2019

A socio-technical-based process for questionnaire development in requirements elicitation via interviews

Abdullah Wahbeh
Slippery Rock University of Pennsylvania

Surendra Sarnikar
California State University, East Bay

Omar F. El-Gayar
Dakota State University

Recommended Citation

Wahbeh, A., Sarnikar, S., & El-Gayar, O. (2019). A socio-technical-based process for questionnaire development in requirements elicitation via interviews. Requirements Engineering, 1-21.

Requirements Engineering
https://doi.org/10.1007/s00766-019-00324-x

ORIGINAL ARTICLE

A socio-technical-based process for questionnaire development in requirements elicitation via interviews

Abdullah Wahbeh[1] · **Surendra Sarnikar**[2] · **Omar El-Gayar**[3]

Received: 20 June 2018 / Accepted: 2 August 2019

Abstract
Software development is the process of building systems that solve users' need and satisfy stakeholders' objectives. Such needs are determined through requirements elicitation, which is considered an intensive, complex, and multi-disciplinary process. Traditional methods of elicitation often fail to uncover requirements that are critical for successful and wide-scale user adoption because these methods primarily focus on the technical aspects and constraints of the systems rather than considering a socio-technical perspective. The success of information system development involves the identification of the social, organizational and technical features of the systems, which in turn can result in a more acceptable system by users. In this paper, we propose a requirements elicitation process based on socio-technical (ST) systems theory. The process leverages ST system components to help identify a set of ST imbalances, which in turn help in requirements elicitation. The applicability of the process is demonstrated using empirical investigation with a randomized two-group experimental design, where the objective is to see the potential of the proposed process to enhance analysts' understanding of socio-technical aspects of a domain, interview readiness, and questionnaire quality.

Keywords Software development · Requirement elicitation · User interviews · Questionnaire development · Socio-technical systems · Design research

1 Introduction

Software development is the process of building systems that meet users' needs and satisfy stakeholders objectives [1]. Requirements elicitation, a step in software development, is used to discover, reveal, articulate, and understand users, stakeholders, and customers' requirements [2, 3]. Requirements elicitation is an intensive, complex, and difficult endeavor [4, 5], and its multi-disciplinary nature further adds to this complexity [6]. As a result, improving the elicitation process is a critical goal for the development of information systems since incomplete requirements are a primary cause of system development failure [7–9]. In essence, successful collection of users' requirements is crucial for the success of information system development [10]. In addition, many system design problems today are new, complex and difficult. Effective requirements collection in such complex projects would require participation of a large number of stakeholders [11], are often outside the system designers' normal experience, and have limited availability of expertise. As a consequence, many such projects are susceptible to serious project failures [12].

Currently, several requirements elicitation techniques exist to help understand users' needs [1]. However, most of the techniques are limited in terms of their applicability as they depend on the practitioners' expertise for effective use and are limited when the analyst may lack necessary expertise to prepare for requirements elicitation [6, 13]. In other cases, it is necessary to investigate and examine the application domain in which the system will reside [6]. Such investigation should not be limited to technical aspects of

✉ Abdullah Wahbeh
abdullah.wahbeh@sru.edu

Surendra Sarnikar
surendra.sarnikar@csueastbay.edu

Omar El-Gayar
omar.el-gayar@dsu.edu

[1] Slippery Rock University, Slippery Rock, PA 16057, USA

[2] California State University, East Bay, 25800 Carlos Bee Boulevard, Hayward, CA 94542, USA

[3] Dakota State University, 820 N Washington Ave, Madison, SD 57042, USA

Published online: 12 August 2019

the problem domain but should also include the political, organizational, and social aspects related to the system [6, 13], which requires taking a socio-technical (ST) approach in system design and requirements elicitation. Current elicitation techniques are limited in their ability to elicit ST requirements as they are either critically dependent on the selection of the right participants in order to guarantee the successful elicitation of requirements, are resource intensive and require the analysts to be familiar with formal methods and techniques that include a significant learning curve and considerable training to be successfully applied, or lack a systematic way to analyze, generate, and validate requirements.

Interviews are one of the most commonly used techniques for collecting requirements from the actors in an organization [7, 9, 14–17]. Interviews, whether they are structured, semi-structured, or unstructured, are considered one of the most effective requirements elicitation techniques. The interview process consists of many phases, including creation of questions, identifying and selecting interviewees, planning the interview process, conducting the interview, and closing the interview meeting [18]. Without proper attention to these tasks, the system analysts are likely to "short-cut" the requirements elicitation process, which in turn affects the completeness and accuracy of the elicited requirements [9].

In interviews, system analysts collect the necessary requirements with a set of questions to gain necessary information about their needs [19]. However, there is limited guidance about the interviews contents or questions [10, 20], and the kind of questions or inquiry that is most effective [9]. While interviews can lead to transfer of tacit knowledge from stakeholders to analysts if conducted effectively, they can also lead to unclear and incomplete requirements due to ambiguity and inadequate upfront preparation [21]. Empirical studies show that the "careful preparation of interviews may have a much more marked effect than experience" [22]. In other words, "*a novice analyst who prepares the interview well beforehand is even capable of eliciting more information than an experienced analyst*" [22].

Overall, the success of information system development requires eliciting social and organizational features of the systems in addition to the technical ones [23], which in turn can result in a more acceptable system by users. Understanding the human context in which software operates is critical to its adoption [24]. Recent research in requirements engineering has emphasized the importance of capturing of social aspects of an organization in domain models [25], and understanding the socio-political issues during the requirements elicitation processes since they may impact the functional and non-functional requirements of the system [26].

Ensuring that the requirements elicitation processes encompass technical and social aspects of the organization is especially important when building solutions for complex problems such as knowledge management [27], or ICT solutions that require elicitation of social and technical requirements from multiple stakeholders with different knowledge [28, 29] or digital motivation systems as parts of organizational and social structure of a business [30]. The importance of considering socio-technical perspective had been acknowledged in multiple complex problem areas including healthcare [31] and security [32].

In that regard, a ST-based process can provide a better understanding of how an organization undertakes its work and provides a clear process for designing and implementing new work systems [33]. A ST systems perspective can contribute to a better understanding of how human, social, and organizational factors affect the ways that work is done and technical systems are used [34] and provide an underlying framework for requirement elicitation activities such as interviews, particularly, in complex problem domains.

In this research, we present a process for the systematic identification of users' requirements for information systems using a Socio-Technical systems model. The key objective of the proposed process is to improve the requirements elicitation process, more specifically, the quality of the questionnaire for requirements elicitation interviews. This is achieved through the development of a model to understand the social as well as the technical characteristics of the problem domain and using the model to develop a requirements elicitation questionnaire.

The rest of the paper is organized as follows; first, we present a review of ST systems and requirements elicitation techniques followed by a problem statement. Next, we discuss the proposed ST process for requirements elicitation followed by an empirical evaluation of the resulting interview questionnaire. Finally, we discuss results and present concluding remarks and an agenda for future work.

2 Related work

2.1 Socio-technical (ST) systems

Baxter and Sommerville [35] refer to ST systems design (STSD) methods as "an approach to design that consider human, social and organizational factors, as well as technical factors in the design of organizational systems". The main premise in ST design is to make sure that, whenever possible, the technical and human factors are given equal weight in the design process [36]. A key criterion for successful design of ST systems is the gathering of ST requirements in the first place.

The term ST systems was originally coined by Emery, Trist [37] to denote any system that consists of complex interaction between humans, machines and the environmental aspects of the work system. Such interactions apply to most systems in the information age [35]. In ST systems, human, organizational and software actors rely heavily on each other in order to fulfill their respective objectives [38]. In that regard, Cherns [39, 40] identifies nine principles for ST design that includes compatibility, minimal critical specifications, the ST criterion, the multi-functionality principle, boundary location, information flow, support congruence, design and human values, and incompletion. Later, Clegg [23] presents a revised set of these socio-technical principles to guide system design, and the applicability and contributions of such principles for system design.

The principles are intended to be applied to the design of new systems, and they attempt to provide a more integrated perspective than is apparent in existing formulations. These principles fall into three highly interrelated types namely meta-principles that are intended to capture a worldview of design, content principles that are focused on more specific aspects of the content of new designs, and process principles that are related to the design process. As stated by the author, these principles are to be used by system managers, users, designers, technologists and social scientists. They provide inputs to those who are engaged collaboratively in design. While ST systems and design principles have been widely discussed in the literature [12, 35, 41, 42] and aim at giving equal weight to the social and technical issues during system design [12], there is very limited literature on specific methods for incorporate ST considerations into requirement elicitation activities, such as questionnaire design in preparation for interviews.

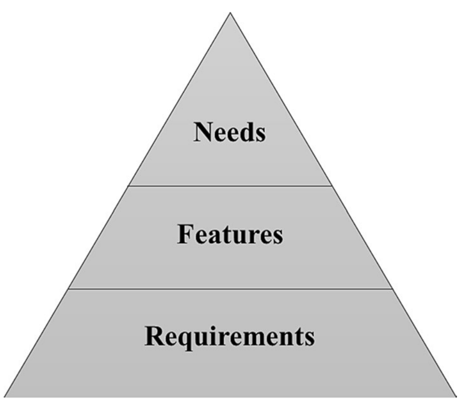

Fig. 1 The Requirements Pyramids

2.2 System requirements

Depending on the format, source, and common characteristics, the system requirements can be split into a three level hierarchy as shown in Fig. 1 [43].

The top most level of the hierarchy consists of stakeholders needs [43, 44]. Understanding these needs will help the development team build a better system [44]. It is common that a project consists of 5–15 of these high-level needs. The lowest level in the hierarchy consists of the most specific and detailed requirements of the system. In general, all the important behavioral characteristics of a system are captured in the requirements specification [45, 46]. The mid-level of the hierarchy, which is the focus of this paper, is features, and is defined as "a grouping or modularization of individual requirements within that specification" [45]. Features are easily expressed in natural language and consist of a short phrase as shown in Table 1. Rarely, if ever, are features elaborated in more detail. "Features are also very helpful constructs for early product scope management and the related negotiation and trade-off processes" [44].

Features, which represent a high level of abstraction of requirements, are very useful to describe the functionality of a new system without the need to drill down into too much details [44]. Therefore, they are very helpful in managing the complexity of newly developed systems or for an increment to an existing system. As such, system functionalities should be abstracted as features, with a maximum number of 25–99 features, with fewer than 50 preferred [44, 47]. This way we end up with a small and manageable amount of information, where 25–99 features can be categorized and arranged, which in turn will help describe and communicate the gestalt of the system. In later stages, these features are developed into detailed requirements that are specific enough for implementation purposes. For the purpose of this work, we will focus on requirements at the features level of the requirements pyramid [44].

