

Article

Joint Impact of Agents and Services in Enhancing Software Requirements Engineering

Mekuria Sinkie ^{1,*}, Tor Morten Gronli ², Dida Midekso ¹ and Abdullah Lakhan ²

¹ Department of Software Engineering, Addis Ababa Science and Technology University, Addis Ababa 16417, Ethiopia; dida.midekso@aau.edu.et or mideksod@yahoo.com

² Department of Technology, Christiania University, 1190 Oslo, Norway; tor-morten.gronli@kristiania.no or tor-morten.gronli@westerdals.no (T.M.G.); abdullah.lakhan@kristiania.no or abduallahazalakhan@gmail.com (A.L.)

* Correspondence: meksin98@gmail.com or mekuria.sinkie@aastu.edu.et

Abstract: Requirements engineering (RE) is a significant aspect of system development stages in generating reliable software (SW). Despite RE's decisive impact on project success, SW systems still fail since there is a perplexity in sorting out requirements correctly. Researchers have tried several paradigms to deal with the specified challenges, such as agent-oriented RE (AORE), model-based RE, and service-oriented RE (SORE). By investigating the limitations of the independent use of these paradigms, this research sets an objective that proposes a framework which integrates the two paradigms (agent and service) on top of social media to enhance the SW RE processes. Thus, the research addresses challenges in gathering adequate requirements, detecting alignment between business requirements and SW products, prioritizing requirements, and recommending innovative ideas. The research has mainly adopted an empirical research methodology for SW engineering. Accordingly, two distinct expert groups have been formed based on their previous experience in AORE and SORE, respectively. The experts have been selected from enterprises and academic institutions, and they participated in our case study. After performing the necessary assessment based on specified criteria, those experts in the first group have reported that CASCARE (Collaboration of Agents and Services for Crowd-based Requirements Engineering) with a score of 93.7% is found to be better than that of AORE with a score of 88.7%. Moreover, experts in the second group have declared that CASCARE, with a score of 92.3%, is better than SORE, with a score of 83.7%. In both cases, improvements have been observed, which reveals that the synergy of the CASCARE features has a better impact on the RE process than utilizing individual approaches. Moreover, in order to demonstrate the applicability of CASCARE, feedback has been gathered from a focused crowd of local pharmaceuticals using a mini-prototype. Accordingly, 250 requirements related comments have been gathered from the discussion forum, and 1400 keywords were generated. Then, after performing a sentiment analysis using NLP algorithms, the result was demonstrated to experts. Therefore, 93% of gurus strongly agreed on the applicability of CASCARE in real projects.

Keywords: requirements elicitation; agents; services; social media; RE-processes; framework



Citation: Sinkie, M.; Gronli, T.M.; Midekso, D.; Lakhan, A. Joint Impact of Agents and Services in Enhancing Software Requirements Engineering. *Electronics* **2023**, *12*, 3955. <https://doi.org/10.3390/electronics12183955>

Academic Editors: Scott Uk-Jin Lee, Sanghyuk Lee, Soo Kyun Kim, Asad Abbas and Seokhun Kim

Received: 2 July 2023

Revised: 24 August 2023

Accepted: 25 August 2023

Published: 20 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The significance of dependable SW products and services spans various domains like healthcare, transport, education, and business. SW development for these services navigates diverse project life-cycle phases. RE plays a pivotal role; nevertheless, many systems falter due to inaccuracies in requirement crystallization. SW project failure issues often stem from misunderstandings, incomplete, or inadequately managed requirements [1]. Michael Kingsman [2] reveals that 68% of SW projects face uncertain success, often attributed to deficient RE. Despite technological progress, projects still need to improve, and poor RE remains a chief contributor.

Compounding these failures is the misalignment of RE process enhancement with shifting IT-related technologies like IoT, Big Data, Multi-Agent Systems, and Cloud Computing. Researchers have tried paradigms, such as goal-oriented RE, agent-oriented RE, model-based RE, and service-oriented RE; nevertheless, the challenges persist. Herein, we propose a hybrid framework that integrates the service and agent paradigms, aiming to revolutionize SW RE practices.

The service-oriented paradigm, integral to the hybrid framework, offers a large-scale SW development approach, countering challenges in component-based methods. As a result, service-oriented computing garners attention, wherein many applications have been developed accordingly [3]. This aligns with future trends where cloud computing and IoT concepts merge seamlessly with services. Service-oriented architecture (SOA) initially promises flexibility, adaptability, reusability, and interoperability. However, current SOA implementations fail to deliver these attributes [4], needing more dynamism and autonomy. In SORE context, the inability to fully exploit these SOA attributes has an adverse impact on providing better alignment between business requirements and SW products.

Moreover, before engineers propose a solution to align services to meet business requirements, there has to be a sufficient requirement elicitation process. Nevertheless, enterprises desire rapid solutions and developers are often under pressure with firm deadlines. Consequently, they attempt to offer a solution where requirements are compromised based on available services. This leads to unsatisfied customers, thereby creating a project falter.

Given these challenges, this research delves into a novel hybrid paradigm, exploring the synergy between agent and service paradigms on top of social network sites (SNS), for improved SW RE processes.

The motivation behind considering agent orientation as part of the proposed framework is that agent paradigms currently offer a considerable idea to many models and SW developments. Agents have properties, such as collaboration, self-organization, and pro-activities [5]. For instance, agent technology might assist significantly during wrapping up service functionality in a way that it becomes highly adaptable, cooperative, and self-organizing [6]. In particular, this service functionality can be accomplished with better efficiency by agents at a higher level than service protocols whose main accountability is just checking the interfaces of service descriptions [6]. This feature of agents is important, since once services are published, they may not have an opportunity to update themselves dynamically.

