Soft Systems in Requirements Engineering: A Case Study

Alejandra Yepez Lopez  
Computer Science and Engineering  
Mississippi State University  
Mississippi State, MS 39762  
ay68@msstate.edu

Nan Niu  
Computer Science and Engineering  
Mississippi State University  
Mississippi State, MS 39762  
niu@cse.msstate.edu

Abstract

Soft systems methodology should offer substantial benefits in managing expectations and requirements for a software-intensive system, but the benefits have not yet been examined empirically. This paper reports an exploratory case study investigating the hypothesis that “soft systems approach would identify all the flaws in requirements practices and suggest improvements suited to an organization’s context”. We analyzed problematic requirements practices in an ongoing software project, modeled potential changes, and asked the project team to assess the organizational fit of these changes. We conclude that soft systems methodology could indeed uncover a relatively complete set of flaws in requirements engineering, but not all changes were necessary.

1. Introduction

Software requirements engineering (RE) is a “soft” human-centered activity concerned with identifying and communicating the purpose of a software-intensive system, and the context in which it will be used [6]. Soft systems methodology (SSM) aims to deal with “fuzzy” situations like RE, where multiple stakeholders’ diverse objectives exist [2]. In this context, requirements engineers must cooperate with the users in understanding the problem situation; however, users have yet to be fully empowered in deciding process and requirements and engineers are seen to deliver what is thought of as the users’ requirements with minimal reference to the users [1]. As a result, a large number of software projects failed. For instance, [4] estimated system failure was between 25% and 90% due to the following generic issues:

- **Correspondence failure** – the delivered system does not correspond to what was required.
- **Process failure** – a system is not forthcoming within time or resource constraints.
- **Interaction failure** – systems, as implemented, which fail to satisfy the users.
- **Expectation failure** – systems that are unable to meet stakeholders’ expectations.

RE addresses these issues more economically only if the problems can be identified. Thus, a key premise of applying SSM in RE is that holistic thinking would provide improvements to all problem issues. However, we are aware of no empirical studies that investigate this basic tenet, nor the scope of its applicability. To shorten this gap, we conducted an evaluation of applying SSM to an organization’s RE practices. Our goal was to explore what differences the use of SSM made to RE.

2. RE and SSM

The RE process is inherently “soft” since the needs that drive requirements are embedded in the relevant social, cultural, and organizational contexts [6]. Checkland’s SSM [2] favors human views of the problem, and has been used to study system characterized by the activities where the human behavior is a key factor that determines the result (satisfactory or not) of these activities, such as requirements elicitation and validation.

At the heart of soft systems thinking is the principle that whole entities exhibit emergent properties which are meaningful only when attributed to the whole, not to its parts [5]. Such a holistic view is particularly useful for uncovering flaws in current RE practices and enabling resolutions in an organized manner.

SSM involves users in software design [1], and [5] demonstrates the usefulness of applying SSM in the technical domain of designing software applications. In contrast, we explore the possibility of applying SSM in the soft domain of RE that identifies and communicates the purpose and the context of a software-intensive system.
3. Study context

The subject in our study is a research organization that specializes in scientific computing and government service. It is located in Starkville, Mississippi, and has approximately 50 employees as of 2009. To honor confidentiality agreements, we will call it SrvU. SrvU’s mission is to plan and develop software to assist scientific communities and government agencies.

SrvU has a simple organizational structure where the project coordinator and the head manage the projects. The research scientists gather and analyze the information of the projects. The database administrator (DBA) manages the technological infrastructure and acts as an intermediary between the clients and SrvU’s project team. The programmers develop the software system required for the project.

The case (unit of analysis) in our study is a traffic and transportation project, whose goal is to provide software support for analyzing and reporting daily activities the customer collects. Key project features include information visualization and predictive road traffic modeling. The project experienced failures in the past, mainly because the delivered product had not satisfied the user requirements. For this reason, the software has been redone several times and is currently within a new development cycle.

4. Methodology

We used an exploratory case study [8] as the basis for our research design. In our case, we were particularly interested in understanding how the use of SSM would affect the RE process.

We derived a central hypothesis to guide our study design: “SSM would find all the RE flaws and suggest improvements suitable for SrvU’s context.” To investigate this, we adapted Checkland’s conventional model [2] to a more appropriate 4-stage SSM, as suggested in [5]. Figure 1 shows the adapted SSM stages.