Table 1 Features examples

Application domain	Example of a feature
Elevator control system	Manual control of doors during fire emergency
Inventory control system	Provide up-to-date status of all inventoried items
Defect tracking system	Provide trend data to assess product quality
Payroll system	Report deductions-to-date by category

2.3 Requirement elicitation

Requirements derive the whole software development process, especially during the design phase where much of the system's qualities are identified [48]. Requirements engineering is the branch of software engineering that is concerned with objectives, functionalities and constraints associated with any software system [49]. Software systems requirements engineering "is the process of discovering and identifying stakeholders and their needs, and documenting these needs in a form that is amenable to analysis, communication, and subsequent implementation" [50].

In requirements engineering, requirements elicitation, sometimes referred to as requirements gathering or requirements collection, is considered one of the first steps in the software life cycle; its importance is becoming more and more prominent [51] as it helps analysts and those involved in system development to learn and discover users and stakeholders' needs [52]. Requirements elicitation is the process of seeking, identifying, discovering, acquiring, and elaborating information systems requirements [6]. Requirements elicitation is one of the most critical steps in software development where poor execution of the elicitation process can results in a complete failure of the project [1, 53]. Furthermore, requirements elicitation involves examining and reviewing existing systems, any relevant documents, interviewing relevant users and stakeholders' [15]. Given that the requirements elicitation process consists of many knowledge-intensive processes [4], accessing such knowledge is considered crucial to the success of the requirements elicitation process, which in turn can result in a more acceptable and successful systems. Therefore, obtaining such knowledge before proceeding with the requirements elicitation task can be beneficial for an analyst [52]. "Knowledge of the business domain such as insurance claim and human resources is crucial to analysts' ability to conduct good requirements analysis" [15]. Prior domain knowledge may especially be necessary for ensuring completeness of requirements [54].

There are different requirements elicitation techniques that can help analysts to identify and elicit users and stakeholders' needs. These techniques include but are not limited to interviews, ethnography, prototyping, data gathering from existing systems, formal methods, card sorting, brainstorming, requirements workshop, JAD, scenarios, and viewpoints [6, 55]. Analysts are not limited to one specific requirements elicitation techniques and they can use different techniques together depending on the situation and the problem domain [52, 54]. One of the most popular and widely used methods for eliciting the necessary requirements is interviews [22, 52, 54, 56]. Interviews are considered effective techniques for collecting requirements using structured, semi-structured, or unstructured questions depending of the situation. In an extensive literature review, none of the available requirements elicitation methods were found to have advantages over semi-structured interviews [22, 56] or structured interviews [1, 56]. Interviews have been rated as being the most effective technique used by analysts for gathering the necessary requirements [57].

Accounting for the social technical perspective, Bryl et al. [38] proposed a tool that supports the process of requirements analysis for ST systems. The proposed tool adopts several planning techniques that can be used for exploring the space of requirements alternative and the number of social criteria for evaluation. RESCUE is an integrated method proposed by Jones and Maiden (2005) for specifying requirements for complex ST systems. RESCUE integrates several components that were used to elicit requirements for ST systems. These components include human activity modeling, creative design workshops, system goal modeling using the *i** notation, systematic scenario walkthroughs, and best practice in requirements management. Recent research in this area includes a method engineering methodology for knowledge management solutions that includes a focus on social and cultural aspects [27].

Alistair and Shailey [58] proposed a method for analyzing ST systems requirements by analyzing dependencies between computer systems and users/stakeholders in an operational environment. Also, the authors have used the scenarios that describe the system and its context to build an environmental model based on *i** notation. Mavin, Maiden [59] suggested the use of scenarios for eliciting requirements for ST systems. Generating and walking through scenarios is considered one of the effective techniques for electing requirements. Ethnographic techniques have also been used for eliciting ST systems requirements by gathering the necessary data on social issues and then generating the requirements from such data [60].

Overall, most of the proposed processes for requirements elicitation only focus on either technical requirements or specific problem areas such as e-collaboration [29], knowledge management [27] or web design. These processes include but not limited to processes that use scenarios [61–63], using processes that combine scenarios, prototypes, and design rationale [64], communication-based model of elicitation [10] or multi-perspective models [29] for e-collaboration, and using viewpoints to elicit requirements [65]. While such techniques are helpful for eliciting technical requirements and system features necessary for near term adoption of information systems, they are not well suited for eliciting deeper social and ST requirements that can govern continued use of information systems.

3 Statement of the problem

Many factors beyond technological issues influence the success, adoption, and use of information technology (IT) solutions. IT solutions often fail to encompass a holistic ST view, where the design of such systems should account for the intrinsic and interrelated characteristics of the underlying tasks, actors, technologies, and structure. Moreover, difficulties in defining and collecting system requirements have been identified as one of the major factors behind the failure of 90% of large software projects [66].

Despite the fact that different methods are used in eliciting systems requirements, these methods are dependents on the practitioners expertise [58] with different preferences and skills [67], thereby affecting the quality of the elicited requirements. With respect to system walkthroughs, the selection of the participants is a critical factor for the successful implementation of such approach and many different participants should be included in the walkthroughs, e.g., end-users, project-leaders, technical experts, and possibly members of the management board. Dealing with such number of necessary participants poses a challenging task on the analyst to choose the appropriate participants [68]. Moreover, frameworks such as RESCUE require analysts to be familiar with specific formal methods, notation and techniques [35], which are resource intensive and requires considerable training in order to be successfully applied [69].

Analyst "from traditional software engineering backgrounds may sometimes focus on the solution not the problem, and rely on only those techniques that they are familiar with for all situations" [6, 13]. In some cases, it is necessary to investigate and examine the application domain in which the system will reside [6]. Such investigation should not be limited to technical aspects of the problem domain but should also include the political, organizational, and social aspects related to the system [6, 13]. Holistic and systems thinking is especially necessary for eliciting requirements in complex problem situations [70].

Interviews are exploratory in nature and tend to be less guided, and characterized by a set of questions, such as "what the system should do", where the depth and breadth of each of these questions is largely dependent on the analysts skills and experience [71], where those analysts usually do not employ any structured or rigorous processes to address requirements elicitations. According to the literature, in some cases, novice analysts are capable of eliciting the necessary requirements exactly the same way as experienced analysts. In fact, careful preparation of interviews has a much more marked effect than analyst experience [22].

As a result, there is a need for a clear and well-defined systematic process for eliciting system requirements that reduces the overreliance on the practitioners' experience and dependence on the quality of user participation. The process should be based on the notion of a ST model for system design to provide weightage to both technical and human factors in the design process [36]. In addition, in the ST System literature, there is a lack of a midrange theoretical model for ST systems analysis and design. Previous studies on ST systems [35, 36, 39, 40, 72, 73] provide a high-level overview of ST design which need to be extended to provide specific steps, or process for the purpose of eliciting requirements from a ST perspective. Horkoff et al. [74] argue that "understanding and evaluating the ST divide between complex human organizations and complex systems is a particularly hard problem" and that given the difficulty of the Requirements engineering field, new ideas are needed to solve such hard problems. In the following sections, we present a new approach for systematic identification of users' requirements for information systems based on ST design theory.

4 Design methodology

The research presented here follows the principles of design science research. Design science research seeks the creation and evaluation of design artifacts such as conceptual models and software systems and the development of new generalizable knowledge about design processes and products, while solving important problems with these artifacts and knowledge [75, 76]. Following Peffers et al. [77] (Fig. 2), the design science research phases include identifying the problem, defining solution objectives, designing the artifact, and demonstrating, evaluating, and communicating research results.

We have employed a design and development centered approach in our research. The main problem we intend to address is the limited consideration of a ST perspective in system requirement and analysis processes. Our objective in this research is to enhance the systems requirements elicitation process with a ST perspective that focuses on both social and technical perspectives and resulting in more comprehensive requirements that capture the breadth and depth of the problem domain. We address this problem by building on past literature on ST models and developing a requirements elicitation process model for ST systems. We demonstrate its feasibility using illustrative examples of questionnaire design in knowledge sharing and self-care problem areas. We then conduct an experimental evaluation to assess the extent to which the constructed ST process model helps the analysts develop questionnaires that help capture the breadth and depth of the problem domain.

We have used an iterative approach to design the proposed socio-technical requirements elicitation process. In

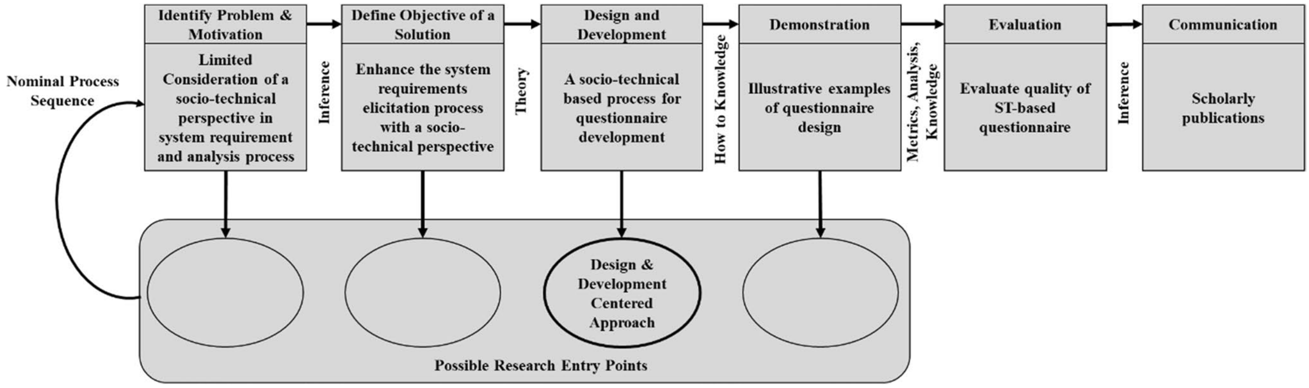

Fig. 2 Design Science Research Phases by Peffers et al. [77]

our first iteration, we explored multiple theoretical perspectives on social and technical inter-relationships in information systems. We then built on the ST model [73] to identify characteristics of socio-technical systems and their imbalances. In further iterations, we refined the model by comparing the model imbalances with those predicted in the literature. Finally, we extended the model to provide further utility by using it to help develop requirements elicitation questionnaires. In the following section, we provide a detailed overview of the proposed socio-technical requirements elicitation process (STREP).

5 Process for questionnaire development in requirements elicitation via interviews

The proposed process consists of three steps, namely identifying STS properties, identifying ST imbalances, and mapping ST imbalances to questionnaire statements as follows (Fig. 3):

1. The process starts by defining the four main ST components. For each component of the ST model, a list of relevant attributes is identified from literature.
2. Imbalances are identified among the combinations of the ST components' attributes. Extant literature is used to confirm that the list of imbalances exists in relevant systems.
3. Finally, the list of identified imbalances is converted to potential questionnaire statements. These statements represent system features (high-level requirements statements) that guide the interviews during requirement elicitation.