The desire to consider crowd collaboration through SNS also facilitates crowd-based RE, supplementing existing RE approaches. This introduces the potential to harness online communities for user-driven innovation in SW RE. Crowd-based information gathering is an emerging approach to enhance RE tasks [7]. In the hybrid service and agent paradigm context, crowd-sourced information becomes a valuable knowledge base. This approach facilitates various RE activities, including stakeholder elicitation, requirements completeness validation, requirement prioritization, and handling of continuous user feedback. Furthermore, analysis of text and usage data from social networking sources yields recommendation features for requirements, thereby elevating SW quality. The proposed framework, CASCARE, harmonizes goal-oriented agents and services, culminating in a social media-driven RE method for SW projects.

In addition to the challenges related to the independent use of SW RE paradigms, discussed in this section, other obstacles have been observed when previous RE approaches have been applied for large-scale and crowd-centric SW developments [8]. There has been a challenge to fully exploit SNS, complicating the effort of devising the RE process to be more innovative and participatory. In addition, there is a problem to thoroughly expedite patterns for detection of requirements completeness and sentiment analysis related to unrelated requirements gathered from users. Moreover, a limitation has been detected in organizing knowledge base for prioritizing requirements [8].

Furthermore, the difficulties in undertaking SW RE processes are more exacerbated by the obscurity in fully perceiving user needs when users are distributed in various locations. According to a research by Chemuturi [9], there are many misunderstandings

about requirements, wherein a large part of SW project failure can be associated with improperly defined and understood requirements.

Therefore, in order to address the stated gaps, three research questions are defined:

1. What will be the components of the proposed framework (CASCRE)? And what are the complementary relationships that exist between agents and services in the context of SW RE?
2. How can intelligence of users in social media assist the task of RE processes?
3. How can an evaluation scheme for the proposed framework be carried out?

Accordingly, the objective of the research is to design and implement a framework that integrates agents and services, on top of SNS, in order to enhance the SW RE process.

It is envisaged that the output of the proposed research, after proper design and implementation of the framework, will have many contributions as follows:

- By taking into consideration the combination of the different approaches, formulation of concepts will be generated, which is represented by an abstract component for correlation between services and agents. Consequently, an effort will be made in order to bring out shared concepts related to SW RE in each paradigm and see how the key concepts of collaboration are achieved.
- A framework will be defined that consolidates services and agents, on top of SNS, in the hybrid paradigm, by describing the interconnection between them, in relation to RE processes.
- The framework will enhance gathering, analyzing, recommending, and prioritizing user requirements.

The rest of this paper is organized as follows. Section 2 discusses related works. In Section 3, we describe the methodology utilized for the study. Results and findings are then presented in Section 4. In Section 5, a case study to demonstrate the practical applicability of the proposed approach is illustrated. Then, discussion and limitations are presented in Section 6. Finally, conclusions and future works are stated in Section 7.

2. Related Works

This part of the study will discuss related works in general about SNS-based RE, and specifically concerning AORE, SORE, service-oriented agents for RE, the role of natural language processing (NLP), and machine learning in RE. Researchers have investigated the utilization of SNS for assisting RE tasks, in recent years. Lim et al. [10] have used SNS for collaborative filtering of RE feedback by identifying key stakeholders and their corresponding consolidated preferences. This is due to the fact that inadequate recommendations from stakeholders during the early stage of an RE process indicates a low degree of user participation, resulting in project failure. The specified research is supported by a web-based tool implementing the StakeNet method [10]. The web application invites each user to enter and rate requirements proposed by another user, and finally, dominant requirements with maximum aggregate value will be selected. The significance this research adds to our work is that it gives us insight into how collaborative filtering can be handled in RE, specifically in recommending and forecasting users' desire for features of software systems, by gathering preference data from several users.

The work by Lohmann et al. has tried to deliver an essential pillar for the early phases of RE by initiating a user-driven knowledge engineering process for requirements elicitation [11]. The researchers have been able to launch a web-based tool called soft-wiki, which allows stakeholders to provide their feedback about requirements. The tool has a semantic web feature that assists requirement engineers in further analyzing gathered requirements from users. According to the researchers, there is less participation from users who did not learn the tool's usage. The research contributes a vital role to our work in relation to the benefits of assessing user feedback from different perspectives and various analysis possibilities.

The study by Jacek et al. has discussed the benefits of analyzing app reviews on SNS and its impact on SW engineering [12]. They have discussed how SW engineers elicit new requirements about intended app features, by considering associated user preferences. The authors have also discussed how this approach facilitated the testing phase with the help of bugs reported by users. Moreover, they have discussed the role of this review in assessing and managing change requests. Our research has customized and adopted the methods stated in the paper, such as data extraction, clustering, and sentiment analysis.

With regard to crowd-sourcing in RE, some research issues have been investigated. Hossein et al. [13] have created a SW prototype, which uses a crowd-sourcing approach to identify stakeholders and their requirements. Adepetu et al. [14] have stated that the use of crowd-sourcing in RE is in the early stage.

Concerning the role of agents in RE, the study by Norifunni et al. [15] has introduced an event mining system in SNS. Based on users' input, the system presents related events that can satisfy users' interests. In order to help users obtain a better experience, the system displays output in the form of a graph structure called event graphs. The authors have implemented two kinds of agents. The first is an agent to detect news articles, and the second is an agent to detect micro-blog posts on the web. The two agents have interacted with each other to share data. It has exhibited a significant role in our research since it elaborates intriguing clues concerning the interaction of agents.

The research by Ahmed et al. [16] has detailed modeling of goals during requirements elicitation to understand problems associated with organizational structures. The paper has tried to address the effect of modeling agents if the goals were elicited in early or late phases. Moreover, the paper has stated how to model these goals in the case of uncertainty. The role of i^* framework in goal-oriented requirements language (GRL) was discussed. The proposed study considers key features of the study, such as the relationship among the RE activities, goals, and social aspects.

The research paper by WaiShiang et al. [17] has emphasized that due to the rapid expansion of mobile devices and associated SW, there is a corresponding swift prototyping approach without sufficiently understanding the domain. To overcome this problem, the researchers have adopted an AORE. To better understand the domain, the proposed research has analyzed the method described in this research by combining it with SNS.

