Complex Systems Informatics and Modeling Quarterly (CSIMQ)
eISSN: 2255-9922
Published online by RTU Press, https://csimq-journals.rtu.lv
Article 218, Issue 40, September/October 2024, Pages 1–35
https://doi.org/10.7250/csimq.2024-40.01

Supplementing the Build Activity in Design Science Research with Soft Systems Methodology: A Technique of Creating Frameworks for Guiding Interventions Against Unstructured Problems

Agnes Nakakawa[1*], Fiona P. M. Tulinayo[1], Geoffrey O. Tabo[2], Patrick van Bommel[3], Hans B. F. Mulder[4], and Henderik A. Proper[5]

[1] School of Computing and Informatics Technology, Makerere University, P. O. Box 7062, Kampala, Uganda
[2] Faculty of Science, Gulu University, Gulu, Uganda
[3] Institute for Computing and Information Sciences, Radboud University Nijmegen P.O. BOX 9010, 6500 GL Nijmegen, The Netherlands
[4] University of Antwerp Management School, Sint-Jacobsmarkt 9-13, 2000 Antwerpen, Belgium
[5] Institute of Information Systems Engineering, TU Wien, 1040 Vienna, Austria

anakakawa@cit.ac.ug, fturinayo@cit.ac.ug, go.tabo@gu.ac.ug, P.vanBommel@cs.ru.nl, hans.mulder@viagroep.nl, e.proper@acm.org

Abstract. Several efforts have been undertaken to define generic guidelines that address specific gaps in the ‘build’ activity of Design Science Research (DSR) artifacts, i.e., constructs, models, methods and *frameworks*, and instantiations. Unfortunately, explicit guidance is still lacking on how to coherently operationalize such guidelines when building a DSR artifact, particularly *a framework*. In addition, there is no an elaborate procedure or logical thinking pattern that can be followed when building a DSR artifact, particularly *a framework for solving an unstructured problem*. Consequently, structural compositions of some artifacts insufficiently subscribe to several general design guidelines, which often hinders the artifacts from fulfilling their intended purposes. To address this gap, Soft Systems Methodology can be leveraged during the design cycle of a DSR initiative, to elaborate the ‘build’ activity and simultaneously support the coherent operationalization of existing general design guidelines. This is demonstrated herein by presenting a **T**echnique of **B**uilding **F**rameworks for guiding **I**nterventions against unstructured problems (TBUFI). From 2011 to 2023, TBUFI has undergone 11 evaluation iterations, which involved: (a) using it to support the building of frameworks for guiding digital interventions in ten research studies; and (b) engaging information systems specialists in a group walkthrough meeting to deliberate its structural composition. Evaluation iterations since 2011 (including feedback from information systems specialists) confirm TBUFI’s ability to successfully guide the design of frameworks that can direct interventions against complex and unstructured problems, by making their ‘build’ activity more elaborate, coherent, and aligned with existing general design guidelines. Thus, TBUFI can be perceived as a supplement for the ‘build’ activity in DSR.
Keywords: Design Science Research, Design Process, Soft Systems Methodology.

[*] Corresponding author

Reference: A. Nakakawa, F. P. M. Tulinayo, G. O. Tabo, P. van Bommel, H. B. F. Mulder, and H. A. Proper, “Supplementing the Build Activity in Design Science Research with Soft Systems Methodology: A Technique of Creating Frameworks for Guiding Interventions Against Unstructured Problems,” Complex Systems Informatics and Modeling Quarterly, CSIMQ, no. 40, pp. 1–35, 2024. Available: https://doi.org/10.7250/csimq.2024-40.01

Additional information. Authors ORCID iD: A. Nakakawa – https://orcid.org/0000-0003-0891-4707, F. P. M. Tulinayo – https://orcid.org/0000-0003-0922-6172, P. van Bommel – https://orcid.org/0009-0005-9384-7183, H. B. F. Mulder – https://orcid.org/0000-0002-3304-9711, and H. A. Proper – https://orcid.org/0000-0002-7318-2496. PII S225599222400218X. Article received: 5 June 2024. Revised: 21 October 2024. Accepted: 24 October 2024. Available online: 31 October 2024.

1 Introduction

March and Smith [1] and Hevner et al. [2] broadly classify and define Design Science Research (DSR) artifacts to include: *a) constructs*, which could include a vocabulary and symbols, or a set of concepts that describe occurrences in a domain or discipline; *b) models*, which are illustrations of the structural nature of things, or statements about relationships among constructs; *c) methods*, which are guidelines or protocols or algorithms or practices that specify steps involved in executing particular tasks; and *d) instantiations*, which are implementations of constructs, models, or methods to obtain prototype systems that are deployed in specific contexts. An essential reflection on artifacts under the category of 'models' reveals that the design of any DSR artifact often (implicitly) requires formulating three subgroups of models, which include: i) Conceptual models which represent concepts that describe a problem domain to enable understanding of the problem domain in which an envisioned DSR artifact is supposed to 'work'; ii) Conceptual models which represent concepts that describe a solution domain to enable understanding of the solution domain that grounds the envisioned DSR artifact; and iii) Formal models that illustrate the structural and logical nature of the actual artifact or solution.

Thus, during the design of a DSR artifact, conceptual models in subgroups (i) and (ii) above are used to create and enrich stakeholders' understanding of the problem domain and solution domain in a particular context. Also, models in sub-group (iii) are frequently referred to, and used, when grounding the design of a DSR artifact. This implies that, although this research focuses on the design of an artifact under the 'methods' category, it is cognizant of two aspects: first, the critical role of conceptual models in subgroups (i) and (ii) during the design process of an instance in the 'methods' category and, second, the critical role of models in subgroup (iii) and other types of DSR artifacts (i.e. in the categories of 'constructs' and 'instantiations') when grounding decisions taken during the 'build' activity of an artifact in the 'methods' category. Figure 1 demonstrates these roles and shows how the four categories of DSR artifacts (a) to (d) supplement each other.

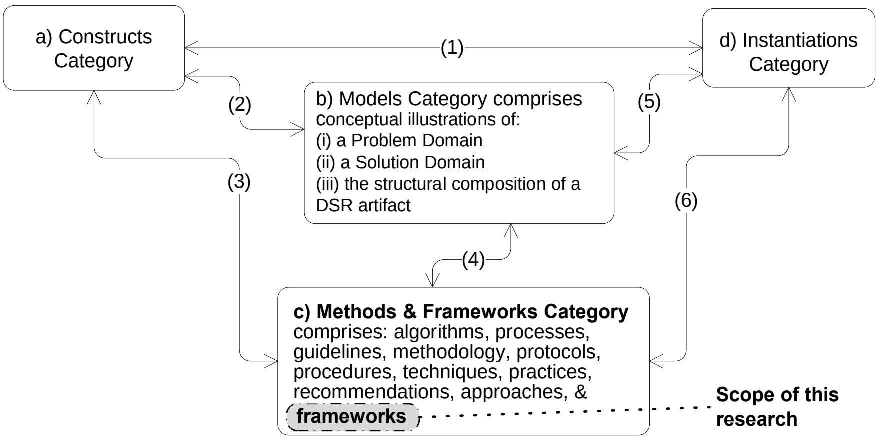

Figure 1. Types of DSR artifacts, implied associations, and the scope of this research

Based on views in March and Smith [1], Hevner et al. [2], Offermann et al. [3], Bucher and Winter [4], the associations represented by lines coded (1) to (6) with double arrowheads in Figure 1 indicate the following. (1) Constructs inform an instantiation, while an instantiation helps to implement constructs and the continuous evaluation of an instantiation in specific contexts helps to refine constructs. (2) Constructs make up a model of a specific context, while a composed model helps to represent the orchestration of a set of constructs and to refine constructs. (3) Constructs make up a method or a framework, while continuous evaluation of a method or framework helps to contextualize and refine constructs properly. (4) Model(s) represent a method or a framework, while continuous evaluation of a method or framework helps to refine model(s). (5) Model(s) represent and guide an instantiation, while an instantiation implements model(s). (6) A method or a framework guides or informs an instantiation, while an instantiation helps to implement a method or a framework, and the continuous evaluation of an instantiation in specific contexts helps to refine a method or framework.

Moreover, Figure 1 shows that the ‘methods’ category of artifacts is renamed herein as ‘methods and frameworks’ category. This arose from a critical reflection on the following existing definitions of a method. First, Hevner et al. [2] articulate that a method specifies processes which guide the search for a solution to solve a problem, and it can be in the form of an algorithm, or a textual description of a best practice, or a combination of them. Second, Winter [5] articulates that a method specifies processes or ‘procedural aspects’ for solving a problem and suggests specific results. Third, Offermann et al. [3] categorize various terms that researchers use when describing DSR artifacts. Our critical review of the categorization that they directly extracted from literature reveals that the following terms closely relate to the above two definitions of a method or framework, i.e.: algorithm, approach, framework, methodology, process, protocol, recommendation. Moreover, the Cambridge dictionary[†] defines a method as a “particular way of doing something”, while a framework as a “supporting structure around which something can be built”. The Merriam-Webster dictionary[‡] defines a method as a “systematic procedure, technique, or mode of inquiry employed” in a particular context, while a framework as a fundamental “structure of ideas”. Therefore, the above insights on the definition of a method and terms that closely relate to those definitions influenced the interpretation that is provided below of what is regarded as a ‘method’ and what is regarded as a ‘framework’ in the context of this research.

Herein, the ‘methods and framework’ category is perceived to include both a *high-level best practice* and a *detailed best practice* that prescribes an intervention for curbing a particular problem. The high-level best practice articulates ‘what’ should be done in an intervention (in terms of steps) and ‘how’ (in terms of tasks involved at each step). The detailed best practice not only specifies the ‘what’ and ‘how’, but also provides details on: ‘when’ to do the ‘what’ and ‘how’ (in terms of conditions), which means or tools to use when addressing the ‘how’, a classification of expected results, and other operational structures or capabilities required to deliver the ‘what’ and ‘how’ effectively and efficiently. In the context of this research, the high level best practice is what is regarded as a *‘method for guiding an intervention’*; and the detailed best practice is what is regarded as a *‘framework for guiding an intervention’*. This is elaborated by the declaration provided in the text box labeled ‘Methods and Frameworks’. Therefore, the research herein explores the extent to which DSR informs the design procedure of a framework, and ways in which the procedure can be enriched.

Methods and Frameworks: A method is perceived herein as a high-level description of a best practice for a specific context, while a *framework is a detailed description of a best practice for a specific context*. This implies that a framework is fundamentally an instance in the ‘methods’ category of DSR artifacts. However, for emphasis in specifying the scope of this research, the ‘methods’ category is renamed as ‘methods and frameworks’ category.

On the design procedure, Hevner et al. [2] and March and Smith [1] articulate that DSR in the Information Technology (IT) discipline involves *building* and *evaluating* four types of artifacts (constructs, models, methods, and instantiations), while Behavioral or Natural Science Research in IT involves *theorizing* and *justifying* the composition and functionality of these artifacts. This article delves into exploring existing support for the research activity of *building* DSR artifacts, specifically focusing on artifacts in the category of *‘methods and frameworks’* (and the sub category of *frameworks*) that guide the implementation of (digital) interventions against unstructured problems (as specified in Figure 1 and Figure 2). Since systems engineering approaches explicitly guide digital interventions for structured problems, the focus herein is put on *building* artifacts that can guide (digital) interventions against unstructured problems. The urgent need for researchers and practitioners to devise holistic and reliable digitally-enabled

† Cambrige dictionary, https://dictionary.cambridge.org/, Cambridge University Press & Assessment 2024.

‡ Merriam-webster dictionary, https://www.merriam-webster.com/, Merriam-Webster, 2024.

interventions motivates this research. The success of such interventions is often affected by the design or structural composition of the methods and frameworks used to guide their implementation. Thus, since design and creation research is risky if the researcher does not have the technical or artistic skills [6], exploring existing support for the activity of designing (in general) and building (in particular) such frameworks in DSR is prioritized in this paper. This is clarified in Figure 2.

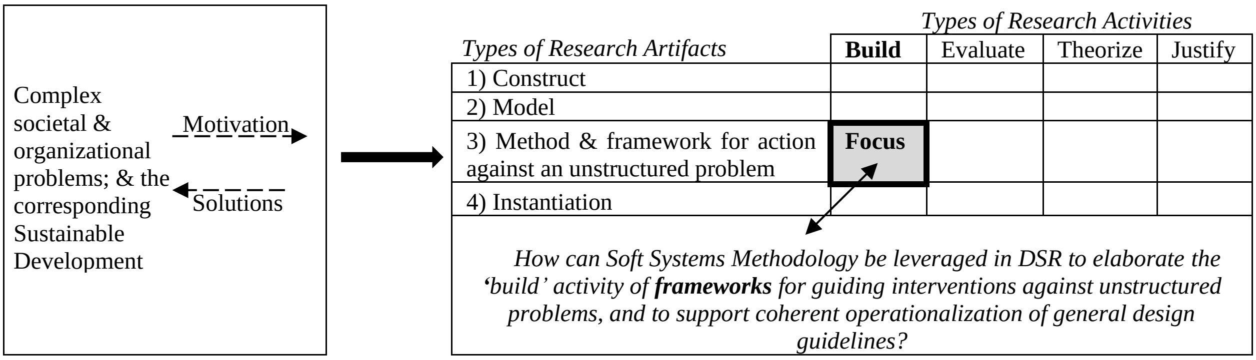

Figure 2. Framework for Research in IT (Adapted from [1])

The design activity in DSR comprises two major activities, i.e.: 'building' or creating an artifact and 'evaluating' it [8], [7], [1]. Figure 2 depicts that the research presented herein focuses on supporting only the 'build' activity in DSR. Existing support for the design activity (in general) and 'build' activity (in particular) of DSR artifacts or outputs in IT is classified herein into two categories. Category A entails foundational approaches (theories, frameworks, or processes) and guidelines or principles (propositions and proven best practices) that articulate the rationale and nature of the design cycle or 'build' activity in DSR efforts (e.g., [9], [10], [7], [2], [1]). Category B entails studies that articulate gaps in the foundational frameworks and principles (under category A), argue the need to strengthen the design activity of DSR, and devise amendments of artifacts in category A (e.g., [11], [12]). Despite existing efforts in categories A and B above, the design activity (in general) and 'build' (in particular) activity in DSR are still negatively affected by the following two issues.

The first issue is that the 'build' activity of DSR is implicit in aspects regarding an *elaborate procedure or logical thinking pattern* that can be followed when constructing or constituting the structural layout and composition of an artifact. Gacenga et al. [12] articulate that various frameworks or approaches of DSR are 'silent' about that design activity because they hardly provide concrete details of the design step. In an attempt to address the silence, Gacenga et al. [12] adopt the integrative Matching Analysis Projection Synthesis approach, which articulates that the design process comprises a macrocycle of three activities (i.e. analysis, projection, and synthesis), each of which iteratively involves a micro cycle of four activities (i.e. research, analysis, synthesis, realization). Unfortunately, the attempt to adopt this approach into DSR did not adequately inform the design process and thus hardly yielded the desired outcome [12]. In addition, Vom Brocke et al. [11] articulate the need to strengthen the quality of DSR outputs by prioritizing the notions of accumulation and evolution of design knowledge in DSR projects. An elaborate overview of existing design guidelines, which are articulated in form of principles, best practices, or perspectives on the design process of DSR is provided in Section 3.