5.1 Step 1: identify ST components' properties

The ST model shown in Fig. 4 consists of four main components, namely tasks, actors, structure, and technology. In order to provide a comprehensive list of ST imbalances that can help identifying ST system requirements, it is necessary to provide a list of properties that define each of these components. *Tasks* describe the goals and purpose of the system and the way work/activities are accomplished, *actors* refer to users and stakeholders who perform and influence the work/activities, *structure* denotes the surrounding project and institutional arrangements, and *technology* refers to tools and interventions used to perform the work/activities.

In order to provide a comprehensive list of ST imbalances that can help identifying ST system features, it is necessary

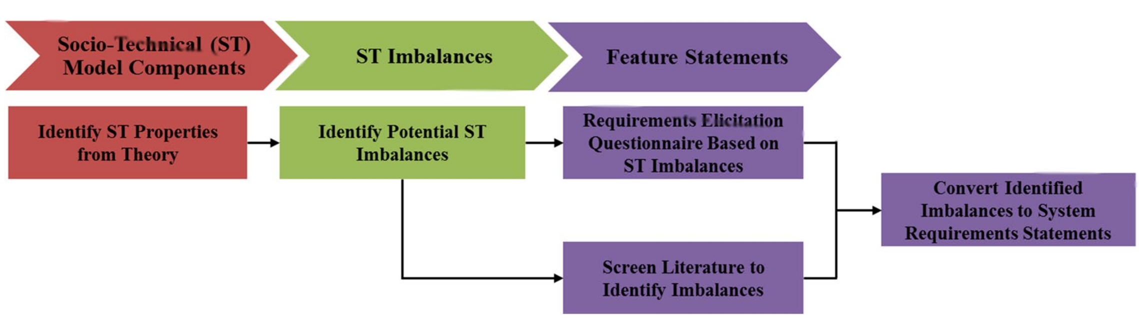

Fig. 3 Socio-Technical System Design Requirements Identification Process

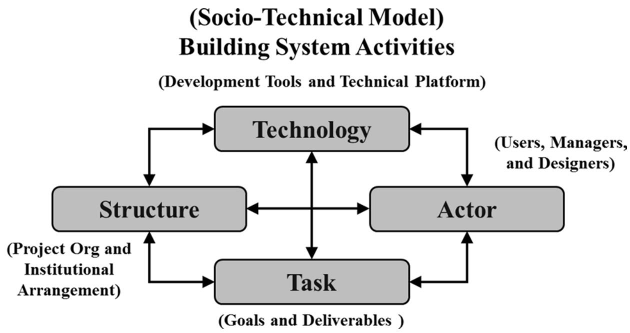

Fig. 4 Components of a ST System [73]

to provide a list of properties that define each of the ST model components. Therefore, a list of relevant properties for heath IT applications are identified for each component based on extensive literature review and grounded in relevant information system theories. These theories include technology acceptance model, unified theory of acceptance and use of technology, social learning theory, and diffusion of innovation theory. The relevant properties from each theory have been identified based on whether they are relevant to each of the ST component or not. The authors have iteratively compared and contrasted all the relevant properties based on the literature to identify the list of properties for each component.

The task component is defined using four properties namely importance to the goal, resources, difficulty, and interdependence. Importance to the goal encompasses whether tasks are performed in a job, if performed, how important they are [78]. Resources are defined in terms of task frequency, the cost of the task, or time required performing the tasks. Task difficulty encompasses the degree of "(non)-routineness", structuredness, and analyzability [79]. Finally, task interdependence is the degree to which a task is related to other tasks and the extent to which coordination with other organizational units is required [80].

The actor component is defined using four properties namely knowledge and expectation, self-efficacy, attitude, and subjective norms. Knowledge is defined as a "body of facts and principles that is learned through life experience, or is taught" [81], and outcome expectation is defined as a "person's estimate that a given behavior will lead to certain outcomes" [82]. Self-efficacy is defined as "people's perception of their ability to plan and take action to reach a particular goal" [83, 84]. Attitude is defined as an "affective or evaluative judgment of some person, object, or event" [85]. Finally, subjective norm is defined as "the person's perception that most people who are important to him think that he should or should not perform the behavior in questions" [86, 87].

The structure component is defined using four properties, namely communication processes, authority, workflow, and economics. Communication processes are defined in terms of systems of communication, as well as means and channels of communication [73]. Authority is classically defined as "the right to influence and direct behavior, such right having been accepted as valid and legitimate by others in the relationship" [88]. Workflow is defined as "automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules" [89]. Finally, economics is defined as the financial consideration associated with goods or services that are manufactured or produced.

The technology component is defined using three properties, namely functionality, usefulness, and ease of use. Technology functionality is the ability of technology to perform specific functions [90]. Ease of use is defined as "the degree to which a person believes that using a particular system would be free of effort" [91]. Finally, ease of use is defined as "the degree to which a person believes that using a particular system would enhance his or her job performance" [91].

5.2 Step 2: identify ST imbalances

The major imbalances between ST elements in the model are shown as gaps; these gaps are identified for the combinations of the four main ST components' properties. The *Task–Actor* gaps are related to actors' capabilities and other actor-related attributes that influence their ability to perform a task. The *Task–Structure* gaps arise when the identified structure is not aligned with the task or there is no adequate structure that is defined for a given task. *Task-Technology* gaps arise when technology is not adequate to support the tasks, or it is unreliable in its support. *Actor-Structure* gaps occur when actors do not know the operating procedures and do not accept the structure, whereas the *Actor-Technology* gaps occur when any of the identified actors do not understand, cannot operate, or do not accept the technology. Finally, in the *Structure-Technology* gaps, the identified structure is not aligned with the identified technology and does not support technology operations and use.

ST imbalances reflect the gaps that need to be addressed in the design of the new ST systems. As described earlier, ST imbalances are defined for the combination of properties of the ST model components. For example, in order to identify imbalances related to the task and actor components, we identify the combination of properties of the task and actor components. This results in a total of sixteen imbalances categories as shown in Table 2. Similar tables have been developed for the remaining combinations of

Table 2 Task–actor imbalances

Task–actor	Importance to the goal	Resources	Difficulty	Interdependence
Knowledge and expectations	Imbalances related to actors' knowledge and expectations, and importance of a task to achieve desired goal	Imbalances related to actors' knowledge and expectations, and frequency, cost, or time required performing the tasks	Imbalances related to actors' knowledge and expectations, and the degree of task's (non)-routineness, structuredness, and analyzability	Imbalances related to actors' knowledge and expectations, and the degree to which a task is related to other tasks and the extent to which coordination with other organizational units is required
Self-Efficacy	Imbalances related to actors' skills and self-efficacy, and perception about the importance of a task to achieve desired goal	Imbalances related to actors' skills and self-efficacy, and frequency, cost, or time required to perform the tasks	Imbalances related to actors' skills and self-efficacy, and the degree of task's (non)-routineness, structuredness, and analyzability	Imbalances related to actors' skills and self-efficacy, and the degree to which a task is related to other tasks and the extent to which coordination with other organizational units is required
Attitude	Imbalances related to actors mental or neural state of readiness, and the importance of a task to achieve desired goal	Imbalances related to actors mental or neural state of readiness, and task frequency, cost of the task, or time required performing the tasks	Imbalances related to actors mental or neural state of readiness, and the degree of task's (non)-routineness, structuredness, and analyzability	Imbalances related to actors mental or neural state of readiness, and the degree to which a task is related to other tasks and the extent to which coordination with other organizational units is required
Subjective norms	Imbalances related to actors' perception that most people who are important to the user think that the user should or should not perform the task, and the importance of a task to achieve desired goal	Imbalances related to actors' perception that most people who are important to user think that user should or should not perform the task, and task frequency, cost of the task, or time required performing the tasks	Imbalances related to actors' perception that most people who are important to user think that user should or should not perform the task, and the degree of task's (non)-routineness, structuredness, and analyzability	Imbalances related to actors' perception that most people who are important to user think that user should or should not perform the task, and the degree to which a task is related to other tasks and the extent to which coordination with other organizational units is required

Table 3 Task–Actor Imbalances Examples

Task–actor	Importance to the goal	Resources	Difficulty	Interdependence
Knowledge and expectations	Patients are unable to meet the expectations of health care providers and fulfilling self-care responsibilities important for maintaining patient's health [92]	Unrealistic patient expectations and demands can make evidence-based cost less effective and efficient [93]	Expectations for self-care autonomy exceeding the patients' cognitive and behavioral capabilities may compromise adherence and diabetic control [94]	Despite recent improvements in glucose control in adults with diabetes [2], < 15% of adults with diabetes simultaneously met the goal for three important components of care (i.e., glucose, blood pressure and low-density lipoproteins (LDL) cholesterol) [95]
Self-efficacy	Barriers to adherence and problem-solving skills can play a major role in affecting ongoing self-management of chronic disease [96]	Patients are not able to keep on top of needing different medication at different time—scheduling and coordination of medication [97]	Due to differences in technical skills, abilities and learning styles, patients find it difficult to perform specific tasks because they did not gain a comprehensive knowledge of how to perform these tasks [98]	Literacy, numeracy (numerical literacy), and health literacy are typically weak in these communities, and people may have a poor sense of autonomy and control over their environments and low self-efficacy for behavior change [99]
Attitude	Adoption and maintenance of health behaviors are often poorly predicted by behavioral intentions [100]	The lack of financial support for IT applications is a major barrier to adoption [101]	Negative attitude toward insulin therapy is associated with a general lack of understanding of the progressive nature of diabetes [102]	Negative patient attitude toward insulin may be due to a reluctance to add yet another medication to their daily regimen [102]
Subjective norms	Low family support prevent patients from performing the necessary self-management tasks [103]	Lower frequency of self-monitoring blood glucose (SMBG)is associated with the lack of family support that negatively affect adherence for SMBG [104]	Diabetes patients show that low support from their family was associated with making their diabetes more serious [105]	Results indicated that the nurse-facilitated social support group achieved higher levels of patient blood pressure control compared to a lecture group and control group which received usual or standard office care [106]

the ST model components but are not included. Example imbalances from the literature review (self-care domain) are shown in Table 3.

