ISD renders it vulnerable to producing waste [3,4], such as partially done work, unnecessary processes, extra features, handoffs, delays, and motions and errors, despite the introduction of various methodologies and tools [5]. Therefore, evaluating the IS success factors and identifying their root cause of waste in ISD by using the quality improvement approach, such as lean IT, are important in enhancing the quality of the IS. ISD interacts with three socio-technical factors (organization, technology, and human) in producing IS [6]. Despite the potential of IS to increase organizational effectiveness and efficiency, IS failure remains to be a widespread concern [7] in terms of the technical, project management, quality, and human aspects [4].

IS failure is commonly associated with various socio-technical factors and ISD waste. Inaccurate development of product features, poor work management, rework, unnecessary problem solving, cognitive over-burden, psychological stress, delays, relearning, and poor communication are examples of

waste identified in ISD [4]. The longer the delay of the waste removal, the higher the cost to be incurred by an organization [8]. Consequently, this study attempts to identify the root cause of ISD waste in the quest of producing high-quality IS, in which the lean IT method and an evaluation are both carried out to achieve IS success/efficiency from the socio-technical perspective.

The paper is organized as follows. The following section provides an overview of IS success criteria and the Lean IT approach. Section 3 describes the case study method, followed by the case study findings that are presented in Section 4 based on ISD success criteria, Leanness level, waste, and document analysis. Section 5 discusses the findings based on their subsections and the Transformation Plan for lean IT in ISD organization. The final section concludes the study.

II. RELATED WORKS

A. IS Success Criteria

Despite its benefits, IS is also costly and risky. Therefore, IS developers need to understand the critical success factors of IS [9]. From the technical perspective, unclear and complex requirement specifications can affect the planning phase, rippling further through the ISD activities [10,11] in software development lifecycle (SDLC) phases [12]. Different perceptions and mental models among clients and IS developers during requirement elicitation and design can create conflict and subsequently weaken the teamwork spirit [13,14]. Limited technology expertise can also contribute to failure, in which the IS cannot meet the required features [4,13]. Furthermore, limited time and focus on IS testing can affect quality and generate errors during early system operation [8]. Poor system quality has been reported in terms of inflexibility [15], low functionality [16], irrelevance with requirements [4], and low usefulness in supporting user-centered design [17].

ISD failure in project management is associated with slow progress, limited funds to hire experts [15], scope creep and poor risk control [10], poor communication, knowledge management, and implementation methodology [4], and poor talent management [18,19]. Scope creep can lead to rework, subsequent delays, and additional cost [4]. An IS that is halted upon completion may result in exorbitant waste [16]. Many human factors, including low productivity [3], limited technical expertise [15], emotional problems [20], low teamwork spirit [13], and low user involvement [21], lead to ISD failure. IS developers are frequently blamed even if the problems are

caused by the user’s unclear requirement specifications and business processes [22]. Human limitations, such as inability to work concurrently and overburdened mental and psychological ability, also hinder ISD success [4].

An evaluation framework known as human, organization, and technology fit (HOT-fit) can be used to specify IS success/efficiency based on the abovementioned three factors and their dimensions, namely, system quality, information quality, service quality, system development, system use, user satisfaction, organizational structure and environment, and net benefits [6,23]. A stable organizational structure ensures that IS project development can be completed on time without drastic changes. Organizational settings and culture, such as good political situation and the absence of bureaucracy, also contribute to IS success.

B. Lean IT Approach

Lean evolution in IT, known as lean IT, encompasses the following five principles similar to those in lean manufacturing: value identification, map value flow, construct flow, supply based on demand, and pursuit for perfection [24]. Lean IT is defined as a holistic management approach in IT organizations, in which continuous improvement is attained by eliminating waste and unevenness by using the lean philosophy, principle, and tools [25,26]. The literature on lean IT is scarce despite its similar principles with manufacturing [25-28]. Moreover, the implementation of lean IT remains to be at its infancy compared with those in other industries [28,29], particularly in IT management [26]. The lean IT approach is also vague due to missing principles or suitable lean activities [30].

Waste in Lean IT. Waste can be defined as any activity that does not add value to products or services or does not fulfill customer requirements [24]. Waste is categorized as overproduction, delay, defect, transportation, inventory, motion, and over-processing [22]. Waste in the lean IT domain has been adopted from the original waste definition in the context of IT management or IT support activities. Most literature on IT waste is related to ISD [4,5].

Lean IT Tools. Various lean tools, including just-in-time, automation (Jidoka), leveling (Heijunka), continuous improvement (Kaizen) and standardized work, 5S, Kanban, Takt Time, Andon, and 5-whys, can be used at different lean transformation stages [19]. Common lean IT tools usually entail virtual Kanban, automation, Kaizen events, and scrum meetings in the ISD context.

Lean IT in ISD. Waste in ISD is inevitable, and it affects cost, schedule, scope, and project survival. The literature reports various waste types, including those in the ISD context [4,5,31]. Waste in ISD includes excessive IS features, relearning due to lost knowledge, partially done work, handoffs, task switching, delays, and defects [32]. Table I illustrates the relationship between the ISD failure issue and potential waste. The seven principles of lean software development (LSD), namely, waste elimination, build-in quality, knowledge creation, commitment deferment, fast delivery, respect for people, and optimize-the-whole approach [33], can be extended to organizational ISD.

TABLE I. RELATIONSHIP BETWEEN ISD FAILURE AND WASTE IN SDLC

SDLC phases	ISD failure issue	Waste types
Requirement elicitation	Lack of involvement among IS developer, user and stakeholders	Extra features, delay
	Poor scope control and constantly changed requirements	Extra features, partially done work, delay
	Difficulty in understanding tacit requirement meaning	
	Poor documentation management	
Feasibility study	Poor evaluation of project resources availability such as financial and expertise	Partially done work, relearning, delay, defect
	Risk control and change management are not implemented during the early phase of development	Extra features, partially done work, delay, defect
Development	Slow project progress due to lack of technology expertise	Delay, relearning
	Loss of expertise with no replacement	Relearning, task switching, handoff
	Recurring rework in the debugging process	Defect, delay
	Unnecessary feature development	Extra features, partially done work, delay, defect
	Complex and irrelevant solution	Extra features, delay
	Pressure and burden on physical and mental ability	Partially done work, task switching, handoff
	Excessive emotional and psychological stress	
	Inability to implement concurrent work	
	Poor morale and teamwork	
	Poor knowledge and communication sharing	Relearning, handoff, task switching
Implementation	Incompatibility between IS and IT infrastructure	Partially done work, defect
	Difficulty in integrating new and current IS (or legacy system)	
	Lack of involvement during training session (transfer of technology)	Defect
Maintenance	Poor customer service management	Partially done work, defect
	No maintenance manual	