Regarding the impact of the service-oriented approach RE, researchers have tried to make a model that understands the new system by exploring existing services and correlating requirements and available services. The research issue identified by Bano and Ikram [18] is focused on dynamic service discovery and fulfilling requirements with service discovery. Moreover, concepts related to requirement change management, semantic gaps in RE, and lack of standards in the RE process for SORE were discussed. The study gives a clue to our research concerning how stakeholder needs could be mapped to service requirements.

The authors in [19] claim the need to apply SORE to enable organizations to adapt to changes quickly. The authors view the study from two aspects: requirements for services and service-oriented RE, which incorporates issues such as the reusability of features about previous services. It contributes to our study regarding the approach the authors have applied regarding reusability features from previous services, to boost the requirement for a newly implemented system.

Although there needs to be more related research concerning the integration aspect of hybrid paradigms, specifically for RE, there are some research works concerning the integration of SW engineering paradigms and models for general SW development. One research is "Integrating Knowledge Modeling and Multi-Agent Systems" by Mario Gomez and Enric Plaza [20]. The paper has illustrated a framework for Agent Systems that enhances the reuse of agent capabilities through a knowledge base consisting of multiple application domains and supports agents' automatic, on-demand configuration for various activities.

The other research concerning hybrid paradigms is "Current usage of component-based principles for developing web applications with frameworks", performed by Matija Novak and Ivan Svogor [21]. The paper has investigated how integrating components

and web applications might produce better reusability, compatibility, and means of using components for web application development.

In order to effectively analyze and extract user feedback, a system is required that enables SW engineers to understand users' need. In this regard, various approaches have emerged to tackle this issue. A work by L.V. Galavics et al. [22] has demonstrated the use of NLP techniques, tools, and resources for a variety of requirement documents, to support the task of analysis performed at the RE phase. The specified research stated the role of NLP in facilitating the detection of ambiguity, key domain concepts, and traceability between requirements. Moreover, user feedback analysis tasks can be facilitated by machine learning algorithms. In this regard, the work performed by A. Perini et al. [23] has discussed the impact of machine learning features and algorithms in RE analysis and validation tasks. Furthermore, Abualhaija et al. [24] has revealed a machine learning approach for sorting out requirements with text analysis and specification.

Some related works in relation to research strategy have been reviewed, such as, empirical research methodology for software engineering and design science research method [25].

3. Research Methodology

The research methodology in the context of the proposed approach is comprised of the following major steps. The first step is suggesting a research strategy for the proposed study. Then, a discussion regarding related works (see Section 2) is accomplished. Consequently, data collection and analysis for ensuring problem relevance and construction of the conceptual framework will be performed. Subsequently, a customized list of evaluation criteria for SW RE methodologies will be prepared in order to undertake verification of the proposed approach. Afterwards, a result analysis procedure will be conducted. Finally, construction of a case study and corresponding data analysis will be carried out in order to visualize the practical applicability of the proposed approach.

3.1. Research Strategy

In order to select a strategy for this research, we need to assess the research intention, nature of the problem, and research approach. Consequently, qualitative and quantitative methods have been used for collecting and analyzing evidences, thereby formulating conceptual framework. Construction of the specified framework and associated evaluation is a central activity in the problem definition and research objective. Therefore, the research will create an artifact intended to clarify the identified research problem. In particular, the strategies for this research must take into account that, software RE in the case of applying hybrid software engineering paradigms on top of SNS, can be improved by creating a framework and corresponding mini-prototype that facilitates the elicitation process for SW development in an enterprise.

Moreover, since a research within RE is often concerned with stakeholder interactions, which is considered to be essential in eliciting complete and quality requirements, empirical research methodology for SW engineering [26] is proposed. This methodology is suitable for providing a relevant framework for interaction-focused methods in RE [27]. Although our methodology is mainly based on empirical research methodology, it is acquainted with the main tenants of design science research (DSR) [28,29]. In particular, the methodology for this research has considered the design of artifact and evaluating it in a relevant context. We have adopted some concepts from DSR, due to motives, such as better enhancement of the knowledge contribution regarding the result of our research, to acquire an improved view on the analysis of research gap and problem relevance, and to obtain more appropriate visualization of artifact design. Table 1 provides core activities in our research strategy, corresponding sections in the article and associated DSR phases.

Table 1. Concepts adopted from DSR methodology in the proposed research.

S. No	DSR Phases	Activities	Sections in the Article
1	Problem Identification	- Identify problem - Literature research - Expert survey - Pre-evaluation for relevance	Sections 1–3
2	Design Solution	- Design artifact - Assess related design solution - Evaluation with software RE verification methods	Sections 3 and 4
3	Research–Rigor	- Iteration of activities in problem identification, design solution, and evaluation - Case study with mini-prototype	Sections 1 and 3–5

3.2. Expert Survey

The purpose of performing an expert survey in this section is to ensure the problem's relevance, which is identified as a research gap in the proposed study, gather essential ingredients, and obtain insights about the design of the conceptual framework. Moreover, as it has been reviewed in Section 2, SW development has been changing from the traditional approach to data-driven, and thus the role of users has become crucial. Therefore, target participants have been first identified by academic institutions and SW development companies to assess this perception and undertake the survey practically. The survey was created by developing questions that align with the research questions, objectives, and contributions. Moreover, to effectively measure respondents' opinions, the authors have conducted repeated discussions to avoid ambiguous or biased questions, such as the order of answer categories. Consequently, the survey was generated by customizing an online form called Forms.app [30]. The survey consists of 30 multiple-choice questions on a 5-point Likert scale. Subsequently, the questionnaires depicted in Table A1 (Appendix A) have been provided to experts.

Although outreaching many experts and academicians is believed to be even better, this preliminary need assessment survey is collected from 96 practitioners (Requirement analysts, Business analysts, SW developers, SW testers, Programmers, IT managers, candidate PhD students, Assistant Professors, Instructors, and IT-related Professionals) located in four SW development companies and four academic institutions operating in Addis Ababa, Ethiopia. The educational background of the respondents is 55% first degree, 35% master's level, 10% PhD and candidate PhDs. The age of practitioners responding to this survey is about 37% under the age of 35, and the remaining 63% are above 35 years of age. It indicates that most of them (more than 62%) are greater than 35 years of age. Regarding professional experience in the industry, about 42% of them have less than 5 years of experience, and the remaining 58% have worked for more than 5 years. Moreover, for the sake of participants' flexibility and considering sufficient time for an interactive discussion, both online and in-person survey types have been considered. Furthermore, a discussion forum session has been arranged to obtain a hint from experts about some of the features that need further elaboration.