The second issue is that several general guidelines exist for informing the design of DSR artifacts, but there is insufficient explicit support for their *harmonized adoption and coherent operationalization* during the execution of the 'build' activity in a particular DSR project. Yet, the coherent operationalization of existing design guidelines is not a trivial task. Consequently, when designing a specific artifact, researchers partially adopt and operationalize a small subset of the existing foundational or general design guidelines. This implies that the resultant designs or structural compositions of some artifacts have flaws that could have been avoided if researchers endeavored to subscribe to a considerable set of the existing general design guidelines. The inability to coherently operationalize design guidelines often hinders artifacts from fulfilling the intended purposes of their formulation. Consequently, if the type of artifact is a *'framework for guiding an intervention against an unstructured problem'* (which is the scope of this research as indicated in Figure 1 and Figure 2), the success and outputs of the intervention are adversely affected.

To address the above two issues, this research draws insights from Soft Systems Methodology (SSM), as shown in the last cell of Figure 2. SSM is a problem structuring method that supports holistic and rational analysis of complex unstructured problems in various organizational and societal settings [13], [14]. SSM is often used by practitioners and researchers to structure interventions that directly address specific organizational and societal problems (e.g., [15]–[18]). However, in this research SSM is adopted to elaborate the 'build' activity of the DSR approach. Thus, the target audience of this research is researchers in DSR and information systems. Although Baskerville et al. [19] introduce Soft Design Science Methodology by adapting SSM to accommodate DSR concepts (as elaborated in Section 3), the above two issues associated with the 'build' activity of DSR artifacts still prevail because the motivation of devising Soft DSR was not inclined towards elaborating the 'build' activity of DSR. Hence the following research question has been stated: *How can the strengths of SSM be leveraged in DSR to elaborate the 'build' activity of frameworks for guiding interventions against unstructured problems and to support coherent operationalization of general design guidelines?* Thus, this article demonstrates how SSM can supplement the 'build' activity in DSR by providing a **T**echnique of **B**uilding **F**rameworks for guiding **I**nterventions to curb unstructured problems (TBUFI).

Section 2 describes how Action Research was adopted in this study, Section 3 gives an overview of existing general design guidelines, Section 4 presents the structural layout and composition of TBUFI, Section 5 describes the detailed design of TBUFI with examples, Section 6 presents findings from evaluating TBUFI, and Section 7 concludes the paper.

2 Adopted Research Approach

To develop TBUFI, the Action Research method was adopted. The Action Research method is used in research efforts where the problem context and possible changes in it need to be analyzed in a social setting that enables interactions between researchers and subjects [20]. For such interactions to happen, action research comprises five stages, i.e.: 1) *Diagnosis stage*, which involves exploring the major cause of the need for change in a specific context; 2) *Action Planning stage*, which involves investigating and specifying the appropriate action that is to be undertaken to address the need for change in a specific context; 3) *Action Taking stage*, which involves operationalizing the chosen action in a specific context so as to realize the required change; 4) *Evaluation stage*, which involves investigating whether the required changes in a specific context were achieved by the implemented action; and 5) *Specify learning*, which involves using knowledge obtained from the strengths and weaknesses experienced in a particular intervention to improve a context or refine an artifact [20], [21]. These stages were executed in this study as highlighted below.

The *Diagnosis Stage* in this research involved specifying the major gap in the 'build' activity of a DSR project and justifying why the gap needs to be addressed. From the literature on the design cycle of DSR (as reported in Section 1 and Section 3) and from using DSR in particular research projects, two major issues were identified, i.e.: (a) lack of an elaborate procedure or logical thinking pattern for guiding the execution of the 'build' activity in DSR; (b) lack of guidance on the coherent operationalization of general design guidelines during the 'build' activity in DSR. This stage yielded two outputs, i.e.: (1) the above gaps in the 'build' activity of DSR and (2) a set of general design guidelines that need to be coherently operationalized during the 'build' activity (see Section 3).

The *Action Planning Stage* in this research involved continuously exploring how SSM techniques can be used to (a) elaborate the 'build' activity in a DSR project and (b) enable the coherent operationalization of design guidelines during the 'build' activity in a DSR project. The *Action Taking Stage* involved the actual use of TBUFI (from 2011 to 2023) to inform the 'build' activity of frameworks for guiding interventions for solving unstructured problems in ten research studies (as elaborated in Section 6). The Action Planning and Action Taking stages yielded the first version and transitional versions of TBUFI as the design procedure used to 'build' artifacts in the ten research studies (listed in Section 6). The *Evaluation Stage* in this research involved evaluating TBUFI in eleven iterations. The first 10 iterations are discretely reported in ten research studies (as specified in Section 6). The last iteration involved engaging information systems researchers in a group walkthrough meeting to deliberate the structural composition and layout of TBUFI (as elaborated in Section 6). The *Specify Learning Stage* involved articulating lessons learned from the 11 evaluation iterations and using them to refine the composition and layout of TBUFI as a technique for guiding the 'build' activity of a framework for guiding an intervention against an unstructured problem. These last two stages yielded the final version of TBUFI (as presented in Sections 4 and 5).

3 Existing Support for the Design Process in DSR Projects

This section gives an overview of the literature reviewed on notions that inform the design process (in general) and the 'build' activity (in particular) of DSR in Sections 3.1 to 3.3. The review herein focused on two types of notions. First, these are the notions that clarify the nature of the design process of DSR and the nature and purpose of particular types of DSR artifacts. They are summarized in Section 3.1. Second, the notions that specify general guidelines that need to be adhered to during the design process of DSR or when building or creating a DSR artifact; and such notions are summarized in Section 3.2. From these two categories, notions that are considered in Section 4 (and in Appendices 1 and 2) are coded to ensure their traceability in subsequent sections that discuss their operationalization. Notions in the first category are coded as DG0.x, and notions in the second category are coded as DGx, where x = {1,…., n}. The disaggregation of codes for specific notions is done based on the perceived similarity between aspects used to describe the notions. The codes are specified using **boldface** text in parentheses against a particular notion. Thereafter, Section 3.3 demonstrates the research gap and the significance of this research.

3.1 Positioning 'Methods' Artifact in the Taxonomy of Theory in Information Systems

March and Smith [1] present a two-dimensional framework that classifies and guides design research and natural or behavioral research in IT by specifying: i) four key types of research activities, which are *build*, evaluate, theorize, and justify (**DG0**); and ii) four key types of research outputs, which are constructs, models, *methods*, and instantiations that characterize IT research (**DG0.1**). These notions are adopted herein to articulate the theoretical scope of this research and the type of output expected from this research. Thus, this research focuses on delving into the 'build' activity of artifacts, which can be perceived as frameworks for directing actions or interventions against unstructured problematic occurrences in a specific organizational or societal setting. Notion DG0.1 is elaborated in Gregor [22] where the 'methods and frameworks' category of artifacts is perceived to be an instance of the *'design and action-oriented'* theory, which is one of the five categories of theory in information systems research, i.e. (A) Artifacts on *design and action* – providing *specific prescriptions on the composition of an artifact in form of methods, techniques, principles of form and function* (**DG0.2**); (B) Artifacts for analyzing facts in a context; (C) Artifacts for explaining *the how, why, when, and where* details of facts in a context; (D) Artifacts for predicting *what is* and *what will be* aspects of facts in a context; and (E) Artifacts for explaining and predicting details of *what is, how, why, when, where, and what will be* aspects of facts in a context.

In addition, Gregor and Jones [23] disaggregate artifacts in category A above into two subcategories, i.e. (A1) Theories or abstract artifacts in the form of constructs, models, methods, and principles (**DG0.3**); (A2) a material artifact derived from instantiating a theory, which could be in form of an instantiated product or method. Furthermore, Gregor [24] also refers to artifacts in category A above as Design Theory, which refers to *normative or prescriptive guidelines or principles that can inform a given practice by providing knowledge on* (i) Process of building an artifact, such as methodologies and tools for developing information systems (**DG0.4**); and (ii) Design principles or explicit specifications (using natural language or diagrams) of *'what an artifact should look like when built'* or design decisions and design knowledge that shape an artifact (**DG0.5**). A closer look at these notions reveals that notion DG0.3 maps notion DG0.2 to notion DG0.1, and notions DG0.4 and DG0.5 elaborate notion DG0. This justifies why notions DG0.2 to DG0.5 are also adopted herein – to clarify the theoretical scope and the type of output expected from this research. Thus, this research is expected to provide an explicit procedure that a researcher can follow to build or create a framework that exhibits 'prescriptive' or 'normative' characteristics that can enable it to be regarded as an instance of the 'design and action' type of theory or 'design' type of theory in information systems research. Section 3.2 provides an overview of existing insights that are adopted herein to inform the development of such a procedure.

3.2 Existing Guidelines for Designing DSR Artifacts

Vom Brocke et al. [11] regard artifacts classified under category A (in Section 3.1) as *'design knowledge'* and articulate the need for researchers to facilitate the accumulation and evolution of design knowledge across DSR projects by adhering to the following four principles. The first principle is *aligning,* which articulates the need for a researcher to specify how the design process used in a specific DSR project progressed or evolved by ensuring that the design process is explicitly documented and justified (**DG1**). The second principle is *positioning,* which articulates the need for a researcher to clearly specify the subsets of the problem space and solution space that a DSR project contributes to by articulating (i) the relevant problem that motivates the research, (ii) the devised solution, and (iii) substantial evaluation evidence that depicts the relationship between the problem and the solution (**DG2**). The third principle is *grounding,* which articulates the need for a researcher to explicitly state the extent to which a specific DSR project builds on prior knowledge by articulating how the project supports the accumulation and evolution of knowledge by explicitly documenting the processes used and results obtained in the search for existing propositional and prescriptive knowledge that is relevant to the problem in a particular DSR project (**DG3**). The fourth principle is *advancing,* which articulates the need for a researcher to specify how results from a complete DSR project augment or extend or improve prior *'design knowledge'* (**DG4**). These notions are adopted herein and used to derive requirements that TBUFI must address if it is to operationalize these guidelines along with other guidelines, so as to serve its intended purpose of elaborating the activity of building of frameworks for action (see Appendix 1).

In addition, Hevner et al. [2] articulate seven guidelines that a DSR project must adhere to, all of which have a direct bearing on the general design process of an artifact: 1) *Design as an artifact,* where the ultimate result of a DSR project must be a feasible artifact, which could be in form of a construct, model, method, or an instantiation (**DG0.1**); 2) *Problem Relevance,* where a DSR project must focus on developing a technology oriented solution for an important organizational or societal problem (**DG5**); 3) *Research Contributions,* where the ultimate result of a DSR project must be a confirmable contribution to at least one of the existing knowledge areas, which include foundational theories or approaches, evaluation methods or methodologies, and design products or processes (**DG6**); 4) *Research Rigor,* where a DSR project must apply rigorous approaches when constructing an artifact (**DG7**) and when evaluating an artifact; 5) *Design as a Search Process,* where a DSR project must ensure that an artifact effectively achieves its purpose by using appropriate means and acceptable practices in the problem environment (**DG8**); 6) *Communication of Research,* where a DSR project must endeavor to effectively communicate its ultimate result to a "technology oriented" audience and "management oriented" audience (**DG9**); and 7) *Design Evaluation,* where a DSR project must rigorously exhibit the quality, utility, and efficacy of an artifact by effectively executing appropriate evaluation methods (**DG8.1**). The first guideline was earlier coded as DG0.1, while guidelines 2 to 7 are adopted herein and used to derive requirements that TBUFI must address if it is to operationalize these guidelines along with other guidelines or notions for the design process (see Appendix 1).

Furthermore, Hevner (in [7], [8]) demonstrates the following 3 cycles that constitute a DSR project, all of which directly inform the 'build' activity of an artifact:

- First is the *Relevance Cycle,* which depicts two perspectives: (i) *Identified need,* where a DSR project is initiated by an identified need or problem in a particular environment and acceptance criteria for assessing the suitability of the desired solution (**DG10**); and (ii) *Iterative Field Testing,* where a designed artifact must undergo iterative field testing in the application environment to ascertain deficiencies in its quality attributes (**DG8.2**).
- Second is the *Rigor Cycle,* which depicts two perspectives (i) *Skillful adoption,* where a DSR project must have a clearly defined knowledge base comprising foundational artifacts (i.e., theories, methods, experiences and expertise, and existing design products or processes), from which a researcher must skillfully select appropriate artifacts and insights and apply them during the construction of a new artifact (**DG11**) and its evaluation; and (ii) *Knowledge contribution,* where the ultimate result can be regarded as a contribution to the knowledge base if it extends an existing artifact, or if it offers new experiences or expertise, or if it is a new artifact or design process (**DG8.3**).
- Third is the *Design Cycle,* which depicts two perspectives (i) *Iterative Creation,* where a DSR project involves the *"hard work"* of iterative building or creating an artifact (by drawing its requirements from the relevance cycle and drawing its foundational insights from the rigor cycle) (**DG12**); and (ii) *Rigorous Evaluation,* where an artifact undergoes rigorous evaluation in a controlled environment until it is ready for field testing, and its performance is assessed in the application environment, prior to considering its inclusion in the knowledge base (**DG8.4**). To ensure that other existing design guidelines are coherently operationalized in a way that subscribes to these cycles, notions **DG10 to DG12** are adopted herein and used to derive requirements that TBUFI must address (see Appendix 1).

Hevner et al. [2] also indicate the design process needs to involve (1) *Generation and testing design alternatives,* which includes iteratively searching for an effective solution by executing the *'generate-test cycle'* of Simon [25], in which design alternatives are first generated and then tested against a set of requirements or constraints to determine a satisfactory one (**DG13**); and (2) Conceptually representing three aspects of the problem and solution domains of a DSR project in a creative and innovative way. These aspects are *Means* or decision variables, which depict the range of accessible actions and resources that can be leveraged in devising a desired solution (**DG13.1**); *Ends,* which depict objectives or requirements and constraints that a desired solution should address (**DG13.2**); and *Laws,* which depict intractable forces in the problem and solution domains (**DG13.3**).

The concept of design alternatives is properly demonstrated by Simon [25], where design is treated as a critical phase in the generic process of problem-solving and decision making, which includes: a) Intelligence phase, where a problem environment is comprehensively examined to determine and understand the need for intervention; b) *Design phase, where a decision maker needs to devise potential courses of action or potential decision alternatives for addressing the specified need in the problem environment* (**DG13.4**); and c) Choice phase, where a particular course of action or decision alternative is determined and chosen. Moreover, Nunamaker et al. [26] emphasize that design in the context of an information system is an engineering concept that requires one to *understand the domain of interest, apply relevant scientific and technical knowledge, devise several alternatives, explore and evaluate the proposed alternatives, make final design decisions, and synthesize them* (**DG13.5**). These notions provide critical insights about the actual process of creating or constructing an artifact. Thus, to ensure that other design guidelines are operationalized with respect to the foundational norms of design, these notions are adopted herein and used to derive requirements that TBUFI must address (see Appendix 1).

Baskerville et al. [19] also present the Soft DSR methodology, which comprises the following seven activities that enable one to consider social aspects when developing technological solutions: (1) identify and describe a specific problem context (**DG5.1**); (2) express the specific problem as an explicit set of (contextual) requirements (**DG10.1**); (3) systemically abstract and translate the obtained set of (contextual) requirements into a general problem that has technical and social dimensions, which is obtained by deriving classes or types of problems from the specific issues or challenges owned by a client or as observed in a particular problem context (**DG10.2**); (4) use systems thinking to derive a general solution design or class of solutions for the general problem, and use imperative logic to express it as a set of general design requirements (**DG10.3**); (5) compare the general design requirements obtained in the fourth activity with the specific problem or contextual requirements that were obtained in the second activity to ensure appropriateness (**DG10.4**); (6) conduct a declarative search for specific elements or components that can be considered to be feasible instance of a solution that addresses the general design requirements that are confirmed in the fifth activity (**DG8.5**); and (7) construct the identified component into a specific solution and deploy it into the social system to ensure that it addresses the specific problem that was identified in the first activity (**DG8.6**). These activities are adopted herein as design guidelines and treated as detailed requirements that TBUFI must address in addition to the high level requirements that are specified in Appendix 1.