III. METHOD

We conducted a qualitative exploratory case study among IS developers from different organizations to analyze the root cause of waste in ISD. We employed interview, observation, and document analysis techniques to collect data. The informants were selected via purposeful sampling. Most informants are full-stack developers who are highly involved in system development activities and skillful in database, network, and system management. The informants were also selected on the basis of their ability to answer the research questions. One-on-one and group semi-structured interviews were conducted with four- and three-pair informants among the ten IS developers with different backgrounds in organizational,

project, designation, role, and work experience (Table II). The document analysis included the use of physical documents, such as meeting minutes, reports, software requirement specifications, user acceptance tests (UATs), change request forms, and user manuals. Digital documents included e-mails, system architectures and designs, storyboards and source codes.

TABLE II. INFORMANT LIST

Code	Designation	Experience (Year)	Organization	Status Project
R1	System analyst	2	Research	Successful
R2	System analyst	2	Research	Successful
R3	IS developer	3	System development	Fail
R4	IS developer	3	System development	Fail
R5	System analyst	5	Research	Successful
R6	IS developer	4	Research	Successful
R7	IS developer	5	Research	Successful
R8	IS developer	3	Research	Successful
R9	IS developer	2	Research	Successful
R10	IS developer	2	Research	Successful

We employed thematic and inductive analyses [34]. The data were analyzed to discover and highlight common phrases or keywords and subsequently identify patterns or themes. Upon identifying certain themes, the coded data were compared with the non-coded ones to verify their accurate meanings and their correlation with the research question. The final themes were refined based on the discovered discrepancies or new themes. First, we analyzed the ISD success criteria from the IS developers' perspective based on the HOT-fit framework [6,22] to understand their holistic views on successful IS. Second, we analyzed the ISD leanness level to evaluate the extent in which IS developers apply basic lean principles in an organization. Third, we analyzed waste in ISD based on the waste category [32]. The analyses were then combined, and a unique transformation plan for ISD was recommended for the case.

IV. RESULTS

The case study was conducted at an ICT Research Unit (IRU) of a Malaysian electricity company. The IRU works dynamically in various IT projects, including system development, management of IT infrastructure and networks, multimedia assignments, digital arts, IT asset acquisition, and policy monitoring. Several large-scale projects used to require IRU to hire third-party experts among vendors, interns, or freelance workers. The ISD managed by the IRU can be categorized as R&D project support, support service, and laboratory services. The IRU, as a non-profit organization, develops IT infrastructures and ISs by using open-source platform resources, in which internal expertise is utilized, to save operational cost. IS developer group used two main ISD methods, namely Agile and SDLC traditional phases.

A. ISD Success Criteria

The ISD success factors in this study were identified from the IS developer perspective (Table III) based on the nine main components of the HOT-fit framework. Fig. 1 illustrates the ISD success criteria factors and their relationships at IRU.

System quality. IS success is influenced by the fulfillment of the user requirement, user interface design, and third-party feedback. The fulfillment of user requirements refers to the extent in which the system can successfully meet the user's finalized business process that is free of bugs upon testing. A system is considered successful if it can be completed according to plan and subsequently used by the user (R1). Moreover, the design should entail an interface that can be easily adopted by users of all age ranges and learned independently given its minimalist and attractive features (R10). System quality can also be measured on the basis of the feedback from a third party independent from the IS team. "Positive feedback and recommendations from an external party who test the system are considered for system improvement" (R3). In general, most informants attest to the importance of system quality; they acknowledge it as a critical success factor, and errors and usability issues should be continuously addressed.

Information quality is associated with secure data storage, information processing accuracy and efficiency, and data security. Data security requires data aggregation and analysis to protect the system from any threat (R3). Service quality is focused on change request and errors. IS success is defined by the level of service support (1) for the user upon IS completion and 2) for fulfilling the requirements of system and business changes. Support provision to address errors is related to finding solutions to semantic or syntax errors in all aspects of system use. IS developers must be prepared to provide support to user request (R6).

TABLE III. ISD SUCCESS CRITERIA AT IRU

Success factor	Success criteria
Technology	
System quality	fulfillment of user requirement, user interface design, and third-party feedback
Information quality	data storage; information processing accuracy and efficiency; data security
Service quality	support on change request and error
human	
System development	requirement clarity; accuracy of hardware specification; requirement consistency; realistic requirement; clarity of business process
System use	use level; intention and purpose of use
User satisfaction	pleasant use; ease of learning
organization	
Organizational structure	management support; change management; teamwork information availability; cooperation and involvement idea sharing; financial stability
Net benefits	support vision and mission alignment support operational planning