3.3. Need Assessment Result

The result of the survey, performed to gather data about the proposed hybrid RE, CASCARE, implementation, and its role in SW systems development, will be elucidated in this section.

Respondents were asked to rate the question “CASCARE enables to obtain more accurate requirements in SW RE process” and their response indicates that 48% of them “Strongly Agree” and 42% of them “Agree”, inferring the demand in SW development companies for improving accuracy in the process of SW RE tasks. Respondents have also rated the question “Applying CASCARE is difficult to organize and coordinate the process in eliciting requirements”. Accordingly, only 5% responded as “Strongly Agree”, 9% as “Agree”, and “81% disagree”, showing the presence of the demand in utilizing the proposed approach in SW RE.

Respondents were also asked to rate the question “Applying CASCARE can facilitate recommendation of related requirements for the intended SW system”. Accordingly, the ability to understand and recommend requirements is rated by more than 83% of respondents. Moreover, a high proportion of the respondents (85%) reported that by applying CASCARE, the crowd competence can support the detection and gathering process of creative requirements. Figure 1 provides distribution of statistical values, concerning users’ preference, in relation to innovative features of the proposed model.

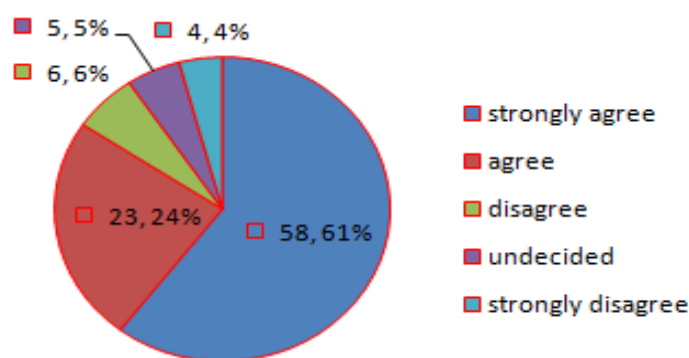


Figure 1. Survey result regarding “Innovative Feature about RE” in the proposed framework.

The survey has also invited respondents to provide their insight regarding the request: “Using CASCARE enables to gather more complete requirements relative to traditional approaches of RE process”. Consequently, the majority of the responses (88%) pointed out that exploiting the proposed approach can impart better completeness features for RE. Figure 2 presents evidence, regarding distribution of statistical values for the completeness feature of the proposed model.

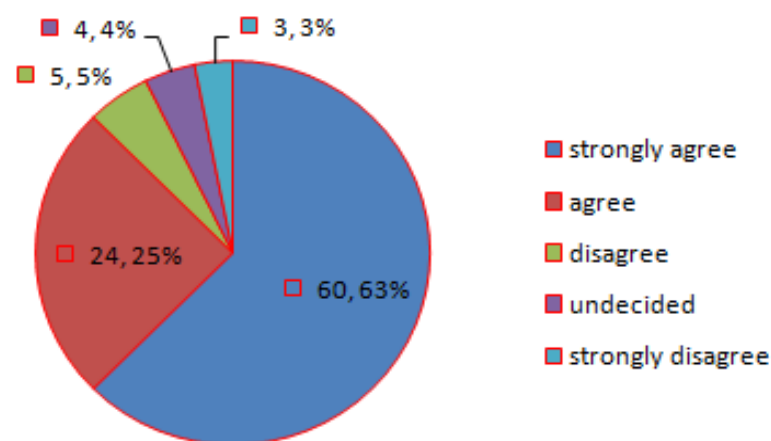


Figure 2. Survey result concerning “Completeness feature for RE” in the proposed framework.

Experts were also asked to suggest an irrelevant input related concept as “CASCRE might provide a chance for malicious users to enter the elicitation process and affect the overall quality of the requirements that is going to be elicited”. The response revealed that 9% of them “Strongly Agree” and 18% of them “agree”, which might be an indicator to devise a mechanism for filtering only important requirements and reject unwanted requirements.

Furthermore, 83% of respondents have agreed that CASCRE can facilitate reuse and prioritization of the requirements engineering process. Moreover, nearly all respondents (96%) agree that CASCRE can be adopted for different case studies, and accordingly recommends that a case study should be considered for the proposed approach. On the other hand, the majority of respondents (88%) suggested that comparison of CASCRE must be made with respect to the traditional approaches in order to ensure its reliability. In addition, statistical values of standard deviation and mean values, for representing the above percentage description, are depicted in Table 2.

Table 2. Statistical value for three types of requirement features.

Requirement Characteristics	Mean	STDEV	Median	Mode
Completeness feature for RE	4.5	0.95	5	5
Innovative feature for RE	4.41	0.98	5	5
Recommendation feature for RE	4.22	1.21	5	5

3.4. High-Level Conceptual Framework

This section describes tasks performed during the process of sketching the logical model. The need assessment result conducted in Section 3.3 is the principal input for construction of the conceptual framework. Moreover, before the high-level framework is designed, the idea has been proposed to expert groups through a zoom discussion. In order to identify the components of the intended framework, major issues were analyzed, such as purpose of the research, boundary of the research, input to the framework and associated output, research questions, contribution of the research, need assessment result, and review of related works. After identifying all the components of the framework, the task of incorporating the components into the draft logical model has been executed. Therefore, a high-level framework has been designed, as shown in Figure 3.

3.5. Description of the High-Level Framework

The CASCRE framework uses organizational goal, previously related service functionalities, and user feedback to provide insights about requirements for an intended system through an integrated framework comprising of agents and services on top of SNS. Figure 3 depicts major parts of the proposed framework and corresponding modules. In the following discussion, further elaboration of key units, modules, tasks, and associated roles will be described.