From the context of user interface design or human computer interaction, Galitz [27] presents various principles for designing user interfaces, some of which include: 1) *Usability,* which articulates the need to ensure that a design or interface can be easily or independently used by an individual; and that an individual can effectively or independently use an interface to accomplish a specific task (**DG14**); 2) *Learnability,* which focuses on ensuring that a first time user of a design or interface can accomplish elementary tasks (**DG14.1**); 3) *Efficiency,* which focuses on ensuring that a user who has understood or is knowledgeable about a design or interface can promptly accomplish tasks (**DG14.2**); 4) *Satisfaction,* which articulates the need to ensure that a user is pleased to use a design or interface (**DG14.3**); and 5) *Consistency,* which focuses on ensuring that related elements on an interface have a similar look (**DG14.4**). Moreover, the ISO [28] articulates the need to prioritize the concept of *Traceability* in product designs, which is the *"ability to trace the history, application, or location of that which is under consideration"* (**DG15**). This implies that in the documentation of an artifact, a target user should be able to trace or verify the source of elements, the location of elements, and the application of elements (or disaggregation or aggregation of elements) that constitute the design of an artifact across various views or levels of granularity in the design of an artifact. These notions are adopted herein and used to derive requirements that TBUFI must address if it is to ensure that existing design guidelines are operationalized with respect to insights from designs of products and prototypes (see Appendix 1).

3.3 Gap Analysis and Research Significance

Existing general design guidelines that need to be coherently operationalized during the 'build' activity of an artifact are coded as DG0 to DG15 in Sections 3.1 and 3.2. However, guidelines coded as DG0.x are stated in a way that assumes that the execution details of the 'build' activity are mutually understood by researchers at various levels of experience across the contexts. Yet, this is not the case, as specified in Section 1. This implies that notions or guidelines coded DG0.x underline the need to elaborate the 'build' activity of DSR. Moreover, it is logically envisioned that undertaking the effort of addressing guidelines coded DG0.x by elaborating the 'build' activity of DSR, simultaneously addresses the issue of coherent operationalization of the other design guidelines coded DG1 to DG15 and vice versa. This justifies why this research prioritizes the need to elaborate the 'build' activity of a DSR artifact, particularly starting with a framework for guiding the implementation of an intervention to curb an unstructured problem. Figure 3 provides an adaptation of the DSR frameworks in Hevner et al. [2] and Hevner [7] that is devised to demonstrate the significance of this research.

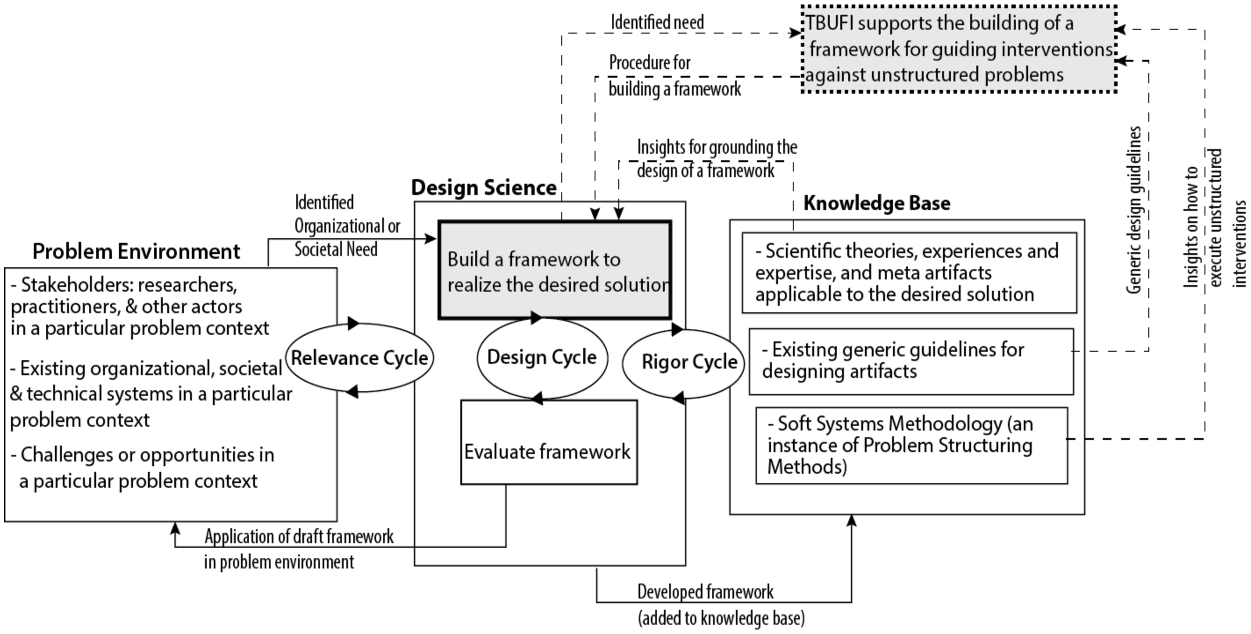

Figure 3. Instantiation of the DSR Framework to specify the significance of this research and the role of TBUFI as a Supplement for the 'build' activity of DSR

The left part of Figure 3 shows that the problem environment (which is a specific organizational or societal context) triggers the 'build' activity of a DSR project by specifying an identified problem that is unstructured in nature. The gray-shaded box in the middle part of Figure 3 shows that addressing the unstructured problem requires a researcher to build a framework that will guide the implementation of a desired solution. The desired solution is often perceived as an intervention that comprises several countermeasures that are deemed appropriate and must be implemented in a harmonized way if they are to address the identified problem effectively. Guidelines coded DG11 and DG12 in Section 3 and Appendix 1 articulate two major expectations of what should transpire in the gray-shaded box in the middle part of Figure 3. These include the following:

- Guideline DG11 demands that a researcher *"skillfully selects and applies"* appropriate approaches from the knowledge base during the construction of a framework.
- Guidelines DG12 demands that a researcher delves into the *"hard work"* of constructing a framework.

However, explicit *'execution details'* associated with the breadth and depth of the *hard work trait* in DG12 and *skillful trait* in DG11 are scarcely available in the existing literature. Thus, this research argues the need for a technique, abbreviated as TBUFI – that can provide an elaborate thinking pattern which can be followed when building a framework for guiding an intervention to curb an unstructured problem. This is indicated by the gray-shaded box with a dashed boarder at the top right corner of Figure 3. Prior to devising TBUFI, its requirements are specified in Appendix 1. To address the requirements it was necessary to explore how techniques of the SSM can be leveraged to operationalize guidelines coded DG1 to DG15. This is indicated by the two upward-facing dashed arrows on the right side of Figure 3. The design of TBUFI is presented and discussed in Sections 4 and 5.

4 Structural Layout of TBUFI

An overview of stages and techniques in SSM is presented in Section 4.1. Thereafter, Section 4.2 discusses how SSM stages and techniques are adapted to address requirements for TBUFI (that are specified in Appendix 1 and elaborated in Section 3.2). Section 4.3 then presents the hierarchical view of TBUFI as an *elaborate logical thinking pattern* that delineates the magnitude and execution details of the 'build' activity of a framework for guiding an intervention against an unstructured organizational or societal need or problem.

4.1 Adopted SSM Stages and Techniques

SSM comprises 4 stages that are executed using 6 techniques [13], [14], [29], which include the following:

- ***SSM-stage 1.*** *Study a problem situation and the implied desired action* using:
  a) *Rich Picture technique* to create an understanding of contextual issues in a specific setting;
  b) *Analysis One Two Three technique* to enable understanding of interdependences among stakeholder roles and perceptions (*Analysis One*), understanding of social and cultural issues in a setting (*Analysis Two*), and understanding of political facilitators in a setting (*Analysis Three*). Herein, the interpretation and corresponding adaptation of concepts in stage 1 is specified in Section 4.2.
- ***SSM-stage 2.*** *Devise 'purposeful activity models' that describe the desired state* using:
  a) Root Definitions technique, which involves defining short phrases that rationally articulate the required actions (or transformations to realize the desired state) and their significance, using the specific pattern of "**Do P** (action to take) **by Q** (how to implement action) **to resolve R** (issue to be addressed by the action);
  b) *CATWOE Analysis technique*, which involves creating understanding of context by defining *Customers* (or stakeholders to be affected by a transformation), *Actors* (or key implementers in a transformation), *Transformation processes* (major processes to be executed in a transformation), *World view* (perspectives on the effectiveness of a transformation), *Owners/sponsors* (facilitators/funders to control a transformation), and *Environmental constraints* (external issues likely to affect a transformation) associated with the desired state;
  c) *Purposeful Activity Models technique*, which involves illustrating how the relevant transformation processes that are associated with the desired state can be assembled and their corresponding monitoring and control measures using a purposeful activity model;
  d) *Multi-level thinking technique*, which involves managing complexity by enabling stakeholders to intentionally structure their views in levels, so as to separately articulate the *whether-why* issues from the *what-how* issues. The interpretation and corresponding adaptation of concepts in stage 2 is specified in Section 4.2.
- ***SSM-stage 3.*** *Debate models that describe the problem situation and feasibility of desired changes*, using them as insightful instruments that trigger deliberations, increase contextual and holistic understanding of aspects, and provide insights into ways of addressing conflicting views. The interpretation and corresponding adaptation of concepts in stage 3 is specified in Section 4.2.
- ***SSM-stage 4.*** *Specify and implement appropriate actions* to resolve the organization or societal problem. The interpretation and corresponding adaptation of concepts in stage 4 is specified in Section 4.2.

4.2 Interpretation and Adaptation of SSM Stages and Techniques

The procedure of designing TBUFI involved customizing or contextualizing concepts in the above SSM stages to demonstrate how the requirements (presented in Appendix 1) could be addressed, so as to coherently operationalize general design guidelines DG1 to DG15 (discussed in Section 3.2). The contextualization or adaptation was done based on the following four interpretations.

First, concepts in SSM-stage 1 are interpreted to invoke the creation of a conceptual model about the problem domain, which can be informal or formal conceptual models that can be created using the two specified SSM techniques and enriched using insights and visualizations or expositions from other (collaborative) modeling approaches. In addition, since multiple key stakeholders are involved in a problem context, the conceptual models about the context help to align or bridge multiple (implicit) understandings of the key stakeholders. Stakeholder perspectives or understandings can be elicited using the various stakeholder involvement approaches. In addition, Analysis One Two Three can be broadly perceived as the 'Categorical Analysis' of all issues in the problem domain. This means that it can go beyond the specified three SSM aspects (i.e., One, Two, Three) so as to cover other existing and emerging classifications in a specific problem context or broader problem domain. Thus, basing on this interpretation, concepts in stage 1 of SSM were adapted herein to derive task T1 of TBUFI (as presented in Figure 4).

Second, concepts in SSM-stage 2 are interpreted also to invoke the creation of conceptual models about the desired situation, which can be created using the four specified SSM techniques and enriched using insights and visualizations or expositions from other (collaborative) modeling approaches. In addition, the Multi-level thinking technique can invoke the iterative use of the other three techniques in stage 2 (i.e., Root Definitions, CATWOE analysis, and Purposeful Activity Models) in various levels of granularity depending on the complexity of issues in a specific problem context and potential solution aspects in a specific solution domain. In other words, interpreting the Multi-level thinking technique in the context of design guidelines DG10, DG11, DG12, DG8, and DG15 (as adapted in Appendix 1) reveals at least four levels of thinking where the three techniques need to be rationally applied. These include:

- level 1 of formulating requirements (to invoke DG10),
- level 2 of generating design alternatives (to invoke DG11 and DG13),
- level 3 of elaborating and assessing design alternatives (to invoke DG8 and DG13),
- level 4 of specifying and operationalizing design decisions (to invoke DG5 along with other design guidelines).

Thus, based on this interpretation, concepts in stage 2 of SSM were adapted herein to derive tasks T2 to T5 of TBUFI (as presented in Figure 4).

Third, concepts in SSM-stage 3 are interpreted to invoke a deliberation on the accuracy of conceptual models about the problem situation and the feasibility of conceptual models about the desired situation. The deliberation of conceptual models is inclined and envisioned toward expressing and communicating aspects of the problem situation and aspects of the desired situation in a way that yields a shared understanding of those aspects among key stakeholders. The need for insights that confirm the accuracy and feasibility of the derived conceptual models helps to enrich the design (layout and composition) of an artifact. Thus, based on this interpretation, concepts in stage 3 of SSM were adapted herein to enrich task T4 of TBUFI (as presented in Figure 4).

Fourth, concepts in SSM-stage 4 are interpreted to invoke a critical task of clearly articulating specific actions to be operationalized so as to achieve the desired situation, and how they need to be operationalized if they are to achieve their intended purpose. This implies the need to provide a clear understanding of constraints and situational factors associated with the effective adoption and implementation of specific components of an artifact, such as a framework for directing action toward a desired situation. Depending on available resources, such constraints and situational factors can be investigated by prototyping and experimenting digital solution(s) or other components prescribed in the framework. The effort to investigate the situational aspects associated with the effective and efficient operationalization of particular design decisions that constitute a framework helps to inform and ensure the completeness of the design (layout and composition) of an artifact. Thus, concepts in stage 4 of SSM were adapted herein to enrich tasks T4 and T5 (as presented in Figure 4).

4.3 Hierarchical View of TBUFI

The structural layout of TBUFI is presented in a hierarchical style using views that present four different levels of granularity, in order to properly demonstrate the coherent nature of tasks and sub tasks derived from the interpretations and implied adaptations that are articulated in Section 4.2. The text box labeled 'TBUFI views' highlights the purpose of each view.

TBUFI Views:

- **Figure 4 is the high-level view (or Task view) of TBUFI:** This presents the major tasks that constitute TBUFI, the relationships between and among them, and the supporting contextual, operational, and foundational aspects or elements. This addresses the issue of 'what' should be done and high-level aspects of 'how' the 'what' can be accomplished when building a framework for guiding an intervention against an unstructured problem.
- **Figure 5 is the detailed-level view (or Task and Sub Task view) of TBUFI:** This presents the tasks and subtasks of TBUFI. It provides detailed-level aspects of 'how' tasks in the high-level view can be accomplished when building a framework for guiding an intervention against an unstructured problem.
- **Appendix 2 is the operational guidance view of TBUFI:** This shows the tasks and subtasks in TBUFI, design guidelines that are operationalized when executing specific subtasks, and expected outputs. It addresses the issue of 'why' tasks and subtasks need to be executed and 'when' they need to be executed in order to achieve the expected result.
- **Detailed narrative view of TBUFI – presented in Section 5:** This describes 'how' the tasks and subtasks of TBUFI need to be executed. It provides insights into which techniques to use when executing specific tasks and 'why' those techniques are used.

As shown in the text box "TBUFI Views:", high-level tasks that constitute TBUFI are summarized in Figure 4 and elaborated in Figure 5. Figure 4 illustrates that the adaptation of SSM stages during the 'build' activity of DSR yields 5 major tasks to which codes T1 to T5 are assigned for the purpose of traceability. Specific SSM stages that motivate the formulation of each task are indicated in small gray-shaded boxes that are discretely presented on the left and right side of Figure 4, with dotted arrows (pointing from them to the boxes of tasks T1 to T5) that depict the interpretations specified above.