5.3 Step 3: develop questionnaire to investigate ST imbalances in problem context

ST problems can be solved by eliciting a set of requirements that can inform the design of a ST system that account for both technical as well as social dimensions of the system. The list of ST imbalances supported by the literature is converted to questionnaire statements for investigating the imbalances and identifying potential system features to address the imbalances. The final list of features is identified based on the list of ST imbalances as well as a comprehensive review of the literature. A sample example is shown in Table 4 that describes how the imbalances can be used to develop questionnaire statements in the context of an employee knowledge sharing system or a self-care health maintenance system. Hypothetical feature requirements that could address the imbalances based on the results of the interview are also included for illustrative purposes.

6 Evaluation

6.1 Assessment model

The design of the user interview's questions is considered a critical step in the process of requirements analysis. We hypothesize that the proposed questionnaire development process will enhance the quality of the requirements elicitation questionnaire. We also hypothesize that it will enhance the analysts' sense of preparedness in terms of their perception of interview readiness and domain knowledge.

The subjects in the user study are divided into two different groups, a treatment group and a control group. The treatment group is provided training on the proposed ST requirement elicitation process, whereas the control group is provided with the IEEE guidance on software requirement specification [107] to help with developing the requirement elicitation questionnaire. The validity of the concept of a "no-treatment" control group in software engineering research has been questioned [108]. A suggested alleviation is that the "no-treatment" group performs the technique/practice that they usually employ in a specific setting [109]. In this study, the subjects in the control group are graduate students who have taken courses on systems analysis, use their own knowledge and experience of requirements analysis as well as the IEEE guidance on software requirement specification [107].

Table 4 Designing questionnaire to elicit feature requirements and investigate imbalances

Imbalance	Example questionnaire statement	Example implementation
Imbalances related to actors' knowledge and expectations, and importance of a task to achieve desired goal	What do employees need to know about how knowledge sharing impacts productivity? What do users need to know about how self-care helps health maintenance?	Dashboard showing hours or dollars saved due to knowledge sharing "Did you know?" HealthTips module for providing knowledge
Imbalances related to actors' skills and self-efficacy, and perception about the importance of a task to achieve desired goal	How can an employee share knowledge? How can a user perform self-care?	FAQ and tips on knowledge sharing. Evidence Knowledge-base and e-learning videos on self-care and health maintenance
Imbalances related to actors mental or neural state of readiness, and the importance of a task to achieve desired goal	What motivates an employee to share knowledge? What motivates a user to perform self-care?	Employee recognition and badges for knowledge sharing Gaming analogy and point system for performing self-care
Imbalances related to actor's perception that most people who are important to the user think that the user should or should not perform the task, and the importance of a task to achieve desired goal	Who influences an employee's knowledge sharing behavior? Who influences a user to engage in self-care?	Metrics showing sharing behaviors of peers and colleagues Messaging functionality that can help share patients' information with family and care givers

6.2 Hypothesis development

Before we proceed with any tests, we need to make sure that the two groups are similar in terms of characteristics. So, the first hypothesis is used to check whether the two groups are equivalent before they are exposed to any treatment.

$H1_0$ There is no difference between the treatment group and the control group ($\mu t = \mu c$)

$H1_a$ There is a significant difference between the treatment group and the control group ($\mu t \neq \mu c$)

Adoption of the proposed process will distinguish the treatment group from the control group. The values representing the variable "access to the process" are 1 which denotes use of the proposed ST-based process by the treatment group and 0 which denotes control group.

The subjects' performance is modeled from two perspectives. The first perspective is to determine how the analysts feel about their understanding of the domain knowledge and how well they are prepared for the requirements analysis interviews (self-assessment perspective). In the self-assessment process, the analysts will evaluate and compare their own knowledge and ability after using the proposed ST-based requirement elicitation support system. We postulate that using the ST requirement elicitation process will increase the analyst's sense of preparedness in terms of their perception of domain knowledge and interview readiness for those who use the ST-based process.

$H2_0$ There is no difference between the self-reported domain knowledge of analysts in the treatment group and the control group ($\mu t = \mu c$)

$H2_a$ There is a significant difference between the self-reported domain knowledge of analysts in the treatment group and the control group ($\mu t > \mu c$)

$H3_0$ There is no difference between the self-reported interview readiness of analysts in the treatment group and the control group ($\mu t = \mu c$)

$H3_a$ There is a significant difference between the self-reported interview readiness of analysts in the treatment group and the control group ($\mu t > \mu c$)

The second perspective is the analysts' relative performance. In this perspective, the analyst performance when developing the interview questionnaire is compared for those who use the ST-based process (treatment group) and those who do not have access (control group). In order to assess analysts' performance, we assess the quality of the questionnaire based on metrics adopted from [9, 10].

Requirements elicitation involves asking the target users and stakeholders a set of questions as part of the interview session. A well-known questioning technique is the interrogatories technique, which involves asking "who," "what," "when," "where," "how," and "why" questions [9, 10, 110]. Such questions can help in better understanding the context and details of a system at different levels of abstraction [110].

- "What" questions request more information about a requirement. They are used to define the objectives and benefits from the system, system functionality, inputs and outputs of information, and a set of features which are considered logically related functional requirements that provide a capability to the user.
- *"How* to" questions ask how some activity, action or use case is to be performed. They are used to describe how the users want to perform goals or tasks, how the system must perform, the relationships between business data, the flow of data, and how the data are used to make decisions, and how the information and materials used. This category focuses on procedures and process. They are used to find out HOW does the WHO use the WHAT?
- *"Who"* questions are used to request confirmation about which stakeholders are responsible for a given action or requirements. They are used to identify the users of the product or system, or performers in the business process. The term user is defined as anyone who affects or is affected by the product.
- *"When"* questions when a process, activity, or feature should start. They are related to when users need to perform a task or achieve a goal, when information and materials are needed. Such questions are used to find out for each WHY, WHEN does the WHO need the WHAT? WHEN is often associated literally to what time of day, and also refer to the sequence of events, triggers, business cycles, as well as the transformation of states.
- *"Why"* questions are used to know more details about why we need a process, activity or feature. They are related to why users need to perform a task or achieve a goal, and why information and materials are needed. Such questions are used to find out WHY the WHO needs the WHAT?
- *"Where"* questions are used to know where the activity, action, or feature is used. They are also related to where be the information and materials used. This category incrementally builds upon the previous interests, for each WHO and WHY, WHERE is the WHAT used?

The questionnaires developed by the participants are evaluated along these six dimensions using the following two measures adapted from [9, 10]

- *Breadth* refers to the number of different questions categories along each dimension
- *Depth* refers to the number of questions obtained within each category

Before using the rubric, the domain expert must examine each question in order to determine whether it is relevant to the problem domain or not. In case a question is not relevant, the domain expert will delete that question from the questionnaire. The content of the questionnaire is analyzed based on the definitions of the "who," "what," "when," "where," "how," and "why" and codes will be assigned to each question based on these six labels. For each questionnaire, the breadth and depth are calculated. The mean values are calculated for the breadth and the depth/breadth values for all questionnaires. Such values will be used to perform a *t*-test between the two groups.

In this context, we hypothesize that using the proposed ST-based process will improve the overall performance of the analysts. The analyst' initial domain knowledge is used as a control variable to account for the variability of the domain knowledge at the start of the study.

$H4_0$ There is no difference between user interview questionnaire quality of analysts in the treatment group and the control group ($\mu t = \mu c$)

$H4.1_0$ There is no difference between user interview questionnaire breadth of analysts in the treatment group and the control group ($\mu t = \mu c$)

$H4.2_0$ There is no difference between user interview questionnaire depth/breadth of analysts in the treatment group and the control group ($\mu t = \mu c$)

$H4_a$ There is a significant difference between user interview questionnaire quality of analysts in the treatment group and the control group ($\mu t > \mu c$)

$H4.1_a$ There is a significant difference between user interview questionnaire breadth of analysts in the treatment group and the control group ($\mu t > \mu c$)

$H4.2_a$ There is a significant difference between user interview questionnaire depth/breadth of analysts in the treatment group and the control group ($\mu t > \mu c$)

6.3 Variable measurements

An instrument was developed based on existing literature [15] to measure analyst's perceived domain knowledge and self-reported interview readiness. A web-based survey instrument was used to collect data from subjects at different stages of the study. Semantic differential scales are used to measure each item in the survey for analyst's perceived domain knowledge and self-reported interview readiness. In the typical semantic differentiation task, "a subject judges a series of concepts against a series of bipolar, seven-step scales defined by verbal opposites". Examples of such verbal opposite can be good–bad, low–high, hot–cold, fair–unfair, etc. [111]. The instrument presented in the appendix is used to collect demographic information, analyst's domain knowledge, and analyst's interview readiness.

6.4 Experimental design and data collection

In design science research, experimental evaluations evaluate the design artifacts in terms of its utility [75, 112]. Also, it helps in empirically demonstrating the qualities of the artifact [75] and allows for generalization of the findings. Walls et al. [113] suggest an experimental design where the performance of the experimental group using the information technology (IT) artifact is compared against the performance of the control group not using the artifact. However, for rigorous purposes, the control groups will use a different IT artifact other than the one used by the treatment group. The experimental design using a treatment and control groups will ensure the rigor of the research in terms of the evaluations of the artifact.

We test the hypothesis empirically using a controlled experiment. A two-treatment pretest–posttest design is used to test the effectiveness of the proposed process. The purpose of the pretest is to make sure that all members of both controls and treatment groups have the same level of knowledge with respect to the main tasks of the experiment. Figure 5 shows the basic pre-/postrandomized experimental design. Each row represents a group of subjects, (R) denotes the random assignment of subjects to each group, outcomes (O) are measured before and after the treatment (X1) is assigned to the treatment group, and the other treatment (X2) is assigned to the control group.

The main task is to develop an interview questionnaire for a diabetes mobile application. The subjects are graduate students at a Mid-West University with systems analysis and design knowledge and course work. Students who have completed any

R	**O**	***X1***	**O**
R	**O**	***X2***	**O**

Fig. 5 The Basic Pre-/Postrandomized Experimental Design

classes related to system and design will be randomly selected to be part of the experiment. The subjects were assigned to the treatment and control groups, where the treatment groups are provided training on the proposed ST-based process, whereas the control group is provided with a non-ST-based process based on the IEEE software requirements specification (SRS) document. The SRS template has been modified to show the control group subjects a general definition for the function and non-functional requirements as well as example questions that they can follow to construct the questionnaire. The treatment group is provided with instructional material about the ST model and imbalances and how they can be used for requirements elicitation. Each of the participants is provided with a description (scenario) about the target mobile diabetes application. Based on the scenario, the participants develop the interview questions. At the time of registration, the data on participant's initial domain knowledge, interview readiness, and participant demographic information are collected. Once the interview questions are ready, participants will fill out the survey on perceived domain knowledge and perceived interview readiness. The quality of interview questionnaires is assessed by two independent raters.