3.5.1. Specify Goals

The agent-based goal in the context of this framework describes a goal as an organizational objective that the system should accomplish through cooperation of agents in executing the roles, thereby achieving associated goals. Therefore, the “Specify Goals” unit in the CASCRE framework represents the requirements for the SW system to be created hierarchically. The hierarchy moves down to a lower level until the leaf level is attained, as indicated in Figure 4. Leaf levels represent concepts and keywords that evolved from the interface in the SNS, which represent specific features of a requirement. The service model complements the agent model in aligning business requirements with the intended software product, as indicated in Figure 3.

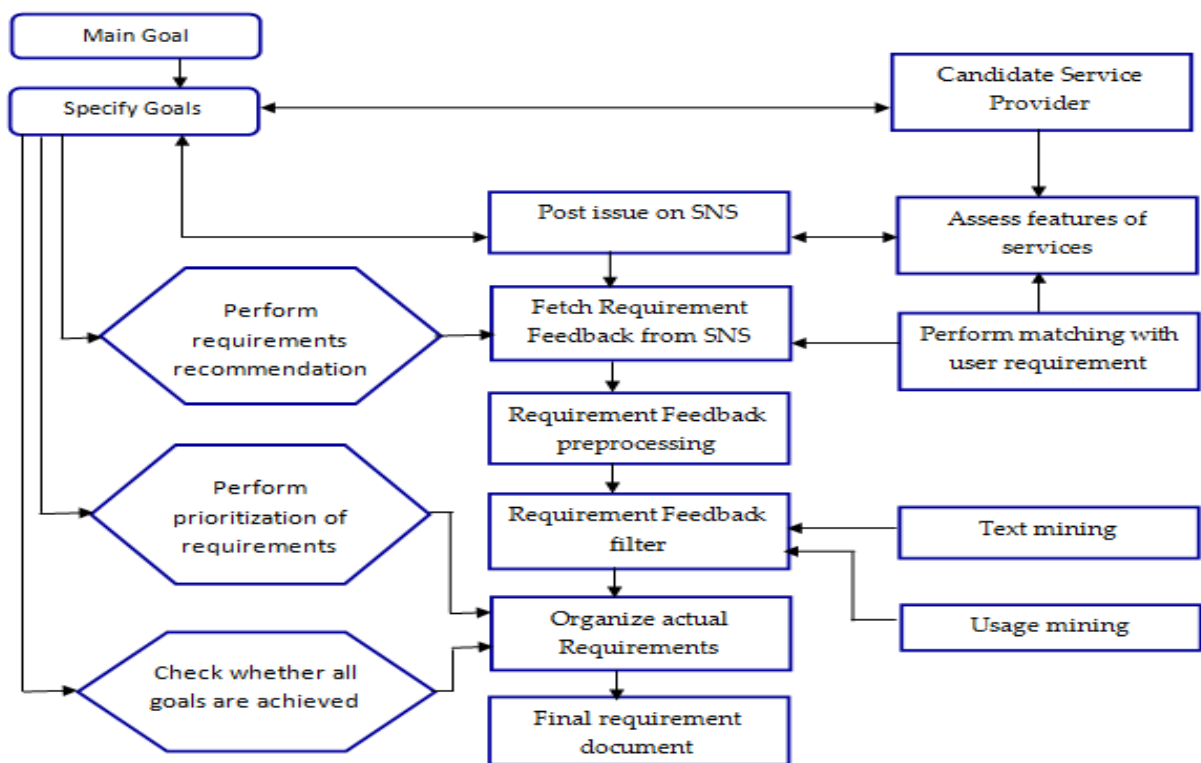


Figure 3. High level framework for CASCRE.

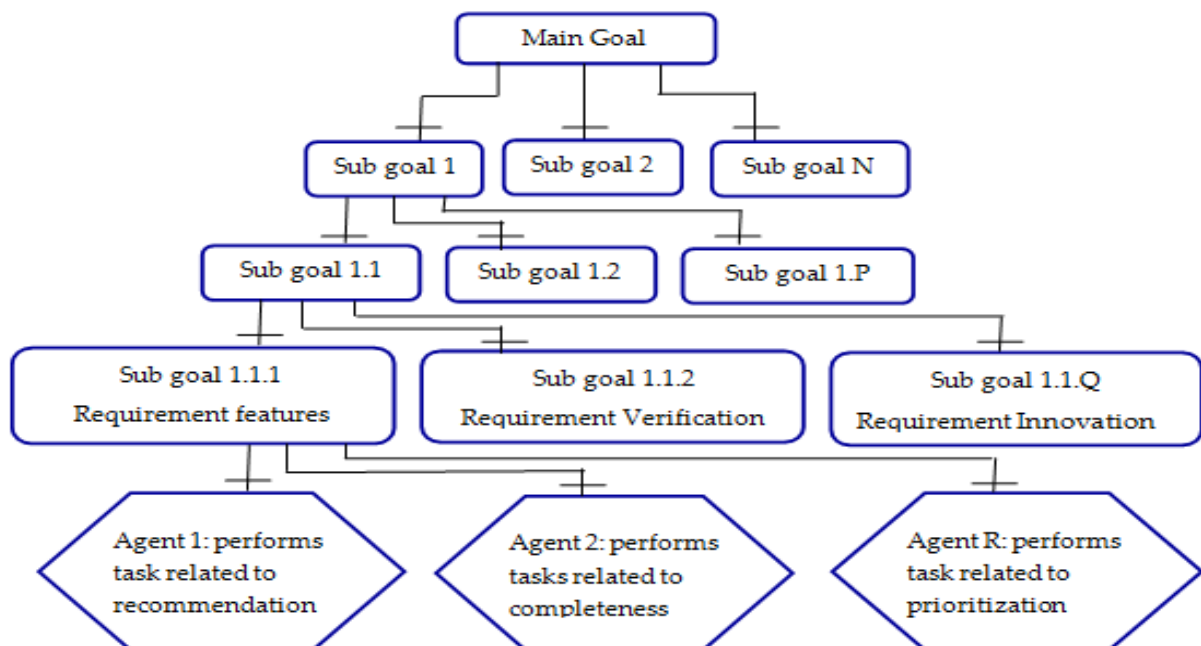


Figure 4. Assignment of goals to agents in AORE.