In addition, the top left side of Figure 4 shows that task T1 is initiated by a specific organizational or societal need or problem identified in a problem environment of a DSR project. The hexagon symbols (on the right side of Figure 4) represent the knowledge bases of the problem space and solution space in a DSR project, and the dashed arrows pointing to them and from them indicate information flows between the knowledge base and tasks T1, T2, T3, and T5. Specifically, the execution of task T1 is informed by insights on modalities of classifying issues, that are drawn from the knowledge base of the problem space in a DSR project. The right side of Figure 4 shows that task T2 interacts with the knowledge base of the solution space in a DSR project to draw insights on potential solutions or actions that can be taken to address specific requirements and problematic issues. In addition, task T3 interacts with the knowledge base on the solution space to draw insights on existing approach(es) that can be used to support the orchestration or synthesizing of design decisions or choices taken from the assessment of the potential actions that are generated in task T2. The bottom left side of Figure 4 shows that task T4 interacts with the evaluation activity of a DSR project to obtain evaluation findings on the draft versions of the design of a framework or method. Lastly, the bottom right side of Figure 4 shows that task T5 intersects with the solution space by effectively communicating the developed framework or method, as an addition to the knowledge base of the solution space in a DSR project.

Tasks T1 to T5 in Figure 4 are disaggregated into subtasks T1.1 to T5.4, as shown in Figure 5. Appendix 2 shows the particular subtasks of TBUFI that address specific design guidelines and the major outputs expected from tasks T1 to T5.

Tasks T1 to T5 that constitute the thinking pattern of TBUFI are all high-level mandatory tasks that help to ensure that the creation procedure of a framework or method is elaborate and repeatable. However, depending on the nature and scope of the problem and desired solution, the optional aspects in TBUFI are the subtasks and techniques for executing them. Thus, the subtasks of TBUFI (that are mandatory) are presented in the gray-shaded boxes in Figure 5. This implies that the gray-shaded boxes in Figure 5 represent a mandatory pathway that can be undertaken by a researcher in case the context cannot allow the execution of all subtasks of TBUFI. All tasks and sub tasks in Figure 5 are subsequently explained in Section 5.

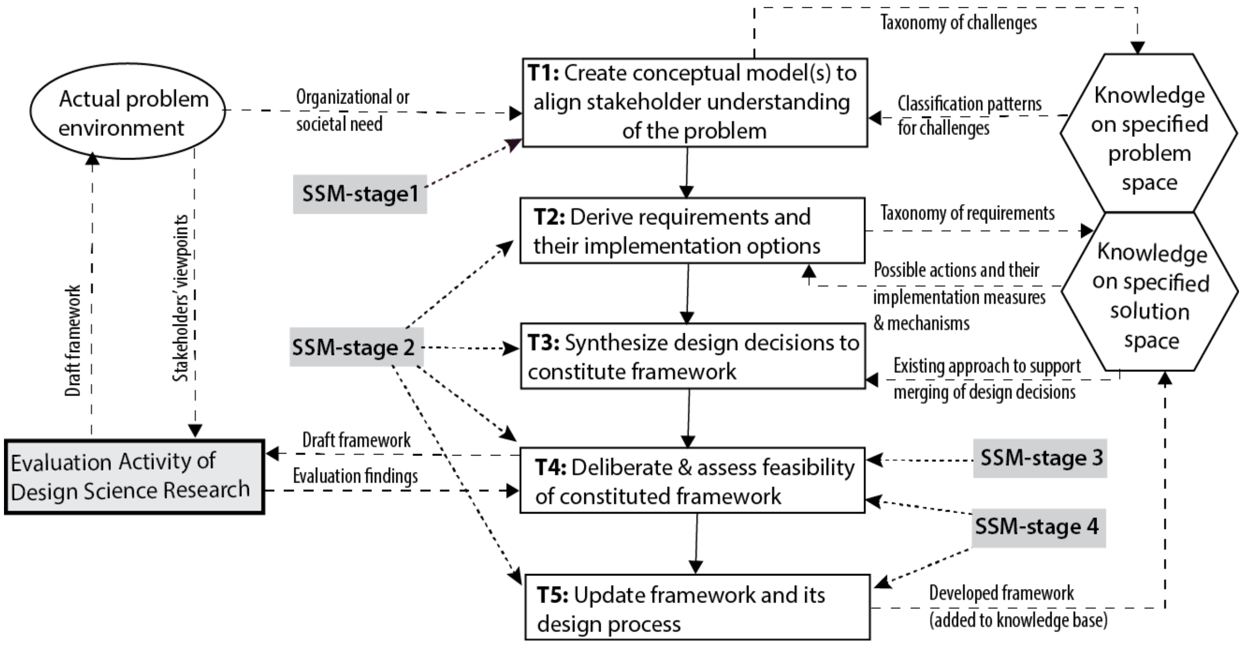

Figure 4. The Structural composition and layout of TBUFI

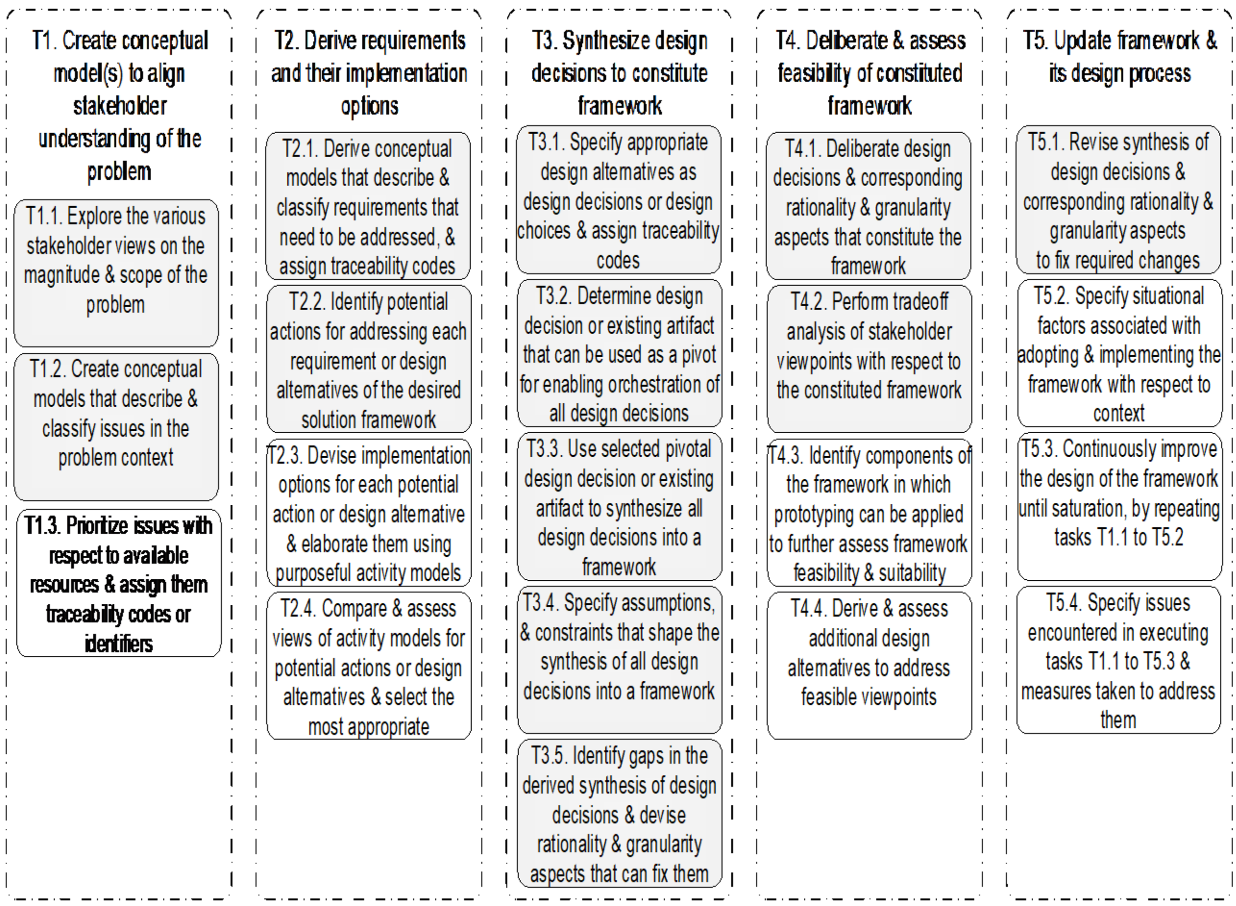

Figure 5. Decomposed tasks of TBUFI (gray-shaded boxes indicate mandatory sub tasks)

5 Detailed Narrative View of TBUFI

This section details the contextual, operational, and foundational aspects that are associated with the tasks and subtasks of TBUFI, and SSM techniques (as well as other techniques) that can support their execution. It also justifies why some subtasks have been considered as mandatory or optional. Moreover, to clarify some aspects (or terminologies) used in the detailed narrative of TBUFI, excerpts of examples are discretely presented using text boxes labeled 'Example X1' to 'Example X5' in Sections 5.1 to 5.5. Examples X1 to X5 in text boxes that are presented in Sections 5.1 to 5.5 are drawn from a real-world societal problem. Due to space limitations, a full case of detailed examples cannot be demonstrated in this article.

5.1 Create Conceptual Model(s) to Align Stakeholder Understanding of the Problem (T1)

Figure 5 shows that task T1 comprises subtasks T1.1 to T1.3, which are elaborated in Sections 5.1.1 to 5.1.3. Design guidelines that are operationalized during the execution of tasks T1.1 to T1.3 are specified in Appendix 2.

5.1.1 Explore the Various Stakeholder Views on the Magnitude and Scope of the Problem (T1.1)

The creation of a framework that directs an intervention needs to be initiated by a comprehensive analysis of stakeholders' perspectives on issues that characterize the extent and complexity of the organizational or societal problem or need. This explains why task T1.1 is a mandatory task. The comprehensive analysis is informed by insights from two sources, i.e. a) evidence obtained from stakeholders by conducting an exploratory survey, conducting collaborative stakeholder engagements that are supported by various existing groupware solutions, or participating in specific scenarios in the problem environment and b) literature that delineates synthesized and analyzed stakeholder perspectives on the urgency of the problem or need in a specific problem space and/or solution space of a project. These sources depict the 'world views' on the problem context (which is an adaptation of 'world views on the desired state' in SSM's CATWOE analysis techniques). Task T1.1 can be executed using the Rich Picture technique from SSM. Examples of Rich Pictures can be found in [30] and [31]. Additional techniques include cause-and-effect analysis models such as the Ishikawa Diagram [32]–[34], Problem Tree [35], and various systemic thinking models or conceptual modeling approaches. Moreover, Group Support Systems such as MeetingWizard[§] can be used to provide an interactive platform for stakeholder engagements when collecting information that is used as input for defining the magnitude and scope of the problem.

5.1.2 Create Conceptual Models that Describe and Classify Issues in the Problem Context (T1.2)

To trigger a comprehensive analysis of the problem or need, it is vital to derive (informal) conceptual models that describe the problem domain and classify issues into an insightful taxonomy that underpins stakeholders' understanding of the problem context (that is specified in task T1.1). The taxonomy helps to abstract and translate specific issues into general problems that can trigger holistic and rational thinking (as supported by guideline DG10.2 in Section 3). The conceptual models or insightful taxonomy of issues helps to invoke rational thinking among stakeholders in a given context in order to trigger comprehensive analysis of problem context. The comprehensive analysis helps to continuously cultivate a shared understanding among stakeholders, on intricacies associated with issues in a particular context. This justifies why

[§] MeetingWizard via https://www.meetingwizard.nl/en/how-it-works/

task 1.2 is regarded as a mandatory task. Task T1.2 involves conducting a thematic analysis of issues characterizing the problem to guide their aggregation and disaggregation. Thereafter, visualizations or conceptual models that represent the aggregation and disaggregation of issues are derived. Task T1.2 can be executed using the Analysis One Two Three technique of SSM (mentioned in Section 4), which prompts for the recognition of stakeholder roles and perceptions (as adopted in task T1.1) and the corresponding social, cultural, and political issues as well as underlying dynamics thereof. Besides, issues can be classified using dimensions in an existing framework, or an already existing classification approach in the problem space or solution space of a project. Like in task T1.1, the conceptual models that classify issues can be enriched and validated by engaging stakeholders using Group Support Systems. *In the text box of Example X1, the challenges are classified into governance issues and technology implementation issues.*

Example X1. Scenario of an unstructured problem: Several community-level development efforts and community-level welfare initiatives across all districts in country Y are inadequately coordinated and unsuccessfully implemented. Thus, there is a need to understand the specific challenges and the categorical challenges that shape the problem context.

Governance-related challenges (C1):
- C1.1. Some community members are not aware of social problems faced by other members within their community; & willing helpers in the community do not know those who are in need of their help, and vice versa.
- C1.2. Important information on undesirable incidents in the community does not reach authorities in time; & information from authorities does not reach target beneficiaries or audiences within a community.
- C1.3. Low awareness on how authorities could benefit from the expertise and experiences of various community members during the implementation of the several community development projects.

Technology implementation related challenges (C2):
- C2.1. Community-level radio stations offer information services to only members who are within their proximity.
- C2.2. Low awareness on how digital technologies can be leveraged to effectively improve information sharing and increase stakeholder participation in solving problems encountered in specific communities.

5.1.3 Prioritize Issues with respect to Available Resources and Assign them Codes or Identifiers (T1.3)

This involves scoping the problem in a particular context, by prioritizing and specifying a subset of challenges that need to be addressed with respect to available resources and the urgency of particular issues in the problem context. Task T1.3 can be considered as an optional task for some researchers. This is because the classification in task T1.2 may dissolve some challenges through aggregation and disaggregation of aspects, so as to ensure that the conceptual model of the problem domain and any underlying taxonomy only reveal or focus on issues that are critical in a particular context. This implies that task T1.3 can be executed along with task T1.2 in some contexts. To enable traceability of how problematic issues in the problem domain are addressed in the desired solution (in line with guideline DG15 in Appendix 2), the classification in task T1.1 and the prioritization in task T1.2 is accompanied by assigning unique codes or identifiers of each prioritized issue in the conceptual model or taxonomy (e.g., Cx as shown in text box of example X1). Similarly to tasks T1.1 and T1.2, task T1.3 can be supported by using Group Support Systems to engage stakeholders to specify their perceived priorities of issues in the problem context, which are then used as input to generate group priorities of specific issues. In the text box of Example X1, the gray-shaded problem (coded C2.2) is chosen as the core or top priority issue that must be addressed in that scenario. This is because efforts to directly address issue C2.2 indirectly address all other issues listed in example X1.

5.2 Derive Requirements and their Implementation Options (T2)

Figure 5 shows that task T2 comprises subtasks T2.1 to T2.4, which are elaborated in Sections 5.2.1 to 5.2.4. Appendix 2 shows design guidelines that are operationalized during the execution of tasks T2.1 to T2.4.

5.2.1 Derive Conceptual Models that Describe and Classify Requirements that Need to be Addressed and Traceability Codes Assign to (T2.1)

Similarly to task T1.2, this task (T2.1) involves deriving conceptual models that describe the solution domain(s) associated with the desired state. As justified in preceding sections, these models underpin stakeholders' understanding of the envisioned desired state, which then informs the generation or formulation and classification of requirements that need to be addressed by the framework that will guide the desired intervention. The conceptual models are expected to reveal any underlying taxonomies of requirements, which are declarations of what needs to be done to address the priority issues from task T1. Taxonomies of requirements can be revealed by leveraging an existing taxonomy or artifact identified from the solution space and/or problem space of a particular project. The conceptual models and underlying taxonomies of requirements allow comprehensive analysis and enable aggregation and disaggregation of requirements, in order to create shared understanding among stakeholders on requirements that must be fulfilled to achieve the desired state. Besides, a well-formulated taxonomy of requirements provides a solid foundation for devising appropriate potential actions to address them. This is why task T2.1 is regarded as mandatory.