6.5 Data analysis

Multivariate analysis of variance (MANOVA) is used to test the hypotheses. MANOVA is used to solve our Type I error rate problem by providing a single overall test of group differences [114, 115]. According to Joseph et al. [114], using MANOVA, the sample size requirements relate to individual group sizes and not the total sample per se. Joseph et al. [114] suggests that at minimum the sample in each cell (group) must be greater than the number of dependent variables. On the other hand, from a practical perspective, they recommended a minimum cell size of 20 observations. According to some experiments with G*Power [116, 117], a MANOVA with two levels and two dependent variables, using an alpha of 0.05, a power of 0.80, and a large effect size (f = 0.40) requires a sample size of 28.

In designing the study (Table 5), the research defines the following elements related to factors used, dependent variable, and the sample size

Table 5 Data analysis design

Independent variable	Dependent variable	
Two levels	Domain knowledge	Interview readiness
Treatment	X	X
Control	Y	Y

where $X + Y = N$, and N represent the sample size, & $X = Y$

- Factors: One factor is defined representing questionnaire development techniques followed, which is represented at two levels, access to the ST requirement elicitation process and no access to the ST requirement elicitation process (analyst uses SRS and his/her own experience)
- Dependent Variables: Evaluation is done for two variables (analyst domain knowledge and analyst interview readiness), measured on a 7-point semantic differential scale
- Sample: A minimum of 28 subjects are needed to participate in the experiment and rate the two dependent measures

7 Results and discussion

7.1 Test subjects

All subjects in both groups have completed a course in systems analysis and design (SAD) or software engineering (SE). For the treatment group, we have 6 out of 14 subjects with experience SAD and/or SE. Among the subjects, we have 2 subjects with age between 18 to 24, 11 subjects with age between 25 to 34, and finally 1 subject with age between 35 to 44. We have a total of 11 males and 3 females. For educational level, we have a total of 8 with master's degree and 6 with doctoral degrees. Finally, the treatment group are presented with a short quiz. We have a total of 14 subjects answered Q1 correctly, 8 subjects answered Q2 correctly, 9 subjects answered Q3 correctly, and 11 subjects answered Q4 correctly. On the other hand, for the control group we have 8 out of 14 subjects with experience in SAD and/or SE. We have 10 subjects with age between 25-34, 1 subject with age between 35 to 44, 2 subjects with age between 45 to 54, and finally, 1 subject with age between 55 to 64. We have a total of 14 males and no females. Finally, for educational level, we have a total of 6 with master's degree, 1 with a professional degree, and 7 with doctoral degrees.

7.2 MANOVA assumptions and descriptive statistics

MANOVA requires a valid set of assumptions, namely sample size and homogeneity of variance, outlier detection, multivariate linearity, multivariate normality, and multicollinearity. The principal consideration in the design of the two-group MANOVA is the sample size in each of the cells, which directly affects statistical power. Having equal cell sizes as described in Table 6 will make the statistical tests less sensitive to violations of the assumptions, especially the test for homogeneity of variance of the dependent variable. Each MANOVA assumption is validated using a specific test. Sample size is determined using Alpha (α) = 0.05, Power ($1 - \beta$) = 80, and Effect Size = 0.40 [116, 117]. For

Table 6 MANOVA assumptions tests for pretest and posttest

Assumption: *Test*	Pretest	Posttest
Sample size: *Alpha* (α) = 0.05, *power* (1 − β) = 80, *effect size* = 0.40	28	28
Outliers: *Mahalanobis distance* (< critical value of 13.82)	6.164	11.446
Multivariate covariance: *Box's M*	0.325*	0.158*
Multivariate Linearity: *Scatter plot using elliptical pattern*	General pattern with no square like plot, which matches the elliptical pattern criteria (Fig. 1)	General pattern with no square like plot, which matches the elliptical pattern criteria (Fig. 1)
Multivariate normality: *Shapiro-Wilks*	Domain knowledge 0.078** Interview readiness 0.128**	Domain knowledge 0.061** Interview readiness 0.001**
Multivariate collinearity: *Correlation matrix of dependent variables*	Domain knowledge 0.747 Interview readiness 0.90	Domain knowledge 0.834 Interview readiness = 0.90

*0.001, **0.05

outliers detection, the Mahalanobis Distance "the distance from the case to the centroid of all cases for the predictor variables", with a large distance indicates an observation that is an outlier in the space denned by the predictors [118]. According to the data description, and following [119, 120], the critical value for Mahalanobis distance is 13.82. Any value greater than 13.82 indicates the presence of outliers. The Box's M tests for equality of the covariance matrices [114]. Scatter plot can be used to test for linearity using the elliptical pattern, where linearity hold if and only if there is no deviation from an elliptical pattern [121] that goes from bottom left to top right [122]. For multivariate normality, no direct test is available. Therefore, "most researchers test for univariate normality of each variable" [114]. The two most common are the Shapiro–Wilks test and a modification of the Kolmogorov–Smirnov test. Each calculates the level of significance for the differences from a normal distribution [114, 123, 124]. For multivariate normality, we used the Shapiro-Wilks test. Finally, the simplest and most obvious means of identifying collinearity is an examination of the correlation matrix for the independent variables. The presence of high correlations (generally 0.90 and higher) is the indication of substantial collinearity [114]. Table 6 summarizes the results and findings from all tests.

According to Table 6, all the MANOVA assumptions are met for both pretest and posttest. The only assumption that has been violated is the test for multivariate normality of "interview readiness" in posttest. According to Table 6, the p-value for interview readiness is 0.001 < 0.05, which violates the null hypothesis that the data are normally

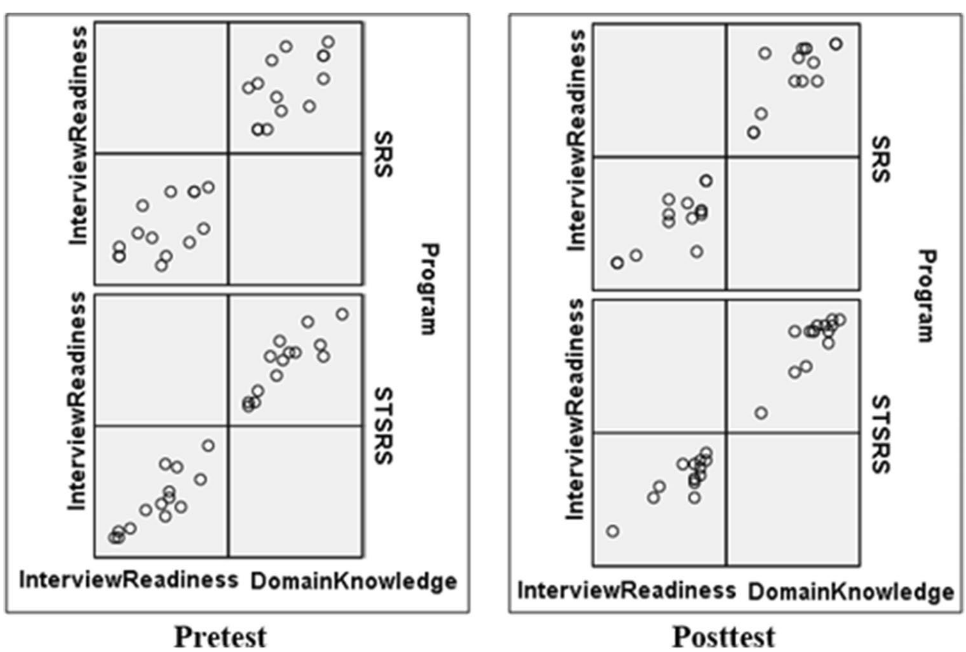

Fig. 6 Scatter plot

Table 7 Descriptive statistics

Program	Mean	SD	*N*
Domain knowledge			
SRS	2.63	1.240	14
STSRS	3.37	1.886	14
Total	3.00	1.611	28
Interview readiness			
SRS	3.57	1.708	14
STSRS	3.84	1.905	14
Total	3.71	1.781	28

Table 9 Multivariate tests

Effect		Value	*F*	Error df	Sig.
Intercept	Pillai's Trace	0.828	60.009b	25	0
	Wilks' Lambda	0.172	60.009b	25	0
Program	Pillai's Trace	0.077	1.047b	25	0.366
	Wilks' Lambda	0.923	1.047b	25	0.366

Table 10 Descriptive statistics

Program	Mean	SD	*N*
Domain knowledge			
SRS	3.671	1.4835	14
STSRS	5.629	1.0979	14
Total	4.650	1.6226	28
Interview readiness			
SRS	4.411	1.7058	14
STSRS	6.107	1.1211	14
Total	5.259	1.6590	28

distributed for interview readiness. However, "a violation of this assumption has minimal impact if the groups are of approximately equal size" (Largest group size/Smallest group size < 1.5) [114]. Figure 6 shows the scatter plot test for multivariate linearity. Scatter plot can be used to test for linearity using the elliptical pattern, where linearity hold if and only if there is no deviation from an elliptical pattern [121] that goes from bottom left to top right [122]. Both, protest and posttest scatter plots present a general pattern, with no square like plots, which match the elliptical pattern criteria, i.e., the data meet the assumption of linear relationship.

Since all the assumptions hold for the Pretest data, then we can proceed with MANOVA test. The objective of the first test is to make sure that the two groups are equivalent before they are exposed to any treatment. We have two groups, a treatment group (STSRS) and a control group (SRS). Each group has a total of 14 subjects. As shown in Table 7, we have comparable means for domain knowledge across the treatment group (3.37) and the control group (2.63). Also, we have almost similar means for interview readiness across the treatment group (3.57) and the control group (3.84).

Interpreting the tests of between-subjects' effects from Table 8, we have no statistically significant difference across the level of independent variable on each of the dependent variables, where the significance value for domain knowledge is (0.229) and the significant value for interview readiness is (0.699). Interpreting the Levene's test [114] of equality of error variances from Table 8, we have non-significant results for both domain knowledge and interview readiness, which means that we have no problems with the homogeneity of variance across outcome variable separately. The values for both dependent variables across different groups in Table 8 are not significant with (0.229) and (0.699) for domain knowledge and Interview readiness, respectively.

The four most commonly used multivariate tests are Pillai's criterion, Wi1ks' lambda, Hotelling's T2 and Roy's greatest characteristic root [114]. Table 9 shows the multivariate tests results. Based on the value of the Wi1ks' lambda as well as other tests, there is no statistically significant difference across the level of independent variable on the linear combination of the dependent variables. In our case, Wi1ks' lambda has a non-significance value of (0.366).