3.5.2. Assess Features of Services

The “Assess features of Services” unit in the CASCRE framework is responsible for facilitating the reusability of features regarding existing related services in the process of RE elicitation in relation to a new SW system to be developed. The updated requirement document, generated after proper negotiation, facilitates extraction of requirement concepts from the SNS for the intended system, as depicted in Figure 5.

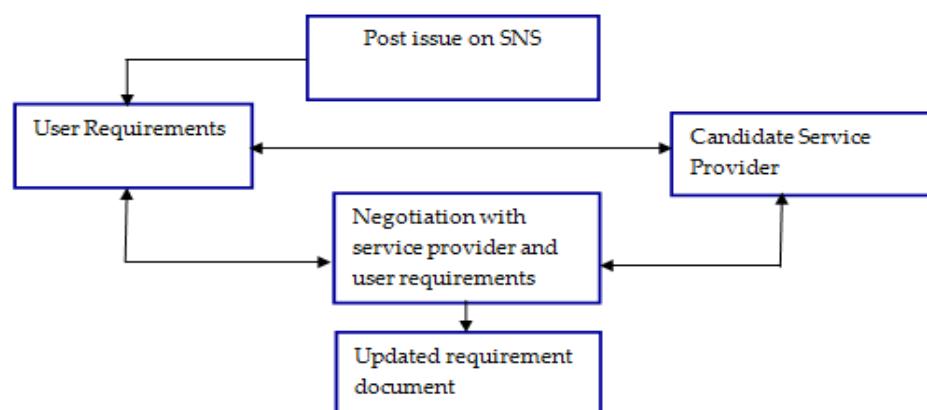


Figure 5. A framework for alignment of business needs to available services.

However, due to tight deadline and developer's reluctance, there may be a tendency to overlook some of the user requirements and utilize available services from service providers, without allotting adequate time for the procedure of selecting relevant services. This, in turn, creates a problem in undertaking the proper alignment of business requirements to SW products and might not satisfy the minimum requirement set for the system to be developed. To prevent these challenges, the agent model complements the alignment process of business needs to service, by assigning a dedicated agent that checks whether all goals are achieved, as shown in Figure 3.

3.5.3. Post Issue

The purpose of this module is to post issues for participants on a dashboard regarding requirement elicitation for a particular problem. The discussion is made on a specific domain and is conducted in an online discussion forum basis.

3.5.4. Requirements Feedback Preprocessing

This module is responsible for tasks of data composition. It constitutes activities, such as gather user input data from "Fetch Requirement Feedback from SNS" module and removal of connectors. There is also a need to further preprocess the data in this module to reduce ambiguity, namely, converting input data into lower cases to reduce ambiguity by normalization. For instance, the words "Property", "PROPERTY", and "property" will be transformed into the same representation "property". Moreover, lemmatization procedure is exploited to normalize texts into their equivalent root form. For instance, words such as "verify", "verified", and "verifying" will be converted to the text "verify".

3.5.5. Requirements Feedback Filter

This module is responsible for identifying feedback pertinent to the business logic of a given scenario. To carry this out, the module correlates information from business objectives which were assessed by "goal-based recommendation agent", updated requirements generated from the negotiation phase of "user requirement and existing service features", and keyword and concepts gathered from crowds in the customized SNS platform. In order to facilitate the feedback filter process, text mining and usage mining modules are utilized for the purpose of detecting key features that represent stakeholders' interest.

3.5.6. Organize Actual Requirements

This module receives key features from "Requirement Feedback Filter" component of the framework and converts them to actual requirements. To this end, it capitalizes dedicated agent, "Perform Prioritization of Requirements", in order to assess the significance of each requirement. This module executes its task by providing a ranking list of the most discussed requirements related comments or keywords (namely, most liked requirements related comments) based on calculated scores on the participants' opinion and sentiment.

As a result, system analysts can pick the requirements that should be favored for the purpose of implementation. Moreover, a dedicated agent “Check achievement of all Goals” is used to verify whether all goals are accomplished in this stage before the final requirements document is generated.

4. Evaluation and Results

4.1. Set Up Evaluation Criteria

In order to undertake a critical assessment for the proposed framework, an evaluation criterion has been prepared by customizing the criteria adopted in [31,32]. The evaluation measure we have used in this case is qualitative and is defined as “H” for highly supportive, “M” for medium or partially supportive, and “L” for less supportive.

Consequently, associated numerical values for (H, M, and L) are (3, 2, and 1), respectively. Then, we have customized and organized 21 evaluation criteria in order to facilitate the comparison between the specified methodologies, thereby demonstrating the impact of the integrated approach over individual approaches.

After the criteria have been identified, we have consulted senior system analysts and the corresponding weight is assigned as shown in Table 3, depending on the significance impact of each criterion. The blank evaluation criteria form, prepared for expert participants, is indicated in Appendix C.

Two distinct expert groups, each with 12 participants, have been formed based on their previous experience in AORE and SORE, respectively in order to carry out the evaluation. Subsequently, after participants’ have filled their response concerning the evaluation criteria, the response has been compiled and findings have been summarized in Table 3. In particular, the specified table recaps the results based on participants’ conclusion about how the three approaches react to the evaluation criteria with strongly, partially, or less likely options.

The following formula is used to calculate the grand average score for each competent methodology:

$$GAs = \sum_{i=1}^{Tc} Wc(i) * As(i) \quad (1)$$

where:

$Wc(i)$ = Weight of criteria (i), $As(i)$ = Average score (i),

Tc = Total Criteria = 21.

And

$$As = [\sum_{j=1}^{Tp} [Rv]pj] / Tp, \quad (2)$$

where:

$Rv \in [1, 2, 3]$, Response value suggested by person j , Pj , and $L = 1$, $M = 2$, $H = 3$.