Task T2.1 can be executed by using the Multi-level thinking technique and Root Definition technique (as defined in Section 4). Adopting the Multi-level thinking technique in task T2.1 encourages researchers to determine whether pursuing a particular goal will help address the problem and why it is considered appropriate. In addition, adoption of the Root Definition technique in the context of task T2.1 focuses on specifying aspects for 3 dimensions that are depicted from the 'Do P by Q to resolve R formula' (defined in Section 4). The dimensions are: **'the action to undertake'** *(which is the ultimate goal to achieve)*, using **'specific means'** *(which are the requirements that must be fulfilled)*, so as to address **'specific challenges'** *(which are articulated in task T1.2)*. This is clarified in the text box of Example X2.

This process yields a set of requirements that a specific framework should fulfill, with each derived requirement assigned a code to (e.g., Rx as shown in Example X2). This is done to ensure traceability between the requirements and issues that are assigned codes in task T1.2 (see guideline DG15 in Appendix 2).

Example X2. Using Root Definitions at the strategic level of thinking (or level 1 thinking in the context of multi-level thinking technique) to derive requirements for addressing the core problem coded C2.2 and other problems from task T1.2 (listed in example X1).

Action to undertake (i.e., ultimate goal or desired solution)	**Specific Means** (i.e., implied requirements that must be fulfilled and their codes)	**Specific Challenges** (i.e., code of problems from task T1.2)
A Framework to support the Strategic Management of an integrated Community-level Information Service (SMACIS)	*R1. Need to develop a policy on the use of the integrated community-level information service.*	• C2.2, C1.1, C1.2
	R2. Need to develop a digital platform that delivers the information service.	• C2.2, C2.1, C1.1 to C1.3
	R3. Need to develop a monitoring, evaluation, and sustainability plan for the information service.	• C2.2, C1.3, C2.1

Note: Challenges in column 3 of this example are the same challenges that are referred to in subsequent examples. Thus, the challenges column is not going to be included in subsequent examples to avoid redundancy.

5.2.2 Identify Potential Actions for Addressing each Requirement or Design Alternatives of the Desired Solution Framework (T2.2)

This involves determining possible courses of action or best practices that can be implemented to address each requirement with respect to a specific challenge and available resources. The potential actions for addressing each requirement are fundamentally the 'design alternatives' of the required framework for directing action. Design alternatives are the alternatives or candidate ways or elements that (a) specify how each requirement can be fulfilled to realize the desired state and (b) can be selected and adopted so as to obtain the required solution framework. The generation of potential actions for addressing requirements (or design alternatives of the required solution framework) can be done by adopting or adapting insights from existing potential approaches in the solution space of a project, to ensure that each potential action or design alternative is credible.

Similarly to task T2.1, task T2.2 can be executed using Multi-level thinking and Root Definitions. Adoption of the Multi-level thinking technique in task T2.2 prompts a researcher to determine: **'what'** are the potential or candidate actions that can be undertaken to achieve each of the requirements (or alternative ways of designing the desired solution framework); and **'why'** they are deemed appropriate. In addition, adoption of the Root Definition in task T2.2 focuses on specifying aspects for 3 dimensions of its formula, i.e., **'the action to undertake'** *(which are the specific requirements to fulfill as articulated in task T2.1)*, using **'specific means'** *(which are the potential actions that can address specific requirements or design alternatives of the required solution framework – assertions on 'how' to achieve specific requirements)*, so as to address **'specific challenges'** *(which are articulated in task T1.2)*. This is clarified in the text box of Example X3. Each generated implementation option or Design Alternative is assigned a unique code to *(e.g., DAx as indicated in the text box of Example X3)*. This is done to ensure traceability between the implementation options for requirements or design alternatives of the required solution framework, with respect to issues coded in task T1.2 (see design guideline DG15 in Appendix 2).

Example X3. Using Root Definitions at the tactical level of thinking (which is level 2 thinking in the context of multi-level thinking technique) to derive potential actions for addressing requirements coded Rx from task T2.1 (under example X2).

Action to undertake (i.e., requirements specified in T2.1)	**Specific Means** (i.e. potential actions for realizing requirements or design alternatives of SMACIS as the required solution framework)
R1. Need to develop a policy on the use of the integrated community-level information service.	R1.DA1. Engage the ministry of local government to assign its legal department or to hire a consultant to develop a policy on the effective use of the required community-level information service.
	R1.DA2. Undertake research that will produce guidelines or key components that should be covered by the required policy.
	R1.DA3. Undertake research that will produce the required policy.
R2. Need to develop a digital platform that delivers an integrated community-level information service.	R2.DA1. Engage the ministry of local government to assign its ICT department or to hire a consultant or to engage an software service provider to develop the community-level information service.
	R2.DA2. Conduct research that will produce a solution architecture of the digital platform for the community-level information service, so as to guide implementers to deliver the digital platform.
	R2.DA3. Conduct research that will produce guidelines for building the solution architecture for the digital platform, for implementing it, and for deploying it.
	R2.DA4. Conduct research that will produce the digital platform for the community-level information service.

Note: Potential actions for realizing each requirement are fundamentally design alternatives for SMACIS (which is the desired solution framework, as specified under example X2). Also, column 3 is removed from this example to avoid redundancy because it contains the same content of issues as given in example X1.

5.2.3 Devise Implementation Options for each Potential Action or Design Alternative and Elaborate them using Purposeful Activity Models (T2.3)

This involves elaborating on each potential action or design alternative specified in task T2.2 by suggesting implementation options for operationalizing it. The implementation options are essentially measures and mechanisms that need to be established with particular quality attributes in order to achieve each potential action or design alternative (with respect to the requirements in task T2.1, challenges faced in task T1.2, and available resources). The identification and elaboration of implementation measures and mechanisms for specific potential actions or design alternatives can be inspired by an existing approach in the solution space and/or problem space of a project. In addition, task T2.3 can be executed using 4 SSM techniques, i.e., Multi-level thinking, Root Definitions, CATWOE analysis, and Purposeful Activity Models (defined in Section 4). The use of these techniques is explained below. The involved aspects are clarified in the text box of Example X4.

First, the adoption of the Multi-level thinking technique in task T2.3 prompts a researcher to determine: **'how'** the potential actions can be realized or their implementation measures (i.e., exploring the 'how' details of 'what' were specified in task T2.2) and **'why'** the implementation measures are deemed as candidates. Accordingly, each potential action or design alternative is first elaborated by identifying candidate high-level transformation processes (which can be perceived as *'implementation measures'*) associated with realizing each potential action or design alternative.

Example X4. Using Root Definitions at the operational level of thinking (which is level 3 thinking in the context of multi-level thinking technique) to derive implementation measures and mechanisms for each potential action or design alternative coded DAx in task T2.2 (under example X3).

Potential Action for realizing a requirement (**or Design Alternative** for desired solution framework) from T2.2	**Implementation options (or measures and mechanisms)** for each potential action or design alternative	Merits of each potential action or design alternative (with respect to its implementation measures & mechanisms)	Demerits of each potential action or design alternative (with respect to its implementation measures & mechanisms)
R2.DA3. Conduct research that will produce guidelines for building the solution architecture for the digital platform, implementing it, & deploying it.	*Measure R2.DA3.M1: Adopt an enterprise architecture framework to guide the development of guidelines for building the solution architecture of the platform.* • *Mechanism R2.DA3.M1.W1: Use TOGAF standard.* • *Mechanism R2.DA3.M1.W2: Use TOGAF with another content framework.*	• *End product can be achieved within a period of 6-12 months in a research project; &* • *End product can be adopted by other ministries to address related needs.*	• *End product is not the required 'last-mile' solution since it will only prescribe the procedure of designing the solution architecture for the integrated community information service (but not the actual digital platform).*
R2.DA4. Conduct research that will produce the digital platform for the community-level information service.	Measure R2.DA4.M1: Build the platform by adopting the Incremental Model of software development. • Mechanism R2.DA4.M1.W1: Execute modelling tasks using Object Oriented Analysis and Design techniques.	• End product is the required 'last-mile' solution of a functional SMACIS that can address the problem at hand.	• High risk of failure without a good plan & design of the platform. • End product will not have a framework that guides authorities on all supporting structures that they should establish before rolling out the platform.

Note: Each potential action or design alternative in column 1 can have more than one implementation measure in column 2, and each implementation measure can have several operationalization mechanisms. This is because task T2.3 is about elaborating each potential action, or design alternative in order to obtain sufficient information that can inform decision-making on its appropriateness. Implementation measures are coded as Mx, while their corresponding implementation mechanisms are coded as Wx.

Second, the adoption of the Root Definition in task T2.3 focuses on identifying implementation mechanisms of specific measures by specifying aspects for 3 dimensions of its formula, i.e., **'the action to undertake'** *(which are the implementation measures for specific potential actions or design alternatives articulated in task T2.2)*, using **'specific means'** *(which are the implementation mechanisms for the specific measures)*, so as to address **'specific challenges'** *(which are articulated in task T1.2)*. Consequently, each identified implementation measure (or high level transformation processes) is further elaborated by identifying specific structures or tools required for its operationalization (which can be perceived as *'implementation mechanisms'*).

Third, the adoption of the CATWOE analysis in task T2.3 focuses on explaining each derived root definition by specifying 6 parameters, i.e., *i) Customers* or stakeholders to be affected by specific measures and their corresponding mechanisms, *ii) Actors* to be engaged in operationalizing specific measures and their corresponding mechanisms, *iii) Transformation processes* that must be executed (at operational level) to achieve outcomes of specific measures and their corresponding mechanisms, *iv) World views* or perspectives on the effectiveness of specific measures and their corresponding mechanisms, *v) Owners* or sponsors who will provide resources required to realize specific measures and their corresponding mechanism, and *vi) External constraints* that may affect the realization of specific measures and their corresponding mechanisms. Thus, the identified implementation measures and corresponding mechanisms are further elaborated by identifying their implied customers or stakeholders, actors, operational level transformation processes, world views or literature perspectives on their effectiveness and gaps, owners or providers of required resources, and external constraints with respect to their operationalization. Facts associated with the 6 parameters in the CATWOE analysis can be obtained from existing literature in the solution space and/or problem space of a project, or they can be obtained directly from stakeholders.

Fourth, the adoption of the Purposeful Activity Models in task T2.3 focuses on prompting a researcher to visualize the assembling or consolidation of transformation processes and monitoring and control aspects associated with specific implementation measures and their corresponding mechanisms. The visualization helps to clarify further and create a shared understanding among stakeholders on the generated implementation measures and mechanisms for each potential action or design alternative. Visualizing the orchestration of implementation measures and mechanisms helps to holistically assess them and explore their aggregation and disaggregation so that they can be rationally analyzed. In some contexts, this visualization can reveal that what was first deemed an alternative option becomes an operationalization measure or mechanism of another alternative.

5.2.4 Compare and Assess Views of Activity Models for Potential Actions or Design Alternatives and Select the Most Appropriate (T2.4)

This involves assessing the feasibility and appropriateness of the purposeful activity models devised in task T2.3 to show implementation measures and mechanisms for each potential action or each design alternative. The assessment of the appropriateness of views for purposeful activity models for potential actions and design alternatives is done with respect to information obtained from these five sources, i.e., a) the requirements (in task T2.1), b) challenges (in task T1.2), c) available resources, d) existing facts about particular implementation measures and mechanisms (that are specified during the CATWOE analysis in task T2.3), and e) literature on/from the problem space and/or solution space of a project. Findings from these sources are used to describe the merits and demerits of each potential action or design alternative. This is clarified in the text box of Example X4 (see columns 3 and 4). The merits and demerits of each potential action or design alternative are indicators of their appropriateness and feasibility in delivering the desired state. Thus, after holistic assessment, the appropriate actions and design alternatives along with their appropriate implementation options (or measures and mechanisms) are selected for further consideration in the ‘build’ activity of the desired solution framework.

5.3 Synthesize Design Decisions to Constitute Framework (T3)

Figure 5 shows that task T3 comprises sub tasks T3.1 to T3.5, which are elaborated in Sections 5.3.1 to 5.3.5. Appendix 2 shows design guidelines that are operationalized during the execution of tasks T3.1 to T3.5.

5.3.1 Specify Appropriate Design Alternatives as Design Decisions or Design Choices and Assign Traceability Codes (T3.1)

This involves adopting the appropriate implementation measures and mechanisms (or design alternatives) from task T2.4 and specifying them as the Design Decisions or Design Choices taken by a researcher to constitute the desired solution framework. Depending on the problem or solution context of a project, these Design Decisions or Choices can be referred to as *Elements* (as used in [36]), or *Variables* (as used in [37]), or *Supporting Structures* (as used in [38], [39]), or *Components or adopted measures and mechanisms* that constitute the desired framework. In task T3.1, each specified Design Decision or Choice is assigned a code (e.g., **DDx** or **DCx**). This is done to ensure traceability between the adopted design decisions for addressing specific requirements in task T2.1 and to resolve specific issues in task T1.2 (see guideline DG15 in Appendix 2). These aspects are clarified in the text box of Example X5.

Example X5. Specifying Design Decisions from design alternatives and corresponding implementation options (i.e., measures and mechanisms) that are deemed appropriate in task T2.4 with respect to requirements specified in task T2.1.

Requirements from T2.1 (in Example X3)	***Appropriate Design Alternatives and their appropriate implementation measures & mechanisms from T2.4 (in Example X4)***	***Design Decisions taken to constitute the framework***
R2	*R2.DA3, and its measures and mechanisms that are deemed appropriate are:* • *Measure R2.DA3.M1* • *Mechanism R2.DA3.M1.W1.*	*DD2. Adapt the TOGAF standard to derive guidelines for designing a solution architecture for the integrated community information service.*

Note: For task T3.2 and T3.3, Design Decision DD2 can be chosen as the pivotal design decision that can inform the synthesizing or orchestration of other design decisions taken (to address other requirements in task T2.1). In addition, TOGAF as an existing approach mentioned in DD2 can be chosen as the pivot that will support the orchestration of other design decisions taken (to address other requirements in task T2.1) and their implementation measures and mechanisms that are deemed appropriate.

5.3.2 Determine Design Decision and/or Existing Artifact(s) that can be Used as a Pivot for Enabling Orchestration of all Design Decisions (T3.2)

This involves rationally scanning the catalogue of the Design Decisions (specified in task T3.1) to identify at least one that can be treated as an axis or pivot of all other Design Decisions. The selection of the pivotal design decision(s) could be inspired or supported by an already existing artifact or perspectives from the solution space of a project. Thus, the scanning enables a researcher to identify an existing artifact (or theory, method, framework, model, technique, standard, approach) in academia or in industry or from the solution space and/or problem space for a project, that can be adapted and used as the major axis for deriving a synthesis or orchestration of all Design Decisions. These aspects are clarified in the text box of example X5.

5.3.3 Use Selected Pivotal Design Decision or Existing Artifact to Synthesize all Design Decisions into a Framework (T3.3)

This involves using the pivotal Design Decision or existing artifact (identified in task T3.2) to logically derive a synthesis or orchestration of all other design decisions, so as to obtain the first draft version of the desired framework. The logical reasoning approach used to orchestrate or synthesize all design decisions can be taken by leveraging the thinking pattern of Root Definitions and Multi-level thinking that is elaborated in Section 4.2 (under tasks T2.1 to T2.3). Examples of derived orchestrations of design decisions in studies that used TBUFI can be found in Section 6.

5.3.4 Specify Assumptions and Constraints that Shape the Synthesis of all Design Decisions into a Framework (T3.4)

The orchestration of design decisions in task T3.3 involves considering several constraints associated with adopting particular approaches or techniques to address specific needs that are associated with each design decision. Thus, specifying any assumptions made or constraints adhered to during the orchestration helps to make the orchestration process transparent and therefore repeatable. Since the design process is highly a 'creativity-centric' initiative, a rigid design guide can inhibit the generation of an agile and flexible product or output [12]. This explains why some tasks in TBUFI (particularly T3.4, T3.3, T2.2, T5.1, and T5.2) appear in an 'open-ended' mode that allows a researcher to explore a range of possibilities and justify them. For instance, in task T3.3, a researcher has the liberty to orchestrate design decisions in all possible ways, but is prompted in task T3.4 to articulate the assumptions and constraints that influence the orchestration. This is done to provide some form of transparency and repeatability of the orchestration task.