Based on the results above, we believe that the two groups are statistically sufficiently similar before they are exposed to any treatments. Since all the assumptions hold for the posttest data, then we can proceed with MANOVA test. For this test, we postulate that access to the ST-based process will increase the analyst's self-reported domain knowledge more than those who have not, which also will make the analyst feel better prepared for proceeding with users' interviews. As shown in Table 10, we have a slightly higher mean for domain knowledge across the treatment group (5.629) than the control group (3.671). Also, there is a slightly higher

Table 8 Test of between subject effects, contrast results, and Laven's test

Variables	B/W subject effect	Contrast results			Leven's test	
	Sig.	Sig.	95% CI		Sign.	*F*
			Lower bound	Upper bound		
Domain knowledge	0.001	0.229	− 1.983	0.497	0.174	1.958
Interview readiness	0.005	0.699	− 1.674	1.138	0.786	0.075

Table 11 Test of between subject effects, contrast results, and Levene's test

Variables	B/W subject effect	Contrast results			Leven's test	
	Sig.	Sig.	95% CI		Sign.	*F*
			Lower bound	Upper bound		
Domain knowledge	0.001	0.001	−2.971	−0.943	0.233	1.489
interview readiness	0.005	0.005	−2.818	−0.575	0.078	3.363

Table 12 Multivariate tests

Effect		Value	*F*	Error df	Sig.
Intercept	Pillai's Trace	0.940	195.156b	25	0
	Wilks' Lambda	0.060	195.156b	25	0
Program	Pillai's Trace	0.377	7.571b	25	0.003
	Wilks' Lambda	0.623	7.571b	25	0.003

Table 13 Number of questions per group

	Number of questions	
	All questions	Interrogatories questions
Treatment group	201	155
Control group	168	55

mean for interview readiness across the treatment group (6.107) than the control group (4.411).

Interpreting the tests of between-subjects' effects from Table 11, we have a statistically significant difference across the level of independent variable on each of the dependent variables, where the significance value for domain knowledge is (0.001) and the significant value for interview readiness is (0.005). This provides the necessary support for $H2_a$ and *H3a* that the two groups are statistically different after they are exposed to our treatment. However, based on these results, we cannot determine the extent to which the difference can be attributed to either domain knowledge or interview readiness. The values for both dependent variables across different groups in Table 11 are significant with (0.001) and (0.005) for domain knowledge and interview readiness, respectively, which both provide evidence to reject our null hypothesis $H2_0$ and $H3_0$ and support $H2_a$ and $H3_a$. Interpreting the Levene's test [114] of equality of error variances from Table 11, we have non-significant results for both domain knowledge and interview readiness, and no problems with the homogeneity of variance across outcome variable separately.

Table 12 shows the multivariate tests results. Based on the value of the Wi1ks' lambda as well as other tests, there is a statistically significant difference across the level of independent variable on the linear combination of the dependent variables. In our case, Wi1ks' lambda has a non-significance value of (0.003).

The results provide sufficed evidence to reject $H2_0$ and $H3_0$. So, $H2_a$ and $H3_a$ are supported and that the two groups are statistically different after they are exposed to the treatments, indicating that analysts who received training on the socio-technical requirements elicitation process report higher confidence in their domain knowledge and interview preparation.

Next, the analyst performance when developing the interview questionnaire was compared for those who have used the ST process (treatment group) and those who do not (control group). In order to assess analysts' performance, a judge (an expert in the domain of system analysis and design) assess the quality of the interviews questions. In this context, we postulate that access to ST requirement elicitation support system will improve the overall performance of the analysts, where the quality of the questionnaire developed by the analysts using the ST requirement elicitation support system will be better than those who have not. The analyst' initial domain knowledge is used as a control variable to account for the variability of the domain knowledge at the start of the study.

Before proceeding with hypothesis testing, we decided to remove yes/no questions that capture only one piece of information at a time, which is considered not effective way to collect requirements. Table 13 shows the number of questions per group with interrogatories questions and all questions.

These numbers obviously show that we have a problem when it comes to writing effective questions, especially for the control group, where the control group was able to write 55 interrogatories questions out of 168 questions, where most of the questions are yes/no questions that are mainly targeting one piece of information each time. Example questions written by the subjects include:

> "Do you wish to receive email or notification on the system daily"?

This question can be rewritten using the interrogatories questioning technique as follow:

Table 14 Means and standard deviations for breadth and depth/breadth—interrogatories questions

	Breadth		Depth/Breadth	
	Mean	SD	Mean	SD
Treatment group	2.64	1.15	5.08	3.54
Control group	1.14	1.10	1.99	2.82

Table 15 Independent samples *t*-test

	Breadth	Depth/breadth
Sig. (2-tailed)	0.002	0.017
Mean difference	1.5	3.09524
SE difference	0.42535	1.20995

"How do you want to receive diabetes related information using the mobile application"?

Another example:

"Do you wish the application to include dashboards and graphs to indicate for your glucose measurements"?

This question can be rewritten using the interrogatories questioning technique as follow:

"How do you want to display glucose measures overtime using the mobile application"?

For each questionnaire the breadth and depth are calculated. Then the mean is calculated for the breadth and the depth/breadth values for all questionnaires. The values are used to perform a t-test between the two groups. Results for these measures are shown in Table 14.

Independent samples test for breadth is shown in Table 15. Results from the analysis reveal a statistically significant value of 0.002, which means that there is a difference between the two groups with respect to the breadth of the questionnaire. This means that $H4.1_0$ is rejected, and our hypothesis $H4.1_a$ is supported. The independent samples test for depth/breadth is shown in Table 15. Results from the analysis reveal a statistically significant value of 0.017, which means that there is a difference between the two groups with respect to the depth/breadth of the questionnaire. This means that $H4.2_0$ is rejected, and our hypothesis $H4.2_a$ is supported. Overall, the test for both breadth and depth/breadth does support our assumption. In other words, the tests support our hypothesis H4 $_a$ and reject the null hypothesis $H4_0$, indicating that analysts who receive training on the socio-technical requirements elicitation process generate higher-quality questionnaires as measured by Breadth and Depth/Breadth of the questionnaire.

7.3 Threats to validity

The two-group pretest–posttest experimental design controls for all threats to validity. The design in Fig. 5 helps avoid the selection bias problem, or what is called selection threat, in which other factors other than the program lead to the posttest differences between the groups [125]. It is essential that the subjects assigned to treatment and control groups be representative of the same population [126]. In such case, random sampling and random assignment of subjects from a common population to one of the treatment and control groups [127] can help make sure that the two groups have similar characteristics and avoid selection threats to internal validity [125]. In this context, random sampling and random assignment are the key to make sure that any differences in the posttest results is related to the treatment and nothing else. A random sample is most likely to distribute any potential biasing characteristics across all the groups being formed through the sampling process [128].

Other threats to validity are also accounted for using the two-group pretest–posttest experimental design [129, 130], these threats include history threat, maturation threat, testing threat, instrumentation threat, regression threat, and mortality threats [129]. The history threat is controlled by the fact that the test for the treatment and control groups is completed at the same time and in the same setting. The maturation and testing threats are controlled the fact that they are manifested equally in both treatment and control groups. The validity and reliability of the data collection instrument has been justified in another study. In addition, we prepared instructions for using these instruments. Subjects were randomly assigned to the treatment and control groups, and they were not informed of the purpose of the experiment. Thus, we control for instrumentation threat. Regression threat is controlled by observing no difference between the treatment group and control group, and both groups regress similarly, regardless of treatment. The Selection threat is controlled using random sampling and random assignment of subjects from a common population to either the treatment group or the control groups. Mortality threat is controlled by the fact that the two groups remain similar through the experiment.

7.4 Limitations

As with most research, this study has limitations that can be noted. The complete potential of the proposed ST process has not been experienced by the subjects in the treatment group. Despite the fact the proposed ST process has proven its usefulness, the subjects were only exposed to a very limited information about the domain due to time constrains within the controlled experimental setting. Moreover, not all of the subjects involved in the study are practicing analysts who are involved in systems analysis and design processes.

Some of the students have the knowledge of systems analysis and design without practicing systems analysis and design as professionals. Accordingly, there is a need to explore the effect of the proposed ST process on systems analysts who are involved in the process of developing systems. Finally, despite the fact that gender differences appear to be a relevant factor in different research studies, we believe that the way the sample was selected and distributed among the groups as well as the pretest helped in avoiding such gender differences issues, where such techniques help control for other major explanatory factors.

8 Future research and conclusions

8.1 Future research directions

Several new extensions to the current study can be explored in future research. One area of study is the impact of gender and other demographic variables on the efficacy of the process. Another avenue for future research concerns the extent of training provided to analysts. As discussed in the limitations section, the subjects were provided a short training on socio-technical requirements elicitation process. In the future, we intend to develop curricular materials to complement existing training of systems analysis and requirements elicitation and explore the impact of comprehensive training on the quality of requirements elicitation outcomes. Moreover, future empirical studies can also be designed to include qualitative methods that can help analyze not only interview questionnaires but also interview notes and stakeholder responses, thus providing a deeper understanding of how socio-technical requirements can be captured, interpreted and analyzed.

8.2 Summary and contributions

In this paper, we presented and evaluated a new ST requirements elicitation process. Our approach for requirements identification consists of (1) defining four main ST components as well as a list of relevant attributes from literature, (2) defining imbalances among the combinations of the ST components' attributes, and (3) converting the list of identified imbalances to potential questionnaire statements, which represent system features (high-level requirements statements) that guide the interviews during requirement elicitation.

The proposed ST-based process was tested for its effectiveness in improving analysts' domain knowledge, readiness, and preparations for the requirements analysis phase using a two-group randomized experimental design. Our findings provide support when it comes to analysts' experience, where analysts' experience does not appear to be a relevant factor when using interviews as an elicitation technique. Our results showed those who are exposed to the ST process model appear to be more effective than those who have not regardless of the level of the experiences of the analysts. The proposed process also helps in improving analysts' self-assessment of domain knowledge and interview preparedness. Moreover, quality of the questionnaire was found to be significantly better for those analysts who are exposed to the ST-based process than those who have not.

Overall, the contribution of this work can be described along three dimensions: empirical, theoretical, and practical. A major empirical contribution of this work is to show how such process can affect analysts' understanding and learning. The process has been used to enhance analyst' self-reported domain knowledge and interview readiness. Results from hypothesis testing and data analysis showed that those who are exposed to the ST process reported enhanced domain knowledge as well as better interview readiness than those who have not. The theoretical contribution is the extension of socio-technical model to the requirements elicitation domain and a new process model for capturing the socio-technical aspects of a problem domain during the requirements elicitation process. In the ST System literature, there is a lack of a midrange theoretical model for emphasizing ST considerations in systems analysis and design. Previous studies we have are at a higher level of abstraction and do not provide any artifact, specific steps, or process that can be readily implemented in practice. The proposed socio-technical requirements elicitation process (STREP) model is an attempt at developing an artifact based on the notion of ST model of information systems that can be ported to practice. Finally, the practical contribution is a new way to improve analysts' domain knowledge and preparation for developing interview questionnaire and ultimately the quality of the requirements elicitation questionnaire.