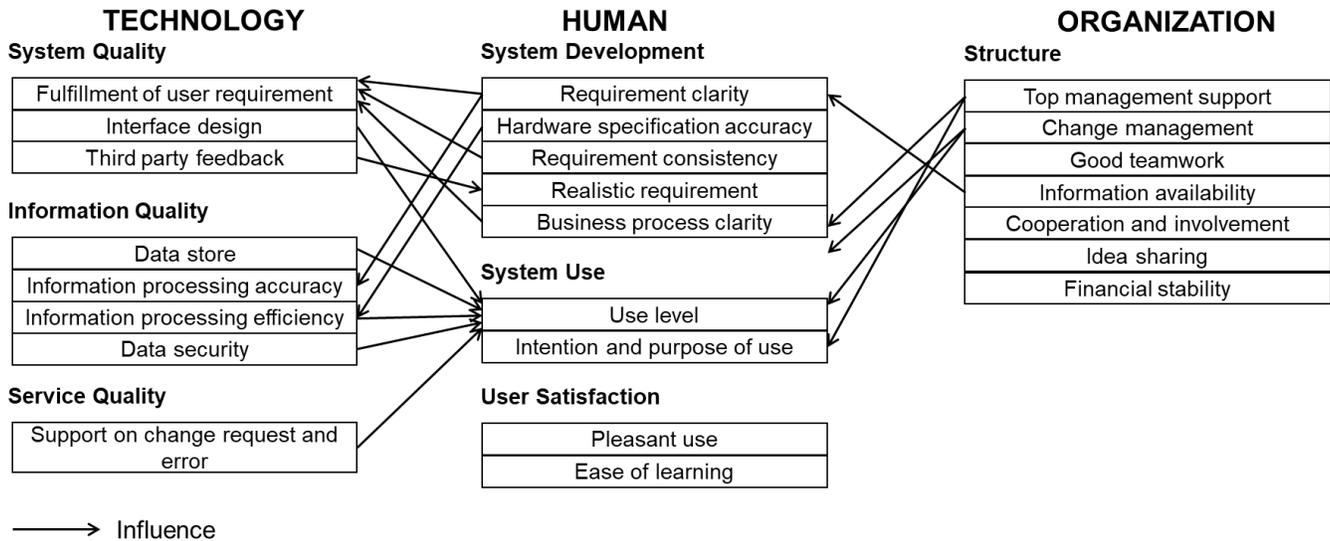


Fig 1. ISD Success Criteria Factors and their Relationships at IRU.

System development factors that influence ISD include requirement clarity and accuracy, hardware specification requirements, realistic requirements, and business process clarity. Clarity is important in ensuring that the IS will be developed according to user requirements in terms of the functional, nonfunctional, and user experience aspects. “Basic function, actual system purpose, accurate definition of main functions, and additional functions to support system efficiency must also be defined” (R2). Consistency refers to maintained, instead of constantly changed requirements that are aligned with the original ISD project scope (R8).

The accuracy of hardware specification is also important for system development in terms of cost-saving targets, compatibility with the developed IS platform, and avoidance of waste, such as the development of a sophisticated but unnecessary hardware for a simple system (R9). The realistic requirement refers to feasible requirements that can be implemented using available technologies, cost, time, and IS developers’ skills. This realistic requirement depends on finalizing a business process instead of simply relying on the expectation and imagination of certain stakeholders (R4). In addition, a clear and finalized business process that is commonly understood by all involved units is critical in avoiding confusion during system development (R3).

System use measures the level, intention, and purpose of use. A high usage level refers to the optimized use of system modules and features (R2). Intention and purpose are also important; without them, ISD becomes a waste (R5). Meanwhile, user satisfaction can be measured on the basis of user experience, including system pleasantness and ease of learning. A successful system is also associated with its attractive, intuitive, and interactive features.

The factors related to organizational structure were highly cited among informants. Management support plays an important role in ensuring the enterprise-wide use of IS and its fit with the organization’s mission and vision, thus facilitating stakeholder involvement, planning, and implementation as a

means of managing changes (R4) and expediting processes that require approval, such as process redesign (R5). A conducive organizational setting and the freedom to share ideas can yield the much needed harmonious ambience among IS developers. Moreover, the ISD team members need to ensure cooperation and support one another to successfully attain their objectives. “Each team member needs to play their respective roles and facilitate one another’s task to achieve the [system] objective” (R10). Financial stability is also critical in ensuring ISD survival; that is, without being affected by adjustment risks or budget limitations, as these concerns affect overhead expenses or hardware acquisition (R2). The organization must constantly be apprised of the information required by IS developers throughout the project. Net benefits refer to the overall benefits obtained by organization. The unit strategic plan should be aligned with that of organization, and the ISD should be able to facilitate the management task of organization during operational planning (R6).

B. Leanness Level of ISD

The analysis of the leanness level or lean compliance in the ISD context is imperative in examining the current business process. The current practices, which are contrasted with the LSD principles for benchmarking purposes [1], are summarized in Table IV and discussed further as follows.

Waste elimination. The root cause of waste is identified and removed using lean tools. IS developers remove waste based on the understanding of the process value and user problems. Focus is given on system functionality to optimize resources. “Problems and requirements are specified before system development; problems that genuinely require a system [as a solution] indicate a valuable task” (R1). The current practice strives to constantly comply and agree with the original project scope and planned schedule (R4) to avoid scope creep and rework due to uncontrollable user requirements. Organizational priorities pull instead of push production, in which ISD is only initiated on the basis of demand (R1). The IRU unit organizes case or scenario presentations related to system use to increase

the team members' understanding and facilitate the acquisition requirement process. Waste in task changes can be minimized by postponing the ad hoc task until all current tasks have been completed. The *build-in quality* principle refers to the IS developers' assurance regarding the absence of system flaws that can affect the system's functionality. Some of the organizational practice for improving system quality is to always understand and fulfill user expectations by obtaining their feedback during system operation, such as "asking users to describe their experience or expectation from the system" (R2). In a workgroup environment, the use of work collaboration software, such as *Sourcetree*, may enhance ISD quality (R3).

Knowledge creation involves the gathering of knowledge in the course of project development for reference purposes. Some common practices include studying problems and mastering problem-solving skills (R1), generating and disseminating knowledge in meetings, mentoring sessions, and weekly or biweekly discussions (R2), and writing code snippet notes that can be reused and shared (R7). When IS developers are stuck with work and cannot generate new ideas, knowledge can be generated by searching the Internet or consulting other experts (R5). Knowledge is also generated from online learning portals, such as Udemy and Coursera, subscribed by the IT manager for subordinates (R6).

Deferred commitment means postponing decision making to the last minute; in this manner, certain ISD modules can be initiated without having to finalize the system requirements. IS developers can apply this principle by synthesizing information and drafting solutions until the actual solution is discovered (R2).

TABLE IV. LEANNESS LEVEL OF ISD AT IRU

LSD principles	Organizational practices
Eliminate waste	Understand value and problem statement Ensure adherence to the original scope and agreement with it Develop system based on requirement Present use-case during requirement elicitation Postpone ad-hoc task until current tasks are completed
Build quality in	Understand and fulfill end user expectation Use work collaboration software
Create knowledge	Revise issue and be skillful in problem solving Share knowledge during meeting, mentoring, and discussion sessions Create and share code snippet Refer to experts Subscribe online training
Defer commitment	Postpone decision making until solution is discovered
Immediate result	Prioritize and quickly completed functional requirement
Respect for people	Avoid arguments and confrontation Establish effective communication Support one another Understand team members' limitations
Optimize the whole	Understand overall problem

Fast delivery refers to the development of the IS as soon as possible without compromising its quality and cost based on the ability of the IS developers. This principle adheres to the prioritization and completion of the functional requirement in a quick manner. "My to-do list is based on priority and completion targets" (R8).