5.4.5 Identify Gaps in the Derived Synthesis of Design Decisions and Devise Rationality and Granularity Aspects that Can Fix Them (T3.5)

Some solution designs can be too abstract, too detailed, or too complex for its target users to understand it or use it. Thus, task T3.5 involves analytically assessing the synthesized design decisions with respect to requirements (specified in tasks T2.1) and challenges (specified in tasks T1.2), with the intention of identifying and resolving two types of design related gaps. *First,* there is a need to identify and resolve gaps associated with the logical mapping or alignment that involves ensuring that each Design Decision addresses at least one particular requirement(s) and challenge(s). This is done to ensure a) Traceability between Design Decisions, requirements and challenges and b) Completeness of the solution by exploring the extent to which particular requirements and challenges are addressed by the designed framework or method. *Second*, there is a need to identify and resolve visualization gaps and granularity related gaps that can affect the logical and mutual understanding of the composition and structural layout of the designed framework, its usability, and its continuous improvement. When these two types of gaps are identified, additional design decisions are taken to address them.

These additional design decisions can be perceived as *'rationality and granularity aspects or elements'* because they serve the purpose of improving the understandability, usability, and continuous improvement of the designed framework. Consequently, these rationality and granularity aspects or elements are consolidated into the initial synthesis or design of the framework to improve it, in preparation for stakeholder deliberations on its feasibility.

5.4 Deliberate and Assess Feasibility of the Constituted Framework (T4)

Figure 4 shows that adoption of the multi-level thinking in task T2 triggers tasks T2, T3, T4 and T5. The preceding sections explain the focus questions that this technique implies in tasks T2 and T3. In task T4, the questions that the multi-level thinking technique implies are whether the 'what' and 'how' described in a particular constituted framework are feasible and make sense to stakeholders. In task T5, the question is what amendments have to be incorporated and how can they be incorporated without negatively affecting the entire synthesis of the framework. Figure 5 shows that task T4 comprises sub tasks T4.1 to T4.4, elaborated in Sections 5.4.1 to 5.4.4. Appendix 2 shows design guidelines that are operationalized during tasks T4.1 to T4.4.

5.4.1 Deliberate Design Decisions and Corresponding Rationality and Granularity Aspects that Constitute the Framework (T4.1)

This involves engaging specific categories of stakeholders in the problem situation to debate the feasibility and appropriateness of design decisions that shape the draft design of the framework. The draft design of the framework is meant to trigger insightful discussions on the socio-cultural, political, technical, and economic feasibility of the desired changes to address issues in the problem environment or needs associated with the desired context. Other quality attributes associated with the understandability, usability, and continuous improvement of the framework are also deliberated. Task T4.1 interacts with the evaluation activity of a DSR project (as indicated by the gray-shaded box in the left side of Figure 2). Thus, the details of how the debate among stakeholders is conducted are beyond the scope of TBUFI, which focuses on elaborating the design activity of a DSR project.

5.4.2 Perform Tradeoff Analysis of Stakeholder Viewpoints with respect to the Constituted Framework (T4.2)

This involves assessing stakeholder viewpoints on the draft design of the desired framework to determine ways of resolving or accommodating two types of conflicting views from task T4.1.

First, conflicting views can occur within and across stakeholder categories on specific aspects or attributes of the designed framework. Second, conflicting interests can occur between particular stakeholder viewpoints on specific design decisions or components that make up the solution framework.

5.4.3 Identify Components of the Framework in which Prototyping can be Applied to Further Assess Framework Feasibility and Suitability (T4.3)

Frameworks that guide digital interventions often prescribe digital solutions required to address the problem or need in a particular context and the required supporting structures (or measures and mechanisms) for developing and operationalizing those solutions. Therefore, a comprehensive evaluation or assessment of the framework's suitability often requires a) implementing one of its core digital solutions, b) evaluating the prototype of at least one digital solution to assess feasibility, and c) using findings from the feasibility assessment to refine the framework. This is done to ensure that there is adequate evidence to confirm feasibility of the prescribed digital solutions and of the prescribed supporting structures in the framework to ensure that it fulfills its intended purpose. Task T4.3 can be executed in parallel with task T4.1 (or after task T4.1), depending on context and the availability of resources.

5.4.4 Derive and Assess Additional Design Alternatives to Address Feasible Viewpoints (T4.4)

This involves assessing any additional implementation options (measures and mechanisms) of addressing feasible viewpoints, selecting appropriate measures and mechanisms, and adopting them as additional design decisions to enrich the design of the framework. Since execution of task T4.4 involves the same logical thinking pattern that is used in executing tasks T2.2 to T2.4, the techniques used to execute the latter tasks can be used to execute task T4.4.

5.5 Update Framework and its Design Process (T5)

Figure 5 shows that task T5 comprises sub tasks T5.1 to T5.4, which are elaborated below. Appendix 2 shows design guidelines that are operationalized during the execution of tasks T5.1 to T5.4.

T5.1. Revise synthesis of design decisions and rationality and granularity aspects to fix required changes: This entails amending the design of the framework to reflect changes implied by the additional design decisions and corresponding rationality and granularity aspects or elements from task T4.4. The execution of this task (T5.1) involves the same logical thinking pattern that is used in executing tasks T3.1 to T3.5 in Section 4.3. Thus, the techniques used to execute tasks T3.1 to T3.5 can be used to execute task T4.4.

T5.2. Specify situational factors associated with adopting and implementing the framework with respect to context: This involves specifying situational factors associated with the implementation and use of the designed framework based on stakeholder deliberations and researcher's experiences and observations from tasks T4.1 and T4.4. In addition, this task involves critically assessing the situational aspects associated with using specific approaches (that have been adopted from the solution space of a DSR project) in a particular problem context.

T5.3. Continuously improve the framework until saturation by repeating tasks T1.1 to T5.2: This involves repeating the execution of tasks T4.1 to T5.2 with respect to available resources until a state is reached. Saturation occurs when any desired changes in the method or framework no longer have significant implications on the quality of the design (e.g., composition, structural layout, understandability, and usability) of a method or framework.

T5.4. Specify issues encountered in executing tasks T1.1 to T5.3 and measures taken to address them: As a researcher executes tasks T1.1 to T5.4 in TBUFI, he or she is likely to face several difficulties associated with the usability and understandability traits of TBUFI. Thus, this task prompts the researcher to specify issues faced in executing each task. These issues inform the further refinement of TBUFI.

6 Evaluation of TBUFI

From 2011 to 2023, ten research studies were undertaken that used *"the knowledge behind TBUFI"* although it was not yet explicitly or formally specified as indicated in the preceding sections. The ten studies used the design knowledge or logical thinking pattern in TBUFI to guide the creation of a framework or method that was regarded as the desired solution for addressing the unstructured problem that was dealt with in a particular study. Thus, the ten studies are treated herein as the first 10 evaluation iterations that underpin the relevance, applicability, and feasibility of the 'design knowledge' coined as TBUFI in this article. The ten studies do not describe the structural composition and theoretical justification of TBUFI. *Instead*, they discretely refer to it and as *the procedure that was followed when building the framework of interest in a specific context*, and do not discuss the strengths and weaknesses of the procedure. This is due to the fact that the focus of the ten studies was entirely on devising a framework or method for a specific function (as the ultimate output of each study), but not the procedure for creating it. Thus, this article provides a coherent description of TBUFI (in Sections 4 and 5), discusses its theoretical justification (in Sections 1 and 3), specifies the evaluation iterations that TBUFI has undergone (in Sections 6.1 and 6.2), and highlights key lessons learned from its evaluation (in Section 6.3).

6.1 List of Evaluation Iterations of TBUFI

To date, the design knowledge in TBUFI has been evaluated in 11 iterations coded as E1 to E11, to enable their traceability in the discussion provided below and in subsequent sections.

a) **Iteration E1** involved using TBUFI to guide the 'build' activity of a *process that supports enterprise architects to engage stakeholders in executing collaboration dependent tasks during the creation of enterprise architectures*. The study followed the DSR approach, and the resultant artifact was a process or method that is coined as CEADA and discussed in Nakakawa et al. [40]. This is considered as the iteration that yielded the first version of the design knowledge behind TBUFI. Lessons learned from this iteration informed its continuous refinement, which yielded the transitional versions that were used in iterations E2 to E10.
b) **Iteration E2** involved using TBUFI to guide the 'build' activity of a *framework for guiding the implementation of e-government enterprise architectures in a developing country*. The study followed the DSR approach, and the resultant artifact was a framework that is coined as EGEAF and discussed in [41].
c) **Iteration E3** involved using TBUFI to guide the 'build' activity of a *framework for leveraging service-oriented architectures during the development of interoperable e-health systems in a low-income country*. The study followed the DSR approach, and the resultant artifact was a framework that is coined as SOFIEH and discussed in [36].
d) **Iteration E4** involved using TBUFI to guide the 'build' activity of a *framework that specifies measures that regulatory authorities and developers of digital solutions can undertake to enhance the adoption of electronic payment in a developing economy*. The study followed the Participatory Action Research approach, and the resultant artifact was a framework that is coined as FAEP and discussed in [37].
e) **Iteration E5** involved using TBUFI to guide the 'build' activity of a *framework for guiding regulatory authorities and solution developers of digital solutions to implement ICT solutions that enable the sharing of agricultural knowledge among smallholder farmers*. The study followed the DSR approach, and the resultant artifact was a framework that is coined as ICT-AKS and discussed in [42].

f) **Iteration E6** involved using TBUFI to guide the 'build' activity of a *framework for guiding the continuous evaluation and improvement of the performance of Electronic Health Information Systems*. The study followed the Participatory Action Research approach, and the resultant artifact was a framework that is coined as PEHIS and discussed in [39].
g) **Iteration E7** involved using TBUFI to guide the 'build' activity of a *framework for regulatory authorities and developers of digital health interventions to develop and implement clinical decision support solutions that enable knowledge sharing (among health workers) in the management of acute child malnutrition in Ugand*a. The study followed the DSR approach, and the resultant artifact was a framework that is coined as CLISAM and discussed in [43].
h) **Iteration E8** involved using TBUFI to guide the 'build' activity of a *framework that specifies measures that can be undertaken to leverage Public Participatory Geographic Information Systems in the management of municipal solid waste*. The study followed the DSR approach, and the resultant artifact was a framework that is coined as GPEP and discussed in [38].
i) **Iteration E9** involved using TBUFI to guide the 'build' activity of a *framework for managing usability challenges in the design of integrated information systems for the Justice Law and Order Sector in Uganda*. The study followed the DSR approach, and the resultant artifact was a framework that is coined as AdUPRO and discussed in [44].
j) **Iteration E10** involved using TBUFI to guide the 'build' activity of a *framework that specifies measures that small and medium enterprises in Uganda can undertake to leverage social media technologies in enhancing customer relationship management*. The study followed the DSR approach, and the resultant artifact was a framework that is discussed in [45].
k) **Iteration E11** involved conducting a group walkthrough meeting in which purposively selected information systems specialists and researchers deliberated the strengths and weaknesses of the structural composition and layout of TBUFI. After the ten research studies recognized above as evaluation iterations E1 to E10, there was a need to a) further evaluate the design knowledge behind TBUFI in a setting where researchers who had used TBUFI (and its potential users) could deliberate and evaluate its strengths and weaknesses and b) use findings to derive the current version of TBUFI (that is presented in Section 4). This evaluation was done by conducting a group walkthrough session, which is referred to herein as evaluation iteration E11. Thus, Section 6.2 discusses the setup of iteration E11, and Section 6.3 discusses the key findings.

6.2 Setup of Evaluation Iteration E11

The aim of the group walkthrough meeting in iteration E11 was to provide an avenue where information systems specialists who had used TBUFI in their research studies (and potential users) would do the following three tasks: (i) Reflect about their experiences in using the framework; (ii) Deliberate its strengths and weaknesses so as to create a shared understanding of its composition and help to address any possible misinterpretations; and (iii) Individually provide feedback on its feasibility, understandability, usability, and functionality. An overview on how iteration E11 was set up is provided below.

A. *Type and Number of Participants or Evaluators.* The group walkthrough involved 12 participants, who were purposively selected to represent information systems specialists and researchers in three categories, i.e., a) those who had used TBUFI in research studies described in evaluation iterations E2 to E10; b) those who were currently using TBUFI, or were planning to use in their research studies; and c) those who had examined research dissertations that had used TBUFI, and those who are examiners of graduate research dissertations.
B. *Agenda of the Group Walkthrough Meeting and Follow-up Sessions.* Materials about TBUFI that were to be discussed in the meeting were shared by participants one week before the scheduled meeting date. The group walkthrough meeting covered five major agenda items.
   - *First,* the facilitator delivered a 30-minute presentation on the layout and composition of TBUFI.

- *Second,* participants were encouraged to share their qualitative views on four quality attributes of TBUFI: feasibility, understandability, usability, and functionality.
- *Third,* the facilitator prompted participants to react or interactively deliberate views shared by other participants on the above 4 quality attributes.
- *Fourth,* the facilitator urged participants to review materials that had been shared about TBUFI (for an additional week after the group walkthrough meeting) and independently fill the evaluation checklist for TBUFI.
- *Fifth,* after the group walkthrough meeting some participants preferred to have additional follow-up sessions with any of the facilitators, so as to get clarifications on some aspects in TBUFI.

C. *Duration.* The group walkthrough meeting lasted for two hours.

D. *Meeting Mode.* The group walkthrough meeting was conducted in virtual mode, supported by a groupware platform, and views of participants were transcribed and summarized as presented in Section 6.3.

E. *Inputs for the Group Walkthrough.* Five inputs were shared before the group walkthrough meeting that included the following: (1) A table that showed the requirements that TBUFI must address (see Appendix 1); (2) A diagram that shows the structural layout and composition of TBUFI (i.e., the earlier version of Figure 3); (3) A table that shows the decomposition of tasks that constitute TBUFI and the design guidelines addressed by each task (i.e., the earlier version of Appendix 2); (4) Illustrative examples of expected outputs of each subtask of TBUFI; and (5) the evaluation checklist for TBUFI.

F. *Evaluation Questionnaire.* Questions in the evaluation questionnaire were closed and open-ended. The closed questions were in the form of statements or phrases that prompted a respondent to indicate the extent to which he/she concurs with the implied argument therein, using a 5-point Likert scale ranging from 1 (for the lowest level of agreement) to 5 (for the highest level of agreement). The approach of using a 5-point Likert scale to determine the extent to which evaluators agree with the outcomes of collaborative activity is adopted from Briggs et al [46]. The open-ended questions prompted an evaluator to comment on missing aspects in TBUFI, its weaknesses, and how its design can be improved. Due to the scope of this article and space limitations, the evaluation questionnaire cannot be included herein.

G. *Outputs of the Meeting.* Section 6.3 presents the qualitative and quantitative user feedback on the feasibility, understandability, usability, and functionality of TBUFI.

6.3 Discussion of Findings from TBUFI Evaluation Iterations

Quantitative feedback on TBUFI is presented in Table 1, and the qualitative feedback is discussed thereafter. Quantitative responses were elicited using a 5-point Likert scale on four major attributes of TBUFI – its applicability, scope of functionality, usability, and traceability (as defined in column 1 of Table 1). To achieve this, each attribute was disaggregated into evaluation criteria that were posed as statements on features of TBUFI, and evaluators were prompted to indicate the extent to which they agreed with the statements. The 12 evaluators individually scored TBUFI on each evaluation criterion under each attribute. Column 2 shows the average score of TBUFI under each of these attributes, and column 3 shows the standard deviation in the scores. Deriving the standard deviation of scores helps to determine the level of consensus among participants on the average score of an aspect that is being evaluated [46].