Tp = Total participants = 12 for Group 1, (also $Tp = 12$ for Group 2).

4.2. Result

For better elaboration and interpretation of the result, we have classified a description concerning the result into two categories: overall findings and specific findings.

4.2.1. Overall Findings

As we can observe from Table 3, participants in both groups claim that CASCRES has performed better than individual approaches. In particular, after performing the necessary assessment based on the specified criteria, those experts in the first group have reported that CASCRES with score of 93.7% is found to be better than that of AORE with a score of 88.7%. Moreover, experts in the second group have declared that CASCRES with a score of 92.3% is better than SORE with a score of 83.7%. A summarized overall finding, generated from Table 3 has been illustrated in Table 4.

Table 3. Summarized differences between CASCARE and individual approaches.

Cr. No	Evaluation Criteria	Criteria Weight (%)		Avg. Capability Score (Group 1)				Avg. Capability Score (Group 2)			
				AORE		CASCARE		SORE		CASCARE	
				Avg. Score (3)	(Avg. Score) × (weight)	Avg. Score	(Avg. Score) × (weight)	Avg. Score	(Avg. Score) × (weight)	Avg. Score	(Avg. Score) × (weight)
1	Assist gathering of complete requirements	6	0.06	2.8	0.17	2.8	0.17	2.4	0.14	2.8	0.17
2	Support detection of innovative idea	5	0.05	2.8	0.14	2.9	0.15	2.4	0.12	2.8	0.14
3	Capability to support COTS-based RE process	5	0.05	2.4	0.12	2.8	0.14	2.8	0.14	2.8	0.14
4	Assist identification of conflicting requirements	4	0.04	2.8	0.11	2.8	0.11	2.4	0.10	3.0	0.12
5	Assist identification of reusable requirements	5	0.05	2.4	0.12	2.8	0.14	3.0	0.15	2.8	0.14
6	Assist identification of ambiguous requirements	4	0.04	2.8	0.11	2.8	0.11	2.0	0.08	2.4	0.10
7	Assist identification of stakeholders	5	0.05	2.8	0.14	3.0	0.15	2.4	0.12	3.0	0.15
8	Facilitate communication and negotiation	6	0.06	2.4	0.14	2.8	0.17	2.4	0.14	2.8	0.17
9	Assist understanding of social issues	4	0.04	3.0	0.12	3.0	0.12	2.4	0.10	3.0	0.12
10	Assist gathering of implicit users interest	5	0.05	2.8	0.14	2.8	0.14	2.4	0.12	2.8	0.14
11	Assist modeling of requirements	5	0.05	2.4	0.12	2.8	0.14	2.8	0.14	2.8	0.14
12	Assist detection of non-functional requirements	5	0.05	2.8	0.14	2.8	0.14	2.0	0.10	2.4	0.12
13	Support decomposition of complex requirements	6	0.06	2.8	0.17	2.8	0.17	2.4	0.14	2.8	0.17
14	Assist prioritization of requirements	5	0.05	2.8	0.14	2.8	0.14	2.4	0.12	2.4	0.12
15	Practical applicability of the method	4	0.04	2.4	0.10	2.8	0.11	2.4	0.10	2.8	0.11

Table 3. Cont.

Cr. No	Evaluation Criteria	Criteria Weight (%)		Avg. Capability Score (Group 1)				Avg. Capability Score (Group 2)			
				AORE		CASCRES		SORE		CASCRES	
				Avg. Score (3)	(Avg. Score) × (weight)	Avg. Score	(Avg. Score) × (weight)	Avg. Score	(Avg. Score) × (weight)	Avg. Score	(Avg. Score) × (weight)
16	Facilitating good learning curve	5	0.05	2.4	0.12	2.8	0.14	2.8	0.14	2.8	0.14
17	Facilitating recommendation feature	4	0.04	2.9	0.12	2.9	0.12	2.0	0.08	2.9	0.12
18	Facilitates verification	4	0.04	2.4	0.10	2.8	0.11	2.8	0.11	2.8	0.11
19	Providing tool support	5	0.05	2.8	0.14	2.4	0.12	2.9	0.15	2.4	0.12
20	Integration with other methodologies	4	0.04	2.4	0.10	2.8	0.11	2.8	0.11	2.8	0.11
21	Gathering requirements from distributed teams	4	0.04	2.4	0.10	2.8	0.11	2.8	0.11	3.0	0.12
Total Average Score (3)				2.66		2.81		2.51		2.77	
Total Average Score (%)				88.7		93.7		83.7		92.3	

Table 4. Overall findings.

Methodology	Total Average Score	Group	Difference	Dominant Method
CASCRE	93.7	1	5	CASCRE
AORE	88.7	1		
CASCRE	92.3	2	8.5	CASCRE
SORE	83.7	2		

The result of the overall evaluation (Table 4) suggests that the average performance of CASCRE compared to AORE and SORE is 93. In particular, in both cases, the improvement reveals that the synergy of the proposed approach has better impact in RE process than that of utilizing individual approaches separately. In particular, CASCRE has enhanced the SW RE process relative to individual approaches.

The percentage score gap between (CASCRE and AORE) which is 5 is better than that of (CASCRE and SORE) which is 8.5. It indicates by extension that agent-oriented RE is somehow harmonious to the proposed model.

There is also a 1.4% score difference for the same framework, CASCRE, when it is evaluated by two different groups (namely, 93.7 and 92.3). The small difference might be due to the relative performance of the competent methods. Nevertheless, the average performance of 93 is promising.

4.2.2. Specific Findings

The specific findings description is made by pinpointing individual criteria from the evaluation list. A summarized specific finding generated from Table 3 has been illustrated in Table 5.

Table 5. Specific findings.