References

1. Hickey AM, Davis AM (2004) A unified model of requirements elicitation. J Manag Inf Syst 20(4):65–84
2. Toro AD, Jiménez BB, Cortés AR, Bonilla MT (1999) A requirements elicitation approach based in templates and patterns. In: Proceedings 2nd workshop on requirements engineering, Buenos Aires, Argentina, September 9–10, pp 17–29
3. Raghavan S, Zelesnik G, Ford G (1994) Lecture notes on requirements elicitation. Educational Materials CMU/SEI–94–EM–10. Software Engineering Institute, Carnegie Mellon University
4. Hickey AM, Davis AM (2003) Requirements elicitation and elicitation technique selection: model for two knowledge-intensive software development processes. In: Proceedings of the 36th annual hawaii international conference on system sciences, Hawaii 2003. IEEE, p 10
5. Brooks (1987) No silver bullet essence and accidents of software engineering. Computer 20(4):10–19

6. Zowghi D, Coulin C (2005) Requirements elicitation: a survey of techniques, approaches, and tools. In: Engineering and managing software requirements. Springer, pp 19–46
7. Davis GB (1982) Strategies for information requirements determination. IBM Syst J 21(1):4–30
8. Byrd TA, Cossick KL, Zmud RW (1992) A synthesis of research on requirements analysis and knowledge acquisition techniques. Manag Inf Syst Q 16(1):117–138
9. Pitts MG, Browne GJ (2007) Improving requirements elicitation: an empirical investigation of procedural prompts. Inf Syst J 17(1):89–110
10. Browne G, Rogich M (2001) An empirical investigation of user requirements elicitation: comparing the effectiveness of prompting techniques. J Manag Inf Syst 17(4):223–249
11. Klier J, Klier M, Muschter S (2017) How to manage IS requirements in complex public sector structures: toward an action design research approach. Requir Eng 22(4):419–432
12. Mumford E (2000) A socio-technical approach to systems design. Requir Eng 5(2):125–133
13. Aurum A, Wohlin C (2005) Engineering and managing software requirements, 1st edn. Springer, Berlin
14. Sampaio do Prado Leite J, Gilvaz APP (1996) Requirements elicitation driven by interviews: the use of viewpoints. In: Proceedings of the 8th international workshop on software specification and design. IEEE, pp 85–94
15. Vitharana P, Jain H, Zahedi F (2012) A knowledge based component/service repository to enhance analysts' domain knowledge for requirements analysis. Inf Manag 49(1):24–35
16. Baloian N, Zurita G, Santoro FM, Araujo RM, Wolfgan S, Machado D, Pino JA (2011) A collaborative mobile approach for business process elicitation. In: 2011 15th international conference on computer supported cooperative work in design (CSCWD). IEEE, pp 473–480
17. Agarwal R, Tanniru M (1990) Knowledge acquisition using structured interviewing: an empirical investigation. J Manag Inf Syst 1:123–140
18. Vasundran M (2012) Comparison of requirements elicitation techniques. Int J Adv Comput Inf Technol
19. Lim SL, Finkelstein A (2012) StakeRare: using social networks and collaborative filtering for large-scale requirements elicitation. IEEE Trans Softw Eng 38(3):707–735
20. Moody JW, Blanton JE, Cheney PH (1998) A theoretically grounded approach to assist memory recall during information requirements determination. J Manag Inf Syst 15:79–98
21. Ferrari A, Spoletini P, Gnesi S (2016) Ambiguity and tacit knowledge in requirements elicitation interviews. Requir Eng 21(3):333–355
22. Davis A, Dieste O, Hickey A, Juristo N, Moreno AM (2006) Effectiveness of requirements elicitation techniques: empirical results derived from a systematic review. In: 14th IEEE international requirements engineering conference St. Paul, Minneapolis. IEEE, pp 179–188
23. Clegg C (2000) Sociotechnical principles for system design. Appl Ergonom 31:463–477. https://doi.org/10.1016/S0003-6870(00)00009-0
24. Fuentes-Fernández R, Gómez-Sanz JJ, Pavón J (2010) Understanding the human context in requirements elicitation. Requir Eng 15(3):267–283
25. Kaufmann A, Riehle D (2017) The QDAcity-RE method for structural domain modeling using qualitative data analysis. Requir Eng 24(1):1–18
26. Thew S, Sutcliffe A (2018) Value-based requirements engineering: method and experience. Requir Eng 23(4):443–464
27. Levy M, Hadar I, Aviv I (2018) A requirements engineering methodology for knowledge management solutions: integrating technical and social aspects. Requir Eng 23(1):1–19
28. Weinert B, Hahn A, Uslar M (2018) Domain-specific requirements elicitation for socio-technical system-of-systems. In: Paper presented at the 2018 13th annual conference on system of systems engineering (SoSE), Paris, France
29. Wang Y, Zhao L (2017) Eliciting user requirements for e-collaboration systems: a proposal for a multi-perspective modeling approach. Requir Eng 254(2):205–229
30. Shahri A, Hosseini M, Taylor J, Stefanidis A, Phalp K, Ali R (2019) Engineering digital motivation in businesses: a modelling and analysis framework. Requir Eng 24(1):1–32
31. Ackerman M, Prilla M, Stary C, Herrmann T, Goggins S (2017) Designing healthcare that works: a sociotechnical approach, 1st edn. Academic Press, Orlando
32. Li T, Horkoff J, Mylopoulos J (2018) Holistic security requirements analysis for socio-technical systems. Softw Syst Model 17(4):1253–1285
33. Eason K (2008) Sociotechnical systems theory in the 21st Century: another half-filled glass. In: Desmond M, John K (eds) Sense in social science—a collection of essays in Honour of Dr. Lisl Klein, Desmond Graves, pp 123–134
34. Vink P (2012) Advances in social and organizational factors, 1st edn. Taylor & Francis Group, London
35. Baxter G, Sommerville I (2011) Socio-technical systems: from design methods to systems engineering. Interact Comput 23(1):4–17
36. Mumford E (2006) The story of socio-technical design: reflections on its successes, failures and potential. Inf Syst J 16(4):317–342
37. Emery F, Trist E (1960) Socio-technical systems. In: West C, Michael V (eds) Management sciences models and techniques, vol 1. Pergamon Press, New York, pp 38–58
38. Bryl V, Giorgini P, Mylopoulos J (2009) Designing socio-technical systems: from stakeholder goals to social networks. Requir Eng 14(1):47–70
39. Cherns A (1987) Principles of sociotechnical design revisted. Hum Relat 40(3):153–161
40. Cherns A (1976) The principles of sociotechnical design. Hum Relat 29(8):783–792
41. Berg M, Toussaint P (2003) The mantra of modeling and the forgotten powers of paper: a sociotechnical view on the development of process-oriented ICT in health care. Int J Med Inf 69(2):223–234
42. Eason K (2007) Local sociotechnical system development in the NHS National Programme for Information Technology. J Inf Technol 22(3):257–264
43. Zielczynski P (2007) Requirements management using IBM rational requisitepro. Pearson Education & IBM Press, Indianapolis
44. Leffingwell D, Widrig D (2000) Managing software requirements: a unified approach, 1st edn. Addison-Wesley Professional, Indianapolis, IN, USA
45. Turner CR, Fuggetta A, Lavazza L, Wolf AL (1999) A conceptual basis for feature engineering. J Syst Softw 49(1):3–15
46. Turner CR, Wolf AL, Fuggetta A, Lavazza L (1998) Feature engineering. In: Proceedings of the 9th international workshop on software specification and design, Mie, Japan. IEEE Computer Society, p 162
47. Tsung SMM-L (2004) ERP: a route toward successful implementation. University of Leeds, University of Leeds Press, Leeds
48. Gross D, Yu E (2001) From non-functional requirements to design through patterns. Requir Eng 6(1):18–36
49. Lee Y, Zhao W (2006) Domain requirements elicitation and analysis-an ontology-based approach. In: Paper presented at the international conference on computational science, Berlin, Heidelberg