During the group walkthrough discussion, evaluators appreciated the relevance of TBUFI as a mechanism that details the artifact construction process in information systems research and shared their views on how TBUFI could be improved. Evaluators gave their feedback based on observations that they made when using TBUFI in iterations E1 to E10 and from deliberations on TBUFI in iteration E11. Feedback was captured using an evaluation questionnaire (described in item F under Section 6.2). Qualitative feedback was analyzed using content analysis, and the following themes of required considerations or improvements in TBUFI were derived.

Table 1. Quantitative Feedback from Iteration E11

Category of attribute used to evaluate TBUFI	**Average Score**	**Standard deviation**
a) Applicability – the extent to which TBUFI is relevant in supporting the design process of a framework or method. • *Four (4) evaluation criteria were used to assess TBUFI under this attribute.*	4.24	0.97
b) Scope in functionality – the extent to which TBUFI addresses design guidelines DG1 to DG13 and other design needs. • *Thirteen (13) evaluation criteria were used to assess TBUFI under this attribute (criteria were derived from requirements implied by DG1 to DG13 as listed in Appendix 1).*	4.44	0.08
c) Usability – the extent to which TBUFI can be easily or independently used or applied by a user or researcher to effectively complete a mission. • *Eleven (11) evaluation criteria were used to assess TBUFI under this attribute (criteria were derived from requirements implied by DG14 as listed in Appendix 1).*	4.38	0.69
d) Traceability – the extent to which it is possible to trace the history, application, or location of elements that constitute TBUFI. • *Six (6) evaluation criteria were used to assess TBUFI under this attribute (criteria were derived from requirements implied by DG15 as listed in Appendix 1).*	4.38	0.70

In evaluation iterations E1 and E3, it was observed that in some contexts, industry-driven approaches can be used as a pivotal approach during the execution of task T3. When using TBUFI, some researchers noted that tasks T3.2 and T3.3 were not explicit on the type of pivotal artifact(s) that can be selected to consolidate selected design decisions for constituting a framework or method. Yet literature in some research contexts may not reveal candidate approaches from academia, but may reveal candidate approaches that originate from, or were developed by, government agencies and consultancy firms. This has been addressed as explained in Section 5.3.

In evaluation iterations E4, E5, and E8, it was observed that some tasks of TBUFI can be executed and others may not be executed due to the limited resources for the research and the nature of the research or desired solution. The group walkthrough discussions also revealed that TBUFI has several tasks and subtasks, some of which may not be applicable in particular research settings. Yet, it is not clear which tasks and subtasks are optional and mandatory for execution by a user or researcher. This implied the need to ensure that TBUFI is flexible with respect to variations in research contexts, by articulating mandatory and optional tasks and subtasks in TBUFI. This has been addressed by using gray-shaded color codes in Figure 5 (see Section 4.3).

From iterations E1 to E10, it is vital to prompt a user of TBUFI to articulate lessons learned at each stage so as to provide feedback that can be used to improve the structural composition and layout of TBUFI continuously. This was addressed in the current version of TBUFI, by incorporating a task that prompts a user to provide feedback after executing the subtasks of TBUFI. The added task is T5.4 (see Figure 5 in Section 4.3 and Appendix 2).

Examples were given for only a few tasks and the examples were from different case studies, which implies the need for a crosscutting example or scenario that is incrementally elaborated for each task so as to facilitate a user to understand and use TBUFI correctly. Evaluators noted that providing illustrative examples of outputs expected from particular subtasks is very important in helping a researcher or user of TBUFI to understand the intentions of each subtask. However, evaluators were concerned that the illustrative examples that were provided in iteration E11 were from different scenarios or contextual settings. Accordingly, evaluators recommended providing a crosscutting scenario, which should be the source of examples for all subtasks of TBUFI. This is envisioned to help to reduce the training that one needs to undergo before using TBUFI. This is because discussions in the group walkthrough revealed that a researcher can only properly understand and effectively use TBUFI, if he/she is first trained on how to execute its tasks and subtasks.

Iterative loops in TBUFI subtasks would have been better demonstrated with a diagram instead of the tabular layout that was provided (see Appendix 2). If a tabular layout is used to represent linkages and iterative loops between subtasks, they end up appearing like repetitions. This has been addressed by using Figure 5 to communicate the sub tasks of TBUFI, and the iterative loops that are likely to be encountered in their execution are described in the detailed narrative of the subtasks (as presented in Sections 5.1 to 5.5).

Prior to the group walkthrough discussions, evaluators were only provided with five types of materials about TBUFI (which are specified in item E under Section 6.2). The detailed description of TBUFI, as presented in Sections 5.1 to 5.5, was not shared with the evaluators in order to avoid too much documentation that would overwhelm them. However, this was not the right decision because feedback from the group walkthrough revealed the following issues, which could have been avoided if the detailed description of TBUFI was also shared with evaluators.

a) Evaluators noted that the TBUFI design had to be accompanied by detailed documentation that explains how to execute each subtask and task. This was suggested because evaluators noted that the materials about the design of TBUFI (that were shared with them prior to the group walkthrough session, as specified in item E under Section 6.2) did not provide details on the following aspects: (i) how to do requirements analysis, (ii) how to implement the framework or derive prototypes of digital solutions from the framework; and (iii) how to test whether the framework and its prototypes coherently inform each other and their continuous improvement.
b) Some evaluators noted that the lack of a detailed description of how subtasks in TBUFI should be executed can make a researcher or user assume that TBUFI is only usable in research studies that have adopted only the DSR approach and not other research approaches. This is because the terminology used to define tasks and subtasks of TBUFI is inclined on DSR vocabulary. To address this misinterpretation, Section 4 indicates how TBUFI can be used as a supplementary technique for creating framework or method when using other research methods besides DSR. For instance, in evaluation iterations E4 and E6, TBUFI was used in research studies that used the Participatory Action research method.
c) Some evaluators were concerned that TBUFI does not (i) specify the hardware and/or software that a researcher or user can use during the execution of its tasks; (ii) advise on the type of stakeholders to include in deliberations of task T4 (i.e., are they only researchers or practitioners in information systems or digital transformations, or institutional stakeholders, or both categories); and (iii) specify techniques that can support the execution of specific subtasks in TBUFI.
d) Some evaluators noted that it is not clear whether a researcher should engage stakeholders of a particular context in only task T4, or also in tasks T1 to T3 and T5. This implied the need indicate specific tasks in which a researcher needs to engage stakeholders of a particular context. To address this, Sections 5.1 to 5.5 indicate that stakeholder views should be incorporated in all tasks of TBUFI. However, the resources for a given study, may not allow this to happen. Thus, a user or researcher needs to choose particular tasks (besides T4), in which they can engage stakeholders or elicit stakeholder views in a cost effective way.

In addition, prior to the group walkthrough meeting, the acronym of TBUFI read in full as "Technique for Building Frameworks for guiding unstructured Interventions". During the walkthrough discussions, it was noted that the wording used in the full name seems to imply that an intervention is the one that is "unstructured", yet the intervention is devised to address an unstructured problem. The discussions revealed the need to revisit the wording, so that this anomaly is rectified. To cater for the wording anomaly, the full name of TBUFI has been refined to read: "Technique for Building Frameworks for guiding Interventions against unstructured problems".

Furthermore, prior to the group walkthrough meeting, TBUFI had tasks T1.5, T2.5, T3.6, T4.5, and T5.5 which were phrased as "document difficulties encountered in executing subtasks of a specific task and how they have been overcome". During the group walkthrough discussions, this seemed to imply that other subtasks in TBUFI do not require documentation of their outputs, yet they do. This implied the need to revisit the naming of these tasks to avoid misinterpretation by a researcher and to consolidate them all into task T5.4 as currently depicted in Figure 5 and Appendix 2.

7 Conclusion and Future Work

Although the design cycle of a DSR project comprises the 'build' activity and evaluation activity, this paper concentrates on only the 'build' activity. March and Smith [1] and Hevner et al. [2] broadly classify DSR artifacts into four types, i.e., constructs, models, methods, and frameworks, or instantiations that address particular types of problems or needs. However, this research renames the category of 'methods' to 'methods and frameworks'. The renaming is motivated by the need to emphasize that a 'method' is high level description of a best practice for solving a specific problem, and a 'framework' is a detailed level description of a best practice for solving a specific problem. Accordingly, this research focuses on elaborating the 'build' activity of artifacts in the category of 'methods and frameworks' (in general), and the sub category of 'frameworks' (in particular) that direct interventions against unstructured (digital) interventions.

Several general guidelines exist that inform the design process of a DSR artifact. However, an explicit procedure of how the existing guidelines can be coherently operationalized during the building or creation of an artifact is still lacking. Besides, a logical procedure or thinking pattern that can be followed during the building, creation, or construction of an artifact hardly exists, and the quality of the artifact ultimately depends on the skills and competences of the researcher. Therefore, this article presents research that addresses these two gaps by leveraging SSM techniques to enrich the design cycle of DSR by elaborating the 'build' activity of an artifact, which falls in the category of 'frameworks that direct interventions against unstructured community or organizational problems'. The research yielded TBUFI as a technique of building frameworks for guiding interventions against unstructured problems. The evaluation of TBUFI in 11 research studies from 2011 to 2023 has revealed that it adequately elaborates the 'build' activity of DSR in a way that coherently subscribes to the existing general design guidelines. Evaluation findings from the 11th iteration show that TBUFI scored satisfactorily on quality criteria that are associated with four quality attributes, i.e., usability, feasibility, traceability, and functionality.

Regarding future work, the advancement of TBUFI will focus on three aspects. *First,* efforts are planned to extend TBUFI so as to ensure that it provides a detailed procedure for building (a) other sub-categories or instances of DSR artifacts in the 'methods and frameworks' category since TBUFI is currently focusing on only frameworks and (b) other types of DSR artifacts in the categories of models, constructs, and instantiations. *Second,* efforts are planned to ensure that TBUFI is further used and evaluated in additional studies that address organizational or community problems in a setting where none of the originators of TBUFI will be actively involved in a research study. This is because, in 10 out of the 11 evaluation iterations of TBUFI, the originators of TBUFI were actively involved in the execution of its tasks. They were involved as either the primary researchers or supervisors of any of the research studies in those evaluation iterations. Evaluating TBUFI in a context where its originators play a passive or observational role will provide further insights into its performance regarding understandability, usability, feasibility, traceability, and functionality. *Third,* efforts are planned to ensure that TBUFI is extended by enriching the build activity of DSR through exploring how Situational Method Engineering can be leveraged to clarify aspects in TBUFI that still need to be elaborated. This is also envisioned to further improve its performance under attributes of understandability or usability, functionality, and traceability.

Acknowledgment

We appreciate everyone who participated in the evaluation iterations of this research. We also appreciate reviewers of the earlier versions of this article.

References

[1] S. T. March and G. Smith, “Design and Natural Science Research on Information Technology,” *Decision Support Systems*, vol. 15, no. 4, pp. 251–266, 1995. Available: https://doi.org/10.1016/0167-9236(94)00041-2

[2] A. R. Hevner, S. T. March, T. Park, and S. Ram, “Design Science in Information Systems Research,” *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004. Available: https://doi.org/10.2307/25148625

[3] P. Offermann, S. Blom, M. Schönherr, and U. Bub, “Artifact types in information systems design science – a literature review,” in *Global Perspectives on Design Science Research: Proceedings of the 5th International Conference DESRIST 2010*, Springer, pp. 77–92, 2010. Available: https://doi.org/10.1007/978-3-642-13335-0_6

[4] T. Bucher and R. Winter, “Dissemination and importance of the “method” artifact in the context of design research for information systems,” in *Proceedings of the Third International Conference on Design Science Research in Information Systems and Technology (DESRIST 2008)*, pp. 39–59, 2008. Available: https://www.alexandria.unisg.ch/handle/20.500.14171/78385

[5] R. Winter, “Design science research in Europe,” *European Journal of Information Systems*, vol. 17, no. 5, pp. 470–475, 2008. Available: https://doi.org/10.1057/ejis.2008.44

[6] O. Saltuk and I. Kosan, “Design and creation,” presentation, Ludwig Maximilian University of Munich, 2014. Available at: https://www.medien.ifi.lmu.de/lehre/ss14/swal/presentations/topic2-saltuk_kosan-DesignAndCreation.pdf

[7] A. R. Hevner, “A Three Cycle View of Design Science Research,” *Scandinavian Journal of Information Systems*, vol. 19, no. 2, pp. 87–92, 2007. Available: https://aisel.aisnet.org/sjis/vol19/iss2/4

[8] A. R. Hevner, “Designing Informing Systems: What Research Tells Us,” in *Informing Science Institute Conference (SITE2015)*, 2015.

[9] S. Gregor, L. Chandra Kruse, and S. Seidel, “Research perspectives: the anatomy of a design principle,” *Journal of the Association for Information Systems*, vol. 21, no. 6, article 2, 2020. Available: https://doi.org/10.17705/1jais.00649

[10] K. Peffers, T. Tuunanen, M. A Rothenberger, and S. Chatterjee, “A design science research methodology for information systems research,” *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007. Available: https://doi.org/10.2753/MIS0742-1222240302

[11] J. Vom Brocke, R. Winter, A. Hevner, and A. Maedche, “Special issue editorial–accumulation and evolution of design knowledge in design science research: a journey through time and space,” *Journal of the Association for Information Systems*, vol. 21, no. 3, 2020. Available: https://doi.org/10.17705/1jais.00611

[12] F. Gacenga, A. Cater-Steel, M. Toleman and W.G. Tan, “A Proposal and Evaluation of a Design Method in Design Science Research,” *The Electronic Journal of Business Research Methods*, vol. 10, no. 2, pp. 89–100, 2012. Available: https://academic-publishing.org/index.php/ejbrm/article/view/1291

[13] P. Checkland and J. Poulter, “Soft Systems Methodology,” in *Systems Approaches to Making Change: A Practical Guide*, Springer, pp. 191–142, 2010. Available: https://doi.org/10.1007/978-1-84882-809-4_5

[14] P. Checkland, “Soft systems methodology: a thirty-year retrospective,” *Systems Research and Behavioral Science*, vol. 17, no. S1, pp. S11–S58, 2000. Available: https://doi.org/10.1002/1099-1743(200011)17:1+<::AID-SRES374>3.0.CO;2-O

[15] H. Augustsson, K. Churruca, and J. Braithwaite, “Change and improvement 50 years in the making: a scoping review of the use of soft systems methodology in healthcare,” *BMC Health Services Research*, no. 20, pp. 1–13, 2020. Available: https://doi.org/10.1186/s12913-020-05929-5

[16] H. Augustsson, K. Churruca, and J. Braithwaite, “Re-energising the way we manage change in healthcare: the case for soft systems methodology and its application to evidence-based practice,” *BMC Health Services Research*, no. 19, pp. 1–11, 2019. Available: https://doi.org/10.1186/s12913-019-4508-0

[17] K. Kotiadis, A. A. Tako, E. A. Rouwette, C. Vasilakis, J. Brennan, P. Gandhi, and P. Webb, “Using a model of the performance measures in Soft Systems Methodology (SSM) to take action: a case study in health care,” *Journal of the Operational Research Society*, vol. 64, no. 1, pp. 125–137, 2013. Available: https://doi.org/10.1057/jors.2012.21

[18] O. L. Bjerke, “Soft Systems Methodology in action: A case study at a purchasing department,” *Report/IT University of Göteborg 2008: 034*, 2008. Available: http://hdl.handle.net/2077/10551

[19] R. Baskerville, J. Pries-Heje, and J. Venable, "Soft design science methodology," in *Proceeding of the 4th International Conference on Design Science Research in Information Systems and Technology*, pp. 1–11, 2009. Available: https://doi.org/10.1145/1555619.1555631

[20] R. L. Baskerville, "Investigating information systems with action research," *Communications of the Association for Information Systems*, vol. 2, article 19, 1999. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ecd26de521f27c8ce85e847206e44ba99524b2cb

[21] G. Susman and R. Evered, "An Assessment of the Scientific Merits of Action Research," *Administrative Science Quarterly*, vol. 23, no. 4, pp. 582–603, 1978. Available: https://doi.org/10.2307/2392581

[22] S. Gregor, "The nature of theory in information systems," *MIS Quarterly*, vol. 30, no. 3, pp. 611–642, 2006. Available: https://doi.org/10.2307/25148742

[23] D. Jones and S. Gregor, "The Anatomy of a Design Theory," *Journal of the Association for Information Systems*, vol. 8, no. 5, pp. 312–335, 2007. Available: https://doi.org/10.17705/1jais.00129

[24] S. Gregor, "Design Theory in Information Systems," *Australasian Journal of Information Systems*, vol. 10, no. 1, pp. 14–22, 2002. Available: https://doi.org/10.3127/ajis.v10i1.439

[25] H. A. Simon, *The New Science of Management Decision*. American Psychological Association, 1960. Available: https://doi.org/10.1037/13978-000

[26] J. F. Nunamaker, M, Chen, and T. D. Purdin, "Systems development in information systems research," *Journal of Management Information Systems*, vol. 7, no. 3, pp. 89–106, 1990. Available: https://doi.org/10.1080/07421222.1990.11517898

[27] W. O. Galitz, *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*. John Wiley & Sons, 2007.