Respect for others is emphasized in communication and activities among IS developers. Arguments and confrontation, despite the different specializations and backgrounds, should be avoided. For example, developers can describe their task complexity to other members who are unfamiliar with specific ISD processes (R2). Respect is also cultivated through effective communication, such as providing opportunities for IS developers to provide suggestions and ideas independently to users during discussions (R4). Helping one another based on abilities, regardless of role and position, in each ISD phase indicate respect. Developers can divide tasks based on the requirement analysis to ensure that they work in the same direction. "When we obtain our module, we divide the task, aid in unfinished job, and ensure that other programmers' codes are in line with ours" (R6). Moreover, the recognition of strengths and weaknesses of each team member can lead to common understanding and a harmonious working ambience (R2).

The *optimize-the-whole* principle views the process holistically. On this basis, IS developers can understand problem as a whole. Moreover, the developers can propose the best solution based on team capabilities and enhance the algorithm and system design system. "We seek to view from wider and different angles in consideration of the solution alternatives" (R2).

C. Waste in ISD

Waste in ISD, which has been analyzed prior the identification of the root cause based on the waste categories [1], is listed Table V. *Extra features* include those items beyond the project scope and are not specified by the user, "but they could have overlooked them" (R1), which do not add value to the IS developer team. These wastes are also caused by inappropriate technology selection, hardware and software acquisition strategies (R6), and unclear business processes, such as the absence of a valid purpose for process improvement or automation. Unclear requirement specification is a common scenario attributed to user difficulty in the finalization of the system requirements (R4).

Relearning refers to the acquisition of the same or previously acquired knowledge. Knowledge loss are costly in terms of skill, expense, and time. This type of waste is also due to the lack of experts or error reference sources online for guidance purposes (R1) and prolonged abandonment or non-use of knowledge. The knowledge obtained from previous training and projects were only stored but not referred as they are not needed. However, stored knowledge may later become irrelevant to the current requirements due to technology advancements. New bug and error discoveries can be categorized as new knowledge, but no specific platform can be used to store and disseminate them, resulting in limited reference for encountered errors. IS developers who focus on specific ISD tasks over long durations tend to forget about their

other existing knowledge. “I have been focusing on development activities for so long and have forgotten how to solve the same encountered issue [in other aspects]” (R4).

Partially done work refers to work that is postponed and “in progress” for too long due to certain reasons. This waste is attributed to external disruptions affecting the IS developers who focus on the ISD (R7) and deadlocks or lack of ideas in solving problems, such as an encountered new bug and the low commitment to execute concurrent tasks. “[Solving] a new bug or problem consumes time and delays my work (R1). As a result, IS developers usually focus on easier tasks, leaving the unsolved task, a partially work done.

TABLE V. WASTE ROOT CAUSE IN ISD PROJECT AT IRU

Waste type	Waste root cause
Extra features	Features beyond scope and not required Inappropriate selection of technology, hardware and software acquisition Unclear requirement specification Unclear business process – no actual need for process improvement
Relearning	Unavailable error reference Prolonged unused knowledge
Partially done work	External disruption Stuck or out of idea
Handoff	Transfer of IT infrastructure to other party Sudden resignation Module based outsourcing Sudden termination of IS developer
Task switching	Significant focus disruption
Delay	Time needed to regain focus Waiting for endorsement or approval Waiting for commitment or user involvement Lengthy meeting Fixing newly discovered error Time consumed to understand source code written by other developer Emotional disruption
Defect	Rework due to uncontrolled change request

Handoff refers to a handover work. In IRU, this waste stems from handing over IT infrastructure to external parties. Handoff occurs immediately upon module completion. An appointed vendor is expected to hand over the task during a technology transfer session with a IS developer. However, the developer is likely unable to completely understand the session, requiring him or her to reexamine the vendor’s work. The process requires a compliance to and an understanding of the company’s standard operational procedure, terms, and conditions. This approach is time consuming compared with the direct handing over of infrastructure to the internal IT management (R9). Waste can also be caused by the sudden resignation or termination of members of the IS developer team

but without leaving adequate handoff documentation. “When a member suddenly resigns, I assume that at least half of the module had been prepared, but it was [actually] less than that, so I had to [re]do it from scratch” (R3). Handoff waste can thus be caused by incomplete outsourced modules that had not been developed on the basis of the specified requirements (R3). The immediate resignation of team members results in stress and additional workload to other members due to unavailable handoff documentation (R10). Transferring IT infrastructure to another party also results in a handoff. IS developers should provide comprehensive documentation based on specific standards to ensure that the system can function well in the new IT infrastructure.

Task switching refers to changing one task to another, causing an IS developer to lose the critical focus required in the ISD. Disrupted focus affects the quality of a module-in-progress due to the time needed to regain focus and emotional stability and address possible confusion and lapses. “System development requires a hundred percent focus. A high-quality system needs to be developed [only] one at a time” (R1).

Delay, the most commonly cited type of waste, requires waiting for a process to be completed before proceeding to the next one. Delay is caused by the time needed to regain the disrupted focus of the IS developer due to his or her disorganized thinking and task switching (R2). Delay can also be attributed to the time wasted resulting from authorization or approval from another party, such as those related to infrastructure or work orders (R7). Unavailable instruction from the top management also delays commitment, cooperation, and user involvement in the requirement elicitation, testing, and system use phases. “My biggest problem is to obtain user commitment to use the system” (R2). In addition, some automation protocols in the business process requires changes in standard documentation, such as ISO, which are controlled by the top management. Lengthy meetings also waste time and cause delay (R1). Encountering a new error results in delays, as the IS developer usually takes time to understand the cause of error (R9). However, commonly encountered errors can be fixed faster than other types of errors through IS developer experience. In the handoff scenario, delays occur when the IS developer takes too much time to understand the source codes developed by another individual (R10). This waste may also stem from emotional disruption, which is closely related to task switching (R1).

Defect refers to resources that are repeatedly used to improve a developed IS. In this case, defect is caused by additional requirements that are significantly diverted from the original scope (R2). Therefore, the developed features need to be removed in spite of the consumed cost and time. This waste may also stem from uncontrolled and improper management of change, as requested by the project manager. “A user requests for a design during the first meeting. In the second meeting, the user requests for a requirement change. It is a major requirement, so I feel that the first design is totally wasted” (R1).