Stage 1 examined SrvU’s current RE practices. Stage 2 formulated root definitions via the mnemonic CATWOE, one of the best known SSM tools, used to define the Customers, Actors, Transformation, Weltanschauung (the worldview), Ownership, and Environmental constraints [2]. Stage 3 built conceptual models and identified potential organizational changes. Stage 4 allowed the stakeholders to debate the consequences of the proposed changes, so that the culturally feasible improvements to SrvU’s RE practices could be applied.

![Figure 1. Adapted SSM stages](image)

We tested our central hypothesis primarily via two evaluations, both of which were intentionally carried out in the “real world” (see Figure 1).

- **Objective assessment.** We used the four generic failure types [4] (correspondence, process, interaction, and expectation) as a baseline to check whether SSM’s holistic thinking would identify a relatively complete set of RE flaws. This was done at stage 1 of Figure 1.

- **Subjective assessment.** We used questionnaires and semi-structured interviews to check whether SSM’s conceptual modeling would generate RE improvements suitable for SrvU. This was done at stage 4 of Figure 1.

In our study, the data was collected mainly by means of interviews and questionnaires which involved project coordinator, DBA, and programmer. Note that we took the actual project duration into account when collecting data because some experiences and answers were only obvious in hindsight. In our evaluation, we used qualitative methods [7] to analyze the collected data. Qualitative research seeks to make sense of the way themes and meanings emerged and patterned in the data records built up from interviews and questionnaires. It is particularly suitable in our study since our collected data consisted of records of observation and interaction that were complex and contextualized.

5. Results

5.1. Understanding current RE practices

One feature of the project under study is that an intermediary user representative participated in requirements meetings with SrvU’s project team. The lack of communication with end users posed the risk that the delivered system was unable to meet their expectations [4]. SrvU’s DBA occasionally had to guess the users’ requirements, which would cause correspondence and interaction failures [4]. Every team member was mandated to attend the weekly meeting, coordinated by upper level personnel. This could result in software engineers’ passive participation in the project, thereby causing a loss of...
motivations to some extent [3]. The lack of requirements analyst’s skill set could contribute to the process failure [4].

As a result, the holistic thinking indeed uncovered a complete set of RE flaws, according to the baseline failure types (correspondence, process, interaction, and expectation) [4] employed in our study.

5.2. Conceptual modeling in soft systems world

We leveraged the CATWOE mnemonic [2] to formulate the SSM’s root definition, a precise description of the emergent properties of a system [5].

- Customers (C): users of the software project.
- Actors (A): SrvU’s project members.
- Transformation (T): producing a software system fulfilling the users’ needs.
- Weltanschauung (W): project members whose goals are to elicit, analyze, and validate requirements so that the high-quality software can be constructed and delivered.
- Ownership (O): SrvU.
- Environmental constraints (E): restrictions and standards applied to the use of information and technology framework used.

Teasing out CATWOE definitions provided a firm conceptual basis for modeling potential changes that SrvU might apply to improve its RE practices. We grouped the changes into four areas:

- **Communication.** A more direct communication with the users should adjust the expectations more properly, and greatly reduce the DBA’s guessing of what the users really want.
- **Elicitation.** The requirements should be gathered using multiple elicitation techniques [1], and reviewed by all stakeholders.
- **Role.** An explicit “requirements analyst” role should be created to better coordinate project meetings and to possibly replace the intermediary user representative.
- **Documentation.** Both SrvU and customers should agree on and maintain a formal document for contractual purposes.

The above areas for improvements were modeled by the first author of this paper. It is our belief that these changes stemmed from the soft systems world would overcome SrvU’s RE flaws in a culturally feasible way. Figure 2 shows how the RE areas could address different types of software failures [4].