50. Nuseibeh B, Easterbrook S (2000) Requirements engineering: a roadmap. In: Proceedings of the conference on the future of software engineering, Limerick, Ireland. ACM, pp 35–46
51. Liu L, Lin L (2008) ARED-CK: an automated requirements elicitation approach based on decision-making with complete knowledge. In: First international workshop on managing requirements knowledge, Barcelona, Spain. IEEE, pp 47–52
52. Kenzi K, Soffer P, Hadar I (2010) The role of domain knowledge in requirements elicitation: an exploratory study. In: Mediterranean conference on information systems Netherlands, vol 2. Association for Information Systems, p 48
53. Hofmann HF, Lehner F (2001) Requirements engineering as a success factor in software projects. IEEE Softw 18(4):58–66
54. Hadar I, Soffer P, Kenzi K (2014) The role of domain knowledge in requirements elicitation via interviews: an exploratory study. Requir Eng 19(2):143–159
55. Sabahat N, Iqbal F, Azam F, Javed MY (2010) An iterative approach for global requirements elicitation: a case study analysis. In: International conference on electronics and information engineering, 2010. IEEE, pp V1-361–V361-366
56. Sutcliffe A, Sawyer P (2013) Requirements elicitation: Towards the unknown unknowns. In: Paper presented at the 21st IEEE international requirements engineering conference, Rio de Janeiro, Brazil
57. Chua BB, Bernardo DV, Verner J (2010) Understanding the use of elicitation approaches for effective requirements gathering. In: 2010 Fifth international conference on software engineering advances (ICSEA). IEEE, pp 325–330
58. Alistair S, Shailey M (1999) Analysing socio-technical system requirements. In: 4th International symposium on requirements engineering, Limerick, Ireland
59. Mavin A, Maiden N (2003) Determining socio-technical systems requirements: experiences with generating and walking through scenarios. In: Proceedings of the 11th IEEE international conference on requirements engineering, Monterey Bay, CA. IEEE, pp 213–222
60. Sommerville I, Sawyer P (1997) Requirements engineering: a good practice guide, 1st edn. Wiley, New York
61. Holbrook H III (1990) A scenario-based methodology for conducting requirements elicitation. ACM Sigsoft Softw Eng Notes 15(1):95–104
62. Asteasuain F, Braberman V (2017) Declaratively building behavior by means of scenario clauses. Requir Eng 22(2):239–274
63. Atladottir G, Hvannberg ET, Gunnarsdottir S (2012) Comparing task practicing and prototype fidelities when applying scenario acting to elicit requirements. Requir Eng 17(3):157–170
64. Sutcliffe, Ryan (1998) Experience with SCRAM, a scenario requirements analysis method. In: Paper presented at the proceedings of the third international conference on requirements engineering, Colorado Springs, CO
65. Sommerville I, Sawyer P, Viller S (1998) Viewpoints for requirements elicitation: a practical approach. In: Proceedings of the third international conference on requirements engineering, Colorado Springs, CO. IEEE, pp 74–81
66. Fuller RM, Davis CJ (2008) The Overreliance on analyst experience in the selection of requirements elicitation techniques. In: Paper presented at the proceedings of the 14th Americas conference on information systems (AMCIS 2008). Toronto, Ontario, Canada
67. Dalpiaz F, Giorgini P, Mylopoulos J (2013) Adaptive socio-technical systems: a requirements-based approach. Requir Eng 18(1):1–24
68. Herrmann T (2009) Systems design with the socio-technical walkthrough. In: Whitworth B, de Moor A (eds) Handbook of research on socio-technical design and social networking systems. IGI Global, Pennsylvania, pp 336–351
69. Jones S, Maiden NA (2005) RESCUE: An integrated method for specifying requirements for complex socio-technical systems. In: Jose LM, Andres S (eds) Requirements engineering for socio-technical systems. IGI Global, Pennsylvania, pp 245–265
70. Katina PF, Keating CB, Raed MJ (2014) System requirements engineering in complex situations. Requir Eng 19(1):45–62
71. Hubbard R, Schroeder CN, Mead NR (2000) An assessment of the relative efficiency of a facilitator-driven requirements collection process with respect to the conventional interview method. In: Proceedings of the 4th international conference on requirements engineering, Schaumburg, IL. IEEE, pp 178–186
72. Clegg C (2000) Sociotechnical principles for system design. Appl Ergonom 31(5):463–477
73. Lyytinen K, Newman M (2008) Explaining information systems change: a punctuated socio-technical change model. Eur J Inf Syst 17(6):589–613
74. Horkoff J, Aydemir FB, Cardoso E, Li T, Maté A, Paja E, Salnitri M, Piras L, Mylopoulos J, Giorgini P (2017) Goal-oriented requirements engineering: an extended systematic mapping study. Requir Eng 24(2):1–28
75. Hevner March, Park Ram (2004) Design science in information systems research. Manag Inf Syst Q 28(1):75–105
76. vom Brocke J, Riedl R, Léger P-M (2013) Application strategies for neuroscience in information systems design science research. J Comput Inf Syst 53(3):1–13
77. Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S (2007) A design science research methodology for information systems research. J Manag Inf Syst 24(3):45–77
78. Hogan J, Hogan R, Busch CM (1984) How to measure service orientation. J Appl Psychol 69(1):167
79. Gebauer J, Shaw MJ, Gribbins ML (2005) Towards a specific theory of task-technology fit for mobile information systems. University of Illinois at Urbana-Champaign, College of Business Working Paper
80. Thompson J (2017) Organizations in action: social science bases of administrative theory. Routledge, New York
81. Fredericks S, Guruge S, Sidani S, Wan T (2010) Postoperative patient education: a systematic review. Clin Nurs Res 19(2):144–164
82. Bandura A, McClelland DC (1977) Social learning theory. General Learning Press, New York
83. Bandura A (1994) Self-efficacy. In: Corsini R (ed) Encyclopedia of psychology, vol 3, 2nd edn. Wiley, New York
84. Bandura A (1977) Social learning theory, 1st edn. Prentice Hall, Englewood Cliffs
85. Barki H, Hartwick J (1994) Measuring user participation, user involvement, and user attitude. Manag Inf Syst Q 18(1):59–82
86. Fishbein M, Ajzen I (1975) Belief, attitude, intention and behavior: An introduction to theory and research. Philos Rhetor 10(2):130–132
87. Venkatesh V, Morris MG, Davis GB, Davis FD (2003) User acceptance of information technology: toward a unified view. Manag Inf Syst Q 27(3):425–478
88. Haug MR, Lavin B (1981) Practitioner or patient-Who's in charge? J Health Soc Behav 22(3):212–229
89. Sadiq SW, Marjanovic O, Orlowska ME (2000) Managing change and time in dynamic workflow processes. Int J Coop Inf Syst 9(01n02):93–116
90. Galloway LJ (2006) A scheme of technology acceptance for mobile computing. In: Proceedings of the 2006 information resources management association international conference, Washington, DC. IGI Global, pp 21–24
91. Davis F (1989) Perceived usefulness, perceived ease of use, and user acceptance of information technology. Manag Inf Syst Q 13(3):319–340

92. Harris Wysocki, Sadler Wilkinson, Harvey Buckloh, Mauras White (2000) Validation of a structured interview for the assessment of diabetes self-management. Diabetes Care 23(9):1301–1304
93. Wagner EH, Austin BT, Von Korff M (1996) Organizing care for patients with chronic illness. Milbank Q 74(4):511–544
94. Wysocki T, Taylor A, Hough BS, Linscheid TR, Yeates KO, Naglieri JA (1996) Deviation from developmentally appropriate self-care autonomy: association with diabetes outcomes. Diabetes Care 19(2):119–125
95. Nam S, Chesla C, Stotts NA, Kroon L, Janson SL (2011) Barriers to diabetes management: patient and provider factors. Diabetes Res Clin Pract 93(1):1–9
96. Toobert DJ, Strycker LA, Glasgow RE, Barrera M, Bagdade JD (2002) Enhancing support for health behavior change among women at risk for heart disease: the Mediterranean Lifestyle Trial. Health Educ Res 17(5):574–585
97. Bayliss EA, Steiner JF, Fernald DH, Crane LA, Main DS (2003) Descriptions of barriers to self-care by persons with comorbid chronic diseases. Ann Family Med 1(1):15–21
98. Tobin S et al (2012) Lesson learned: effective training strategies for electronic data capturing. Society for Clinical Trials, Miami Annual Meeting
99. Ershow AG (2009) Environmental influences on development of type 2 diabetes and obesity: challenges in personalizing prevention and management. J Diabetes Sci Technol 3(4):727–734
100. Schwarzer R, Schüz B, Ziegelmann JP, Lippke S, Luszczynska A, Scholz U (2007) Adoption and maintenance of four health behaviors: Theory-guided longitudinal studies on dental flossing, seat belt use, dietary behavior, and physical activity. Ann Behav Med 33(2):156–166
101. Anderson J (2007) Social, ethical and legal barriers to e-health. Int J Med Inf 76(5):480–483
102. Marrero DG (2007) Overcoming patient barriers to initiating insulin therapy in type 2 diabetes mellitus. Clin Cornerstone 8(2):33–43
103. Richard AA, Shea K (2011) Delineation of self-care and associated concepts. J Nurs Scholarsh 43(3):255–264
104. Fisher WA (2007) Barriers and behaviours in blood glucose monitoring. US Endocrine Dis 2(1):51–53
105. Skinner TC, John M, Hampson SE (2000) Social support and personal models of diabetes as predictors of self-care and well-being: a longitudinal study of adolescents with diabetes. J Pediatr Psychol 25(4):257–267
106. Morisky DE, DeMuth NM, Field-Fass M, Green LW, Levine DM (1985) Evaluation of family health education to build social support for long-term control of high blood pressure. Health Educ Behav 12(1):35–50
107. IEEE (1984) ANSI/IEEE Standard 830-1984: IEEE guide for software requirements specifications. IEEE STD, pp 1–26
108. Kitchenham BA, Linkman SG, Law D (1994) Critical review of quantitative assessment. Softw Eng J 9(2):43–54
109. Juristo N, Moreno AM (2003) Lecture notes on empirical software engineering, vol 12, 1st edn. World Scientific, River Edge
110. Gaska MT, Gause DC (1998) An approach for cross-discipline requirements engineering process patterns. In: Proceedings of IEEE international symposium on requirements engineering, Colorado Springs, CO. IEEE, pp 182–189
111. Osgood CE (1964) Semantic differential technique in the comparative study of Cultures[1]. Am Anthropol 66(3):171–200
112. D'Aubeterre F, Iyer LS, Singh R (2009) An empirical evaluation of information security awareness levels in designing secure business processes. In: Proceedings of the 4th international conference on design science research in information systems and technology, Philadelphia, Pennsylvania. ACM, p 16
113. Walls JG, Widmeyer GR, El Sawy OA (1992) Building an information system design theory for vigilant EIS. Inf Syst Res 3(1):36–59
114. Joseph H, William B, Barry B, Rolph A (2010) Multivariate data analysis, 7th edn. Pearson, Upper Saddle River
115. Karimi J, Somers TM, Gupta YP (2001) Impact of information technology management practices on customer service. J Manag Inf Syst 17(4):125–158
116. Faul F, Erdfelder E, Buchner A, Lang A (2013) G*Power Version 3.1.7 [computer software]
117. Statistics_Solutions (2013) Sample Size Write-up Statistics Solutions. http://www.statisticssolutions.com/resources/sample-size-calculator/manova-groups/manova-2-levels-and-2-dependent-variables
118. Stevens JP (1984) Outliers and influential data points in regression analysis. Psychol Bull 95(2):334
119. Pallant J (2013) SPSS survival manual: a step-by-step guide to data analysis using SPSS, 5th edn. Open University Press, Philadelphia
120. Pearson ES, Pearson K (1954) Biometrika tables for statisticians, vol 2, 2nd edn. Cambridge University Press, Cambridge
121. Kenneth A, Richard C, Ingram BEO (1996) Education in a research University, 1st edn. Stanford University Press, Palo Alto
122. Arthur W (2002) The interaction between developmental bias and natural selection: from centipede segments to a general hypothesis. Heredity 89(4):239–246
123. Dufour JM, Farhat A, Gardiol L, Khalaf L (1998) Simulation-based finite sample normality tests in linear regressions. Econom J 1(1):154–173
124. Mendes M, Pala A (2003) Type I error rate and power of three normality tests. Pak J Inf Technol 2(2):135–139
125. Trochim WM, Donnelly JP (2006) Research methods knowledge base, 3rd edn. Atomic Dog Publishing
126. Everitt BS (2002) The Cambridge dictionary of statistics. Cambridge University Press, Cambridge
127. Montgomery DC (2012) Design and analysis of experiments. Wiley, Hoboken
128. Neil S (2012) 100 questions (and answers) about research methods. SAGE Publications, Washington, DC
129. Campbell DT, Stanley JC (2015) Experimental and quasi-experimental designs for research. Ravenio Books, New York **Online (e-textbook)**
130. Thomas C (1976) Quasi-experimentation: design and analysis issues for field settings, 1st edn. Houghton Mifflin, Chicago