D. Document Analysis

Table VI summarizes the results of our document analysis. Most projects have properly documented the agreed system

specification requirements for the sake of the IS developer and the user. The different waste types, including constantly changed business processes, ambiguous requirements, lack of details and confusing documentation, inaccurate business process modeling, inability to consider user satisfaction, absence of risk planning and change management plan, and poor management of document version, have been identified.

In particular, UAT documents can help identify weaknesses in the documentation of user feedback obtained during UAT sessions. Moreover, IS developers may be affected by the limited time to perform unit testing. The analysis of change request forms can present weaknesses in terms of managing the change request, sophisticated requirement, and unrealistic time to address changes. Poorly written user manuals may manifest other weaknesses that are unable to guide in solving issues.

TABLE VI. DOCUMENT ANALYSIS IN ISD AT IRU INFORMANT LIST

Document type	Findings
System requirement specification	Constantly changed business process Overly vague requirement specification Lack of details in producing document requirements Inaccurate business process modelling Not considering user satisfaction Unavailable risk planning Unavailable change management Weak management of document version
User acceptance test	Poor user feedback documentation Lack of time to perform unit testing
Change request form	Poor management of change request Sophisticated change requirement Unrealistic time duration to address request
Manual User	Unhelpful in solving issue Poorly written

V. DISCUSSION

A. ISD Success Criteria

The IS developer team is also committed in delivering the best results in each ISD phase, proving that they are enthusiastic in producing quality IS that can add value to the organization. Meanwhile, the accurate processing of information is highly emphasized, as most in-house developed ISs require their own logic and formula. Inaccurate information can jeopardize organizational credibility in generating reports, which are eventually shared to the third party. As for service quality, support service, which includes warranty, service level agreement, or change request, may be changed by formally submitting a request form via manual, e-mail, phone, and during meetings to ensure that the IS can be continuously developed according to the organization's requirement. In system development, the clarity requirement in the IRU context refers to the current business process that needs to be automated through IS. A clearly defined business process can minimize waste risk in terms of rework or development of

unnecessary extra features. This requirement should also be maintained by the developers while finalizing the work scope with the users even before development phase is implemented. Inconsistent requirements are expected to challenge the original project plan, particularly the management of the work schedule.

The system's usage level refers to the extent in which the IS can be used to support the business process. In this study, the factors that discourage system use include system sophistication and poor user interface design. We learned that some ISs were only developed to fulfill departmental aspirations in the organizational digital transformation plan but with no real intention to use the system. A user can be regarded highly satisfied if he or she can use the system pleasantly, proving that ISs should possess high usefulness and able to attract user interest to learn and understand the system. Many interface frameworks, such as bootstraps or material design, can be used to design ISs according to industrial best practices.

Top management support plays an important role in determining IS success, given the authority in decision making that they provide for resource planning and influencing the sponsorship of the ISD. Improper management decisions developers. The management should also ensure that the ISD team remains unchanged during system development to facilitate smooth communication, cooperation, and information acquisition and ensure that work synchronization can be managed optimally. The collaborative nature of ISD projects requires a good team, high commitment among team members, and alignment between the IS developer and the user. IS developers are responsible in ensuring the fulfillment of user requirements; by contrast, users are accountable in finalizing the information, business process, and requirements for the IS developers. Most informants have encountered problems in adhering to project schedule due to the constantly changing user requirements, which result in waste, such as delays and rework. In addition, a user may encounter problems in providing accurate information for the system requirement due to poor information availability and inefficient idea sharing. Nonetheless, many informants argue that a well-controlled and managed change management can lead to IS success. A conducive work environment can be established by encouraging good rapport among team members.

Organizations can move towards the fourth industrial revolution if they focus on adapting IT in their business models and daily operations. Thus, developing an IS in line with the organizational vision and mission can add value and benefits, and the organization can manage its resources effectively and efficiently. A successful IS also supports operational planning. Technologies, such as big data analytics, can assist the organization predict organizational resources accurately and support the management in making decisions effectively.

B. Leanness Level of ISD

Leanness level or compliance to lean IT principles is crucial in understanding the current ISD situation. Although most informants have no specific knowledge on LSD principles, their work practices are aligned with most of the abovementioned principles. In waste elimination, although the developers have attempted to maintain the original project

scope, non-finalized and constantly changing business process can result in scope creep. However, the IS developers become highly motivated with their work upon understanding the values that can be achieved through their respective ISD specializations, thus indirectly eliminating waste in terms of partially done work and delay. From the build-in quality aspect, feedback on IS improvement are gathered not only through users but also from all parties who use the system. This effort is encouraged by offering system updates or improvements to users. Work collaboration software, such as Sourcetree and Jira, are also used to expedite source code generation, ease the work planning, and ensure the creation of non-conflicting codes by multiple IS developers. In terms of knowledge creation, despite the good sharing practices, some of the knowledge are lost upon acquisition due to the lack of proper documentation or the absence of a knowledge management system.

Postponing decisions enable users and developers to gather more information, including risks, to avoid rework resulting from requirement changes. In terms of fast delivery, prioritization can be achieved by considering the system module, functional (instead of non-functional) requirement, and user experience. Some functional requirements consume time due to their logic or calculation complexity. Respect for others can cultivate a harmonious group and a conducive work environment that encourage freedom of speech in sharing ideas and constructive comments, motivate the IS developer team, and avoid stress that affect work productivity. Finally, the optimize-the-whole practice enables an enterprise to seek wide buy-in solutions.

C. Waste in ISD

Waste in ISD must be identified prior the finalization of the transformation plan. Fig. 2 summarizes the root cause of ISD IS developers can apply the *Plan, Do, Check, and Act* (PDCA) cycle throughout the ISD to enhance its quality. In the *Plan* phase, the problem is examined to identify the best method to fulfill the user requirement. Shortlisted methods are tested in small scale to measure their effectiveness during the *Do* phase. In the *Check* phase, the obtained information is analyzed to determine the feasibility of the selected method in solving the problem. If successful, the tasks will be implemented in the *Act* phase. Moreover, the quality of the whole IS function can be enhanced using *Poka-Yoke*, which means to prevent mistakes and identify and avoid errors. Applying these elements during the ISD can ensure IS quality, subsequently avoiding various inefficiencies, such as errors and rework at IRU. Delay and defect are the most significant waste categories at IRU. Table VII summarizes the relationship between ISD failure at IRU and waste type [2].