[28] ISO. International Standards Association ISO/TC 176/SC 1 9000:2000 "Quality Management Systems – Fundamentals and Vocabulary" 2000.

[29] P. Checkland, "Systems Thinking," in *Rethinking Management Information Systems: An Interdisciplinary Perspective*, John Wiley and Sons, pp. 45–56, 1999. Available: https://doi.org/10.1093/oso/9780198775331.003.0004

[30] A. Monk and S. Howard, "Methods & tools: the rich picture: a tool for reasoning about work context," *Interactions*, vol. 5, no. 2, pp. 21–30, 1998. Available: https://doi.org/10.1145/274430.274434

[31] S. Bell, T. Berg, and S. Morse, "Towards an understanding of rich picture interpretation," *Systemic Practice and Action Research*, vol. 32, pp. 601–614, 2019. Available: https://doi.org/10.1007/s11213-018-9476-5

[32] K. Ishikawa, *Guide to Quality Control*. Asian Productivity Organization, 1986.

[33] L. Liliana, "A new model of Ishikawa diagram for quality assessment," in *IOP Conference Series: Materials Science and Engineering*, vol. 161, pp. 012099, IOP Publishing, 2016. Available: https://doi.org/10.1088/1757-899X/161/1/012099

[34] K. C. Wong, "Using an Ishikawa diagram as a tool to assist memory and retrieval of relevant medical cases from the medical literature," *Journal of Medical Case Reports*, no. 5, pp. 1–3, 2011. Available: https://doi.org/10.1186/1752-1947-5-120

[35] A. Veselý, "Problem tree: A problem structuring heuristic," *Central European Journal of Public Policy*, vol. 2, no. 2, pp. 60–81, 2008.

[36] B. Abima, A. Nakakawa, and G. M. Kituyi, "Service-Oriented Framework for Developing Interoperable e-Health Systems in a Low-Income Country," *The African Journal of Information Systems*, vol. 15, no. 3, pp. 141–174, 2023. Available: https://digitalcommons.kennesaw.edu/ajis/vol15/iss3/1/

[37] S. Eelu and A. Nakakawa, "Framework towards enhancing adoption of electronic payment in a developing economy: a case of Uganda," *The African Journal of Information Systems*, vol. 10, no. 3, pp. 222–245, 2018. Available: https://digitalcommons.kennesaw.edu/ajis/vol10/iss3/5

[38] I. Arinaitwe, "A framework for GIS-enabled public e-participation in municipal solid waste management," Ph.D. dissertation, Makerere University, 2020.

[39] M. Nagwovuma, "Framework for Evaluating Performance of Electronic Health Information Systems," unpublished Ph.D. dissertation, Makerere University, 2022.

[40] A. Nakakawa, P. V. Bommel and H. A. Proper, “Supplementing enterprise architecture approaches with support for executing collaborative tasks – A case of TOGAF ADM,” *International Journal of Cooperative Information Systems*, vol. 22, no. 02, article 1350007, 2013. Available: https://doi.org/10.1142/S021884301350007X

[41] F. Namagembe, A. Nakakawa, F. P. Tulinayo, H. A. Proper and S. Overbeek, “Towards an E-Government Enterprise Architecture Framework for Developing Economies,” *Complex Systems Informatics and Modeling Quarterly*, vol. 35, pp. 30–66, 2023. Available: https://doi.org/10.7250/csimq.2023-35.02

[42] F. P. Tulinayo, E. Mwesigwa, A. Mugisha, and H. Nyende, “Explore the factors that influence smallholder farmers’ use of ICTs as enablers for knowledge sharing,” *African Journal of Rural Development*, vol. 7, no. 4, pp. 537–562, 2022. Available: https://afjrdev.org/index.php/jos/article/view/314

[43] R. S. Mukisa, “Framework of Clinical Decision Support Solutions for Enabling Knowledge Sharing in the Management of Acute Child Malnutrition in Uganda,” unpublished M.S. thesis, Makerere University, 2021.

[44] E. Kuhimbisa, “Method for Developing Usable Integrated Information Systems: A Case of the Justice Law and Order Sector in Uganda,” unpublished M.S. thesis, Makerere University, 2021.

[45] J. Galaige, “Method for Leveraging Social Media to Support Customer Relationship Management in Small and Medium Enterprises in Uganda,” unpublished M.S. thesis, Makerere University, 2014.

[46] R. O. Briggs, B. A. Reinig, and G.-J. de Vreede, “Meeting satisfaction for technology-supported groups: An empirical validation of a goal-attainment model,” *Small Group Research*, no. 37, no. 6, pp. 585–611, 2006. Available: https://doi.org/10.1177/1046496406294320

Appendices

Appendix 1. Requirements for TBUFI

Design Guideline	**Implied Requirement for TBUFI (derived from the source definition of the design guideline as specified in Section 3)**
DG1. Aligning	TBUFI must enable a researcher *to demonstrate how the design process of an artifact progressed or evolved in a specific DSR project by ensuring that the design process is explicitly documented and justified.*
DG2. Positioning	TBUFI must prompt a researcher *to clearly specify the following:* *a) The relevant problem addressed in a specific DSR project;* *b) Subsets of the problem space in a specific DSR project;* *c) The devised solution in a specific DSR project;* *d) Subsets of the solution space in a specific DSR project.* *e) Substantial evaluation findings that confirm the relationship between the problem and the solution in a specific DSR project.*
DG3. Grounding	TBUFI must prompt a researcher to *explicitly state the extent to which processes executed and corresponding results in a specific DSR project, build on existing knowledge or support accumulation and evolution of knowledge in a specific problem space and/or specific solution space*
DG4. Advancing	TBUFI must prompt a researcher to *specify how processes executed and/or results obtained in a particular completed DSR project augment or extend or improve existing knowledge on the design process of artifacts*.
DG5. Problem relevance	TBUFI must prompt a researcher to specify and demonstrate the importance and urgency of the problem that is to be addressed by a technology-oriented solution or intervention.
DG6. Research Contributions	TBUFI must prompt a researcher to ensure that research efforts ultimately yield confirmable contributions in the form of additions or improvements to at least one of the following knowledge areas: *a) Existing foundational approaches (theories, methods, frameworks, constructs, models, methods, and instantiations).* *b) Existing evaluation approaches (methods, techniques)* *c) Existing designs of products.* *d) Existing design processes.*
DG7. Research Rigor	TBUFI must prompt a researcher to *ensure that research efforts in constructing an artifact involve applying existing rigorous (or detailed and scientifically acceptable) approaches.*
DG8. Design as a Search Process	TBUFI must prompt a researcher *to explore, assess, and determine the extent to which particular means appropriately satisfy or conform to accepted practices in the problem space and solution space of a specific DSR project.*
DG8.1. Design Evaluation	TBUFI must prompt a researcher to select appropriate evaluation methods that can be used to expose an artifact to stakeholders who are its target users and incite or encourage them to rigorously examine the quality, utility, and efficacy of the artifact.
DG8.2. Relevance cycle – *iterative field testing*	TBUFI must prompt a researcher to iteratively expose the artifact to field testing experiments in the application environment so as to ascertain deficiencies in its quality attributes.
DG8.3. Rigor Cycle – knowledge contribution	TBUFI must prompt a researcher to consider that the created artifact as a contribution to the knowledge base, only if it extends an existing artifact or if it offers new experiences or expertise, or if it is a new artifact or design process.
DG8.4. Design Cycle – *rigorous evaluation*	TBUFI must prompt a researcher to ensure that: a) The artifact first undergoes rigorous evaluation in a controlled environment until it is ready for field testing settings, and b) The performance of an artifact is rigorously assessed in the application environment, prior to considering its inclusion in the knowledge base.
DG9. Communication of Research	TBUFI must prompt a researcher *to devise avenues of effectively communicate the ultimate research result to a technology-oriented audience (with subject matter experts); and a management oriented audience (or non-technology oriented audience).*
DG10. Relevance cycle – identified need	TBUFI must prompt a researcher *to explicitly specify the need or problem that should drive research in a specific DSR project; and specific quality criteria for assessing suitability of the desired solution.*
DG11. Rigor Cycle – skillful adoption	TBUFI must prompt a researcher to: *a) Specify existing foundational approaches that constitute the knowledge base of the problem space and solution space of a specific DSR project.* *b) Determine appropriate foundational approaches that can offer insights into how to devise the desired solution, or can be innovatively extended or improved to obtain the desired solution.*
DG12. Design Cycle –*iterative creation*	TBUFI must prompt a researcher to execute that the hard task of building an artifact entails numerous iterations of: *a) Drawing requirements of the desired solution from the problem environment, and* *b) Drawing foundational insights from the knowledge base of a project to address the requirements.*

Design Guideline	Implied Requirement for TBUFI (derived from the source definition of the design guideline as specified in Section 3)
DG13. Generate and test design alternatives	TBUFI must prompt a researcher to demonstrate that the design process of a solution involved repeatedly or iteratively: *a) Generating potential alternative designs of a particular solution for addressing a given requirement;* *b) Testing and assessing the extent to which the potential design alternatives fulfill a given set of constraints or criteria or rules associated with effectively or efficiently realizing a given requirement; and* *c) Selecting a satisfactory design alternative.*
DG14. Usability (including learnability, efficiency, satisfaction, and consistency)	Efforts should be undertaken to ensure that: *a)* The design of TBUFI is e*asily and independently understood by a target user*. *b)* TBUFI *can be effectively used by a target user, with minimal or no support from its originators or expert users.* c) A first-time user of TBUFI can accomplish elementary tasks (i.e., learnability). d) A user who has understood or is knowledgeable about TBUFI can promptly accomplish tasks (i.e. efficiency). e) A user is pleased to use TBUFI or has a good experience when using TBUFI or when executing tasks that constitute it (i.e., satisfaction). *f)* Related elements in TBUFI have a similar look (i.e., consistency).
DG15. Traceability	Efforts should be undertaken to ensure that: a) The source of each element in TBUFI can be easily identified. b) The relationship between elements and sub-elements of TBUFI can be easily understood. c) Elements of TBUFI that address specific design guidelines can be easily identified.

Appendix 2. Mapping Tasks in TBUFI to Design Guidelines and Expected Outputs

Tasks in TBUFI	Sub Tasks in TBUFI	Design Guidelines (from the literature in Section 3)	Outputs
T1. Create Conceptual Model(s) to align Stakeholder Understanding of the Problem	T1.1. Explore the various stakeholder views on the magnitude and scope of the problem.	• DG5. Problem Relevance • DG10. Relevance cycle – identified need	Catalog of issues or challenges (with codes) that characterize the problem
	T1.2. Create conceptual models that describe and classify issues in the problem context	• DG2. Positioning • DG11. Rigor cycle – skilled adoption • DG10. Relevance cycle – identified need	
	T1.3. Prioritize issues with respect to available resources and assign them codes or identifiers.	• DG5. Problem Relevance • DG10. Relevance cycle – identified need	
T2. Derive requirements and their implementation options	T2.1. Derive conceptual models that describe and classify requirements that must be addressed, and assign traceability codes.	• DG2. Positioning • DG10. Relevance cycle – identified need • DG11. Rigor cycle – skilled adoption	Catalogue of requirements (with codes) for a framework
	T2.2. Identify potential actions for addressing each requirement or design alternatives of the desired solution framework.	• DG3. Grounding • DG13. Generate and test design alternatives • DG11. Rigor cycle – skilled adoption • DG12. Design cycle – iterative creation • DG8. Design as a search process • DG7. Research rigor	Design Alternatives of the desired framework for guiding an intervention
	T2.3. Devise implementation options for each potential action or design alternative and elaborate them using purposeful activity models (i.e., models which show elements that constitute or operationalize each design alternative of the desired solution framework).		
	T2.4. Compare and assess views of activity models for potential actions or design alternatives and select the most appropriate.		
T3. Synthesize Design Decisions to Constitute Framework	T3.1. Specify appropriate design alternatives as Design Decisions or Design Choices and assign traceability codes	• DG3. Grounding • DG11. Rigor cycle – skilled adoption • DG12. Design cycle – iterative creation • DG7. Research Rigor • DG8. Design as a search process • DG6. Research contribution	Set of Design Decisions (with codes) taken to construct a framework
	T3.2. Determine Design Decision and/or existing artifact(s) that can be used as a pivot for enabling orchestration of all Design Decisions		
	T3.3. Use the selected pivotal Design Decision or existing artifact to synthesize all Design Decisions into a framework.		The first version (version 1) of

Tasks in TBUFI	Sub Tasks in TBUFI	Design Guidelines (from the literature in Section 3)	Outputs
	T3.4. Specify assumptions and constraints that shape the synthesis of all design decisions into a framework.	DG14. Usability	the design of a framework
	T3.5. Identify gaps in the derived synthesis of design decisions and devise rationality and granularity aspects that can fix them	DG14. Usability	
T4. Deliberate and Assess Feasibility of the Constituted Framework	T4.1. Deliberate Design Decisions and corresponding rationality and granularity aspects that constitute the framework	• DG9. Communication of research • DG6. Research contribution	Transitional versions (with numbers) of the design of a framework
	T4.2. Perform tradeoff analysis of stakeholder viewpoints with respect to the constituted framework	• DG3. Grounding • DG8. Design as a search process • DG13. Generate and test design alternatives • DG11. Rigor cycle – skilled adoption • DG12. Design cycle – iterative creation	
	T4.3. Identify components of the framework in which prototyping can be applied to further assess framework's feasibility and suitability		
	T4.4. Derive and assess additional design alternatives to address feasible viewpoints		
T5. Update the framework and its design process	T5.1. Revise synthesis of design decisions and rationality and granularity aspects to fix required changes	• DG2. Grounding • DG12. Design cycle – iterative creation • DG14. Usability	The final version of the design of a framework
	T5.2. Specify situational factors associated with adopting and implementing the framework with respect to context	• DG12. Design cycle – iterative creation • DG8. Design as a search process	
	T5.3. Continuously improve the framework until saturation, by repeating tasks T1.1 to T5.2		
	T5.4. Specify issues encountered in executing tasks T1.1 to T5.3, and measures taken to address them	• DG4. Advancing • DG9. Communication of research	