### Table 1. Summary of subjective evaluation

<table>
<thead>
<tr>
<th>RE Area</th>
<th>Question</th>
<th>Answer</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>Current time interval comm. with the user</td>
<td>All responded “more than twice a week”</td>
<td>Positive</td>
</tr>
<tr>
<td></td>
<td>Ideal time interval communication</td>
<td>One said “twice a week” &amp; others “as needed”</td>
<td></td>
</tr>
<tr>
<td>Elicitation</td>
<td>Current elicitation techniques applied to the project</td>
<td>“meetings” and “personal knowledge and experience”</td>
<td>Positive</td>
</tr>
<tr>
<td></td>
<td>Preference of other elicitation techniques</td>
<td>Meetings, modeling, prototyping, etc.</td>
<td></td>
</tr>
<tr>
<td>Role</td>
<td>Having a requirement analyst role</td>
<td>It does not seem to be necessary</td>
<td>Negative</td>
</tr>
<tr>
<td></td>
<td>Categorization of the role of the current intermediary user</td>
<td>Two responded “very effective” &amp; one “somewhat effective”</td>
<td></td>
</tr>
<tr>
<td>Documentation</td>
<td>Presence of requirements documentation</td>
<td>“(Yes,) under contract” &amp; “(Yes) in the initial (project) proposal”</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Is it useful?</td>
<td>All responded “Yes”</td>
<td>Positive</td>
</tr>
<tr>
<td></td>
<td>Opinion about the best way to document</td>
<td>Lightweight formats, e.g., e-mails, feature list</td>
<td></td>
</tr>
</tbody>
</table>

Although common elicitation techniques were meetings and personal knowledge, the project team realized the importance to incorporate other techniques like prototyping and modeling. Proper documentation was
crucial not only for contractual purposes, but also for “keeping everyone on the same page.” In this regard, lightweight formats like feature list were preferred.

To our surprise, the idea of setting up the “requirements analyst” role was not appealing to SrvU. According to some project members, SrvU enjoyed the culture of small project teams, supported by their flat organizational structure. They regarded the current division of labor, especially with the intermediary user representative, was effective. They were concerned that adding another role would increase latency and reduce the throughput of SrvU’s business processes.

Our work has provided SrvU much insight into their RE practices. Although we cannot claim that our results have had direct impacts on all SrvU’s RE processes, we can claim to have initiated changes in some areas. Already, SrvU has introduced rapid prototyping to the project’s current iteration, and also started exploring more flexible ways to document system features.

6. Threats to validity

Several factors can affect the validity of our exploratory case study. Construct validity concerns establishing correct operational measures for the concepts being studied [8]. The main constructs in our case study are “SSM”, “all the RE flaws” and “improvements suitable for SrvU’s context”. As for the first construct, our 4-stage design, shown in Figure 1, was in line with many SSM studies, e.g., [5]. As for the second construct, we do not feel that using the generic software failure types [4] as the baseline posed a serious limitation. As for the third construct, due to the lack of metrics for organizational fit, our best measure came from the subjective opinions of SrvU’s project members.

Regarding internal validity [8], a major limitation of our study design is the SSM modeling skills and experiences of the researcher, which compounds the problem of experimenter bias [8]. We plan to address the threat by involving multiple experts and stakeholders in developing conceptual models. Another likely confounding variable is the interview data. Participants in our current study may have omitted important facts when answering questions or we may have misinterpreted the data. This threat was mitigated by on-site ethnographic observation, as well as applying pre-defined qualitative data analysis methods (coding and categorizing) jointly by the two authors of this paper.

The results of this study might not generalize beyond SrvU’s organizational conditions and its traffic and transportation project’s situational characteristics, a threat to external validity [8]. Nevertheless, the ongoing industrial-strength project, together with the participation of software professionals, provided a firm footing for applying SSM in RE. Finally, in terms of reliability [8], we expect replications of our study to offer results similar to ours. Of course, the experience of SSM modelers may differ, but the underlying trends and implications should remain unchanged.

7. Conclusions

This case study was set up to investigate the role of SSM in understanding RE practices. To that end, we discussed some fundamental aspects of soft systems approach in relation to RE. We then studied a socio-technical software system developed by a research organization. We found that SSM’s holistic thinking could indeed identify a relatively complete set of RE flaws. However, we were unable to confirm that all SSM changes would fit in the organization’s context.

From our experience, we feel that SSM has a rich value in scrutinizing and improving the human-centered RE activities. More in-depth empirical studies are needed to lend strength to the preliminary findings reported here. Our future work also includes helping SrvU to select a proper set of requirements elicitation techniques, analyze coordination patterns, and tackle requirements prioritization and evolution problems.

References