IS developers can choose different methods to obtain knowledge, but this knowledge may not be managed, stored, and disseminated properly. An online- and cross-boundary knowledge sharing [2] portal enables knowledge to be stored in various forms and shared among IS developers intra and inter organization. In disseminating knowledge, the experienced and trained IS developers can provide training or tutorial sessions based on their expertise. A project manager may seek to postpone his or her decision making until a solution is discovered, while an IS developer should not make a decision hastily without scrutinizing the existing resources.

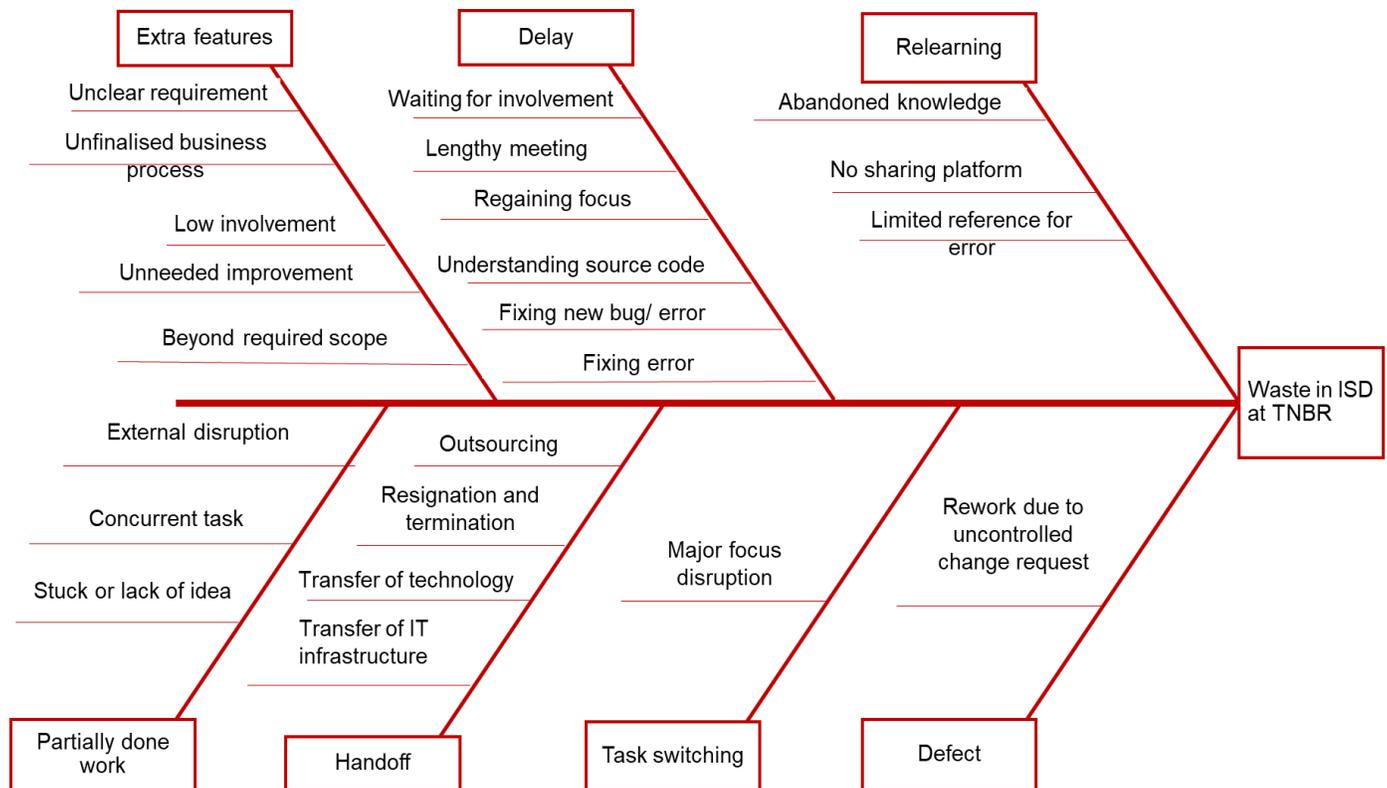


Fig 2. Fishbone Diagram for Waste Root Cause in ISD at IRU.

TABLE VII. RELATIONSHIP BETWEEN ISD FAILURE AND WASTE TYPE IN SDLC PHASES AT IRU

SDLC phases	Failure issues of ISD at IRU	Waste types
Requirement elicitation	Unclear requirement specification	Extra features
	Unable to finalize business process	
Analysis	Vague business process and no valid reason for improvement or automation	Extra features
Development	Develop features that are out of scope, extra, and excessive	
	Knowledge that is abandoned and stored but not referred to	Relearning
	Unavailable specific platform for storing and disseminating knowledge	
	Limited reference to encountered error	
	External factor disruption	Partially done work
	Concurrent work	
	Out or lack of idea	
	Module based outsourcing	Handoff
	Sudden and unplanned resignation and termination of the IS developer team	
	Time needed to regain lost focus	Delay
	Rework due to uncontrolled change request	Defect
	Significant focus disruption	Task switching
Testing	Time to understand third-party source code	Delay
Implementation	Transfer of technology upon a module completion	Handoff
	Transfer of IT infrastructure to another party	
Maintenance	Fixing new encountered error	Delay
	Recurring defect	

D. Transformation Plan for Lean IT in ISD Organization

The lean IT approach is correlated with the organizational IS success factors. The IS developer team must understand what comprises a successful IS criteria set, besides practicing lean IT principles based on the HOT-fit framework and applying lean tools to manage ISD optimally. Fig. 3 illustrates the transformation plan based on the findings derived from this study. An ISD can be improved using various lean tools, such as the fishbone diagram, to understand a problem and its root cause and subsequently identify alternatives to the best

solution. Scrum meeting sessions can be held at the early or late part of the week among IS developers to expedite tasks. In the elicitation phase, the identified business process can be optimized through automation to significantly eliminate waste and increase operational efficiency. Smooth value flow can be ensured by properly planning the task assignments, thus avoiding unevenness and overburden.

In expediting deliverables, IS developers can easily prioritize certain tasks by using a *Kanban* board. This lean tool is extremely useful in visualizing project progress. Tasks are usually drawn on a white board or depicted using online application tools, such as *Asana* or *Jira*, in which case the *To-Do*, *In Progress*, and *Done* tasks can be easily viewed. In realizing the respect-for-others principle, teamwork spirit, instead of individualistic attitude, must be cultivated because IS deliverables require different types of expertise from the team. As for overall optimization, the IS developers can apply value stream mapping (VSM) to understand the value flow, allowing them to easily identify and eliminate waste. VSM can also improve ISD management and reduce unevenness and over-burden of IS developers. Table VIII summarizes our recommendations for ISD improvement at IRU.

The framework discussed above ensures system quality with high user fulfillment, good interface design, and positive feedback from the third party. Information quality depends on the data storage method, the accuracy and efficiency of information processing, and the data security level. Service quality needs to be offered to the users in the form of change request or error mitigation. IS developers must ensure that the requirements are clear, consistent, and realistic to subsequently warrant smooth system development. Business processes must also be finalized to avoid uncontrollable changes. The accurate selection of hardware specification is also crucial in optimizing the IS developer specialization.

Moreover, IS developers need to learn the level of IS use to identify its value and contribution towards the ISD. User intention and IS purpose should be identified to ensure that they will be revisited upon project completion. IS developers need to focus on pleasant system use and ease of learning to achieve user satisfaction. The IS will likely be successful if it can gain the support of the top management. Good rapport among team members is important in cultivating the respect-for-others attitude. IS developers should also be given access to sufficient information to ensure the fulfillment of user requirements. Cooperation and involvement among IS developers and stakeholders are both critical in obtaining feedback and ideas. The project manager must play his or her role in ensuring the best project financial state. IS developers must contribute towards supporting both the organizational vision and mission and the operational planning.

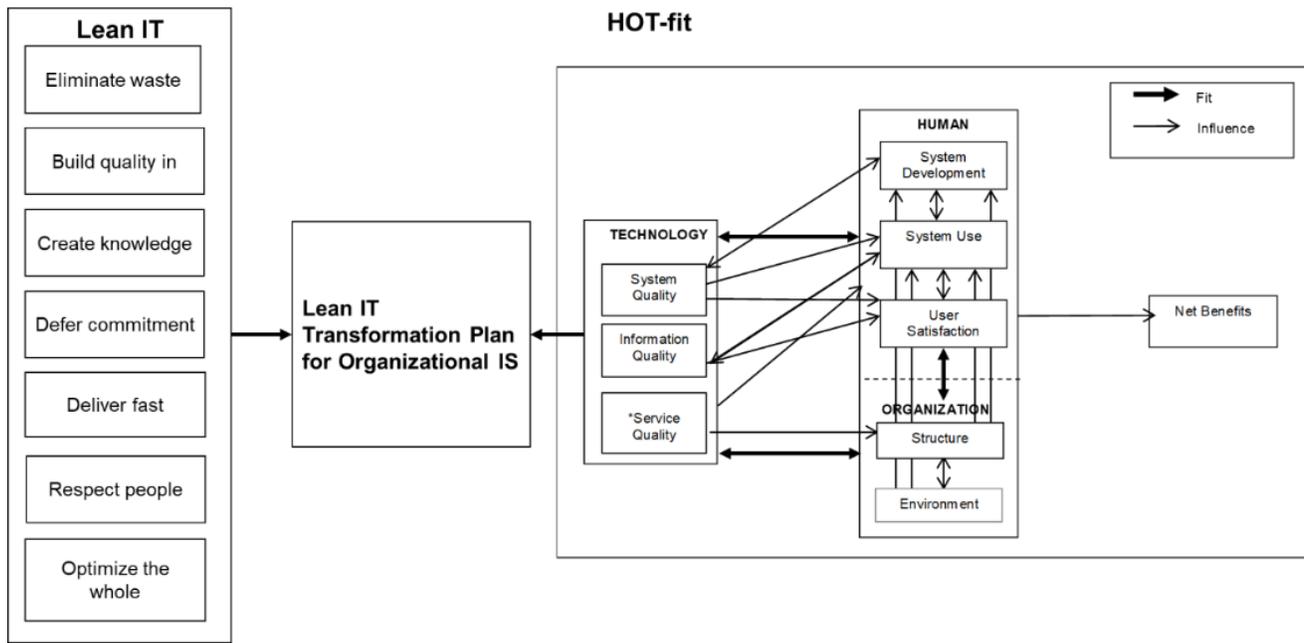


Fig 3. Lean IT Transformation Plan in IS Organization.

TABLE VIII. RECOMMENDATIONS FOR ISD IMPROVEMENT AT IRU

Principle	Recommendations for ISD improvement
Waste elimination	Use root cause analysis to understand problems Organize weekly scrum meeting Optimize business process through automation (<i>Jidoka</i>) Avoid uneven workload assignment
Build quality in	Use the PDCA methodology to improve IS Use <i>Poka-Yoke</i> in all IS functions
Create knowledge	Establish portal for online knowledge Held specific training and tutorial sessions
Defer commitment	Postpone decision until solution is discovered
Deliver fast	Sort and determine priority using Kanban board
Respect for others	Enhanced teamwork spirit among IS developers
Optimize the whole	Understand the value flow by using VSM

VI. CONCLUSION

The lack of understanding in developing a successful IS and avoiding waste in the ISD context triggers a chain reaction that hinders the achievement of a lean and efficient ISD. Therefore, IS developers need to understand the IS success factors and practice lean IT principles. This study contributes in the identification of the ISD success criteria, waste categories in ISD, and levels of organizational compliance to lean IT principles. More importantly, this study proposes a lean IT transformation plan to evaluate ISD success and waste in the ISD context based on the HOT-fit framework and a lean IT approach to guide IS developers in producing successful

systems. The plan features a holistic, systematic, and structured approach in ISD and supports IT and business alignment in the organization.

The study scope is limited to the IS developer perspective only, and access to document analysis has been constrained by data confidentiality. However, the study is applicable to other ISD endeavors with the same development environment and project scale. Future work can be carried out to refine and test the proposed ISD transformation plan in terms of measures, implementation method, breadth of scope, and inclusion of other stakeholders' perspective, including end-users or the management. The transformation plan may guide IS developers in delivering high-quality IS to support the organizational vision and mission and in progressing towards the fourth industrial revolution.

ACKNOWLEDGMENT

We thank all study participants from IRU for their collaboration.

REFERENCES

- [1] R. Marques, G. Costa, M. Mira da Silva, and P. Gonçalves, "A survey of failures in the software development process. 25th Eur Conf Inform Syst", pp. 2445-2459, 2017.
- [2] V. Rastogi, "Software development life cycle models- comparison, consequences" Int J Comp Sc IT vol. 6 (1), pp.168-172, 2015.
- [3] Y. Lindsjörn, D.I.K. Sjøberg, T. Dingsøy, G.R. Bergersen, and T. Dybå, "Teamwork quality and project success in software development: A survey of agile development teams", J Syst Softw vol. 122, pp. 274-286, 2016.
- [4] T. Sedano, P. Ralph, and C. Peraire, "Software development waste". IEEE/ACM 39th Int Conf Softw Eng (ICSE), pp.130-140, 2017.
- [5] H. Alahyari, T. Gorschek, and S.R. Berntsson, "An exploratory study of waste in software development organizations using agile or lean approaches: A multiple case study at 14 organizations", Inform Softw Tech vol. 105, pp. 78-94, August 2018.
- [6] M.M. Yusof, J. Kuljis, A. Papazafeiropoulou, and L.K. Stergioulas, "An evaluation framework for Health Information Systems: human,

- organization and technology-fit factors (HOT-fit)", *Int J Med Inf* vol. 77 (6), pp. 386-398, 2008.
- [7] S.A. Ebad, "Influencing Factors for IT software project failures in developing countries — a critical literature survey", *J Soft* vol. 11 (11), pp. 1145-1153, 2016.
- [8] M.S. Hossain, "Challenges of software quality assurance and testing", *Int J Softw Eng Com Syst* vol. 4 (1), 133-144, 2018.
- [9] K.C. Laudon and J.P. Laudon, *Management Information Systems: Managing the Digital Firm*. New York:Pearson, 2019.
- [10] A. Alami, "Why do information technology projects fail?", *Procedia Com Sc* vol. 100, pp. 62-71, 2016.
- [11] M.Z. Yusoff, M. Mahmuddin, and M. Ahmad, "A knowledge work productivity conceptual model for software development process in SME", *ARNP J Eng App Sc* vol. 10 (3), pp. 1123-1130, 2015.
- [12] M.K. Sharma, "A study of SDLC to develop well engineered software", *Int J Adv Res Comp Sc* vol. 8 (3), pp. 520-523, 2017.
- [13] M. Shameem, B. Chandra, C. Kumar, and A.A. Khan, "Understanding the relationships between requirements uncertainty and nature of conflicts: a study of software development team effectiveness", *Arab J Sc Eng* vol. 43 (12), pp. 8223-8238, 2018.
- [14] S.Z. Salam, and M.M. Yusof, "Knowledge integration in determining user requirements" *The 9th Malaysian Softw Eng Conf, Seri Kembangan, Malaysia*, pp. 112-116, 2015.
- [15] S. Aitzaz, G. Samdani, M. Ali, and M. Kamran, "A Comparative Analysis of In-house and Outsourced Development in Software Industry", *Int J Com App* vol. 141 (3), pp.18-22, 2016.
- [16] L. Mieritz, *Gartner survey shows why projects fail. This is what good look like*. 2013.
- [17] B.S. Masood, "Usability evaluation method for agile software development", *Int J Com Sys Soft Eng* vol. 1 (1), pp. 29-40, 2015.
- [18] V. Blijleven, K. Koelemeijer, and M. Jaspers, "Identifying and eliminating inefficiencies in information system usage: A lean perspective", *Int J Med Inform* 107:40-47, 2017.
- [19] J.K. Liker, *The Toyota way: 14 management principles from the world's greatest manufacturer*, New York: Mc Graw Hill. 2004.
- [20] T. Taipalus, V. Seppänen, and M. Pirhonen, "Coping with uncertainty in an agile system development course", *J Inform Syst Edu* vol. 29 (2), pp.117-126, 2018.
- [21] A. Janes, "A guide to lean software development in action", *IEEE 8th Int Conf Softw Testing, Verification and Validation Workshops, ICSTW* pp.1-2, 2015.
- [22] S. Bell and M. Orzen. *Lean IT: enabling and sustaining your lean transformation*, Boca Raton: CRC Press, 2011.
- [23] M.M. Yusof, "A case study evaluation of a Critical Care Information System adoption using the socio-technical and fit approach", *Int J Med Inf* vol. 77 (6), pp. 377-385, 2015.
- [24] J.P.Womack and D.T. Jones, "Lean thinking—banish waste and create wealth in your corporation", *J Oper Res Soc* vol. 48 (11), pp. 1148-1148, 1997.
- [25] J. Kobus and M. Westner, "Lean management of IT organizations: a perspective of IT slack theory", *Int Conf on Inform Syst*, pp 1-12, 2016.
- [26] J. Kobus, M. Westner, S. Strahinger, and D. Strode, "Enabling digitization by implementing Lean IT: lessons learned", *TQM J* vol. 30 (6), pp. 764-778, 2018.
- [27] A.N.A. Wahab, M. Mukhtar, and R. Sulaiman, "A conceptual model of lean manufacturing dimensions", *Procedia Technology* vol. 11, pp. 1292-1298, 2013.
- [28] N.A. Kalong and M.M. Yusof, "Waste in health information systems: a systematic review", *Int J Health Care Q Assur* vol. 30(4), pp. 341-357, 2017.
- [29] J. Wanitwattanakosol and S. Noamna S, "Action research framework in Lean Information Technology", *Int Conf Syst Sci Eng (ICSSE)*, pp. 1-3, 2018.
- [30] R.K. Yadav, M.L. Mittal, and R. Jain, R, "Adoption of lean principles in software development projects", *Int J Lean Six Sigma*, vol. 11(2), pp. 285-308, 2020.
- [31] M. Poppendieck and M.A. Cusumano, "Lean software development: A tutorial", *IEEE Softw* vol. 29 (5), pp. 26-32, 2012.
- [32] M. Poppendieck and T. Poppendieck, *Implementing Lean software development: from concept to cash*. Boston: Addison-Wesley, 2006.
- [33] M. Poppendieck and T. Poppendieck T, *Lean software development: an agile toolkit*, Upper Sadle River: Pearson, 2003.
- [34] D.R. Thomas, "A general inductive approach for analyzing qualitative evaluation data. *Am J Eval* vol. 27 (2), pp. 237-246, 2006.