Requirement Feature	SORE (%)	AORE (%)	CASCRE (%)	Remark
Assist requirement completeness	80	93	93	
Assist prioritization	80	93	93	
Facilitate recommendation	67	99	99	
Providing tool support	99	93	80	
Detection of innovative idea	80	93	99	

In the evaluation criteria “Assist gathering of complete requirements”, both AORE and CASCRE have a score of 2.8 (93%). The reason for the similar value could be that both approaches have the “check all goals accomplishment” role in their respective models. Therefore, the accomplishment of all goals is an indicator for the success of completeness property. On the other hand, CASRE performed better than SORE by a score of 0.4 for the specified criteria. This indicates that CASCRE has enhanced the completeness feature by 13% compared to that of SORE.

In terms of the criteria “Support detection of innovative idea”, 13% of the CASCRE enhancement has been observed with respect to SORE and 6.7% improvement relative to AORE. The reason for the better performance of CASCRE in both cases is that, in addition to collaboration of agents and services, the crowd competence in SNS has supported the detection of creative requirements for the intended software system.

In relation to the evaluation criteria “assist prioritization of requirements”, both CASCRE and AORE have exhibited equal performance. However, both have outclassed SORE by 0.4 (13%). Both CASCRE and AORE examine the frequency of requirements for the purpose of deciding their priority. This is facilitated by a dedicated agent that comprises the keyword rank module in their framework.

With regard to “facilitate recommendation” evaluation criteria, both CASCARE and AORE demonstrate equal performance. However, both have surpassed the SORE method. The expert’s viewpoint could be that both support intelligent software components in the form of dedicated agents which assist the RE task. Moreover, the SNS facilitates the recommendation process by providing many stakeholder preferences, from which frequently stated keywords can be generated.

In terms of “providing tool support” criteria, SORE has performed better than both AORE and CASCARE. The experts’ outlook could be that a tool for CASCARE is under construction. Moreover, compared to AORE-based tools, SORE exploits various types of industry standard UML tools.

5. Case Study

As part of our research methodology, this section discusses a case study for the purpose of visualizing the practical applicability of the proposed framework.

5.1. Introduction

“MEDICOR Africa Pharmaceuticals and Medical Supplies PLC” is a pharmaceutical company located in Ethiopia. The goal of the PLC is to provide health care solutions to the health sector market and meet the demand of pharmaceuticals, medical supplies, and diagnostics by importing the specified products and distributing them. Besides the normal business activity of the company regarding importing, distributing, and selling of pharmaceutical items through its branches, the company also would like to integrate its system with major hospital systems in order that it is able to know what kind of drug is frequently ordered, for which type of disease, and the most common types of disease that patients register for frequently. This is to balance demand and supply, since some drugs are imported in excess and result in wastage due to the expiration date. On the other hand, there is a critical scarcity problem for other drugs.

The other concern is regarding a fake expiration date and counterfeit drugs, which is the reason for customers to receive an expired drug that consequently affects their health. Therefore, the company would like to register the original expiration date in its server and intends to allow customers in branch pharmaceuticals to access the genuine expiration date before they purchase it by using an application on their smart phone. Therefore, pharmaceutical PLC currently outsources a software development company called “Server IT Solution Provider”. It is composed of seven recently graduated software engineering professionals. The SW development company usually performs limited RE activities due to budget and few professionals with small experience. Moreover, there is a need to develop “online Pharmaceuticals items distribution and verification application” rapidly in order to be competent in the market. Therefore, the company is suitable for making an appropriate validation for the proposed CASCARE approach.

5.2. Sample Mapping of Natural Language Input into Usecases Using Components of the Specified Framework

To illustrate how the various steps in the proposed framework are utilized to map NL input to basic requirements, let us consider two specific scenarios:

Scenario 1: Drug registration by checking counterfeit status (Importer viewpoint).

Scenario 2: Drug purchase by checking expiration date (Customer viewpoint).

The following are sample feedbacks representing the two scenarios and are gathered from the discussion forum, which stand for the case study:

- Users want to check the expiration date before buying the medicine. Some users have also used the keyword *perish* in place of *expiration*.
- Users want to access the application from their cell phone.
- Check whether the drug is counterfeit or not, before buying, registering, and storing. Some users have also used the keyword *forged* in place of *counterfeit*.

- Users also mentioned that it would be better if the interface supports local language features, such as “Amharic local language”.
- Users request a description about the adverse effect of drug with respect to related cultural medicine.

Table 6 elaborates how the various steps of the framework transform the natural language input, stated in this section, to basic requirements in the form of use cases.

Table 6. Sample mapping of NL input to basic requirement.

Framework Elements	Detail Process and Algorithm		Scenario 1	Scenario 2	Clustered Keywords	Frequency	Feedback to Requirement
Post-Issue	Put comment on SNS browser		NL input	NL input			
Fetch-Requirement	-	Read from SNS browser	NL input + Org. goal recommend + Existing service recommend	NL input + Org. goal recommend + Existing service recommend			
	-	attach related concept from Org. goal					
	-	attach related concept from related existing services					
Preprocess-requirements	-	NLP usage	Input preprocessed	Input preprocessed			
	-	Split, Tokenize					
	-	Stop word removal					
	-	Lemmatization					
Requirement-feedback filter	-	Text mining Usage mining show frequency of keywords	Encode, register, record	Fake, forged, counterfeit	Fake Counterfeit Forged	33 27 40	
			Expire Perish	Encode Register Record	22 36 24		
					Expire Perish	80 20	
			Purchase, Acquire, Buy	Purchase Acquire Buy	50 20 30		
Organize-Requirements	-	Text Rank	“Register” selected “Expire” Selected	Register Expire	36 80	Register-Drug Check-Expire-Date	
	-	Word2Vec					
	-	K-mean					
	-	Prioritize keywords	“Forged” selected	Forged	40	Check-Forged-Drug	
	-	Cluster Keywords	“Purchase selected”	Purchase	50	Purchase-Drug	

Furthermore, the conversion of requirements to UML, de-facto standard, shows that it is possible to transform the key business operations assigned for a goal. For instance, Figure 6 shows a use case diagram that describes the process: customer requests medicine items online, system has carried out various processes, and then the intended item is sent to the customer. The other use cases with associated descriptions and actors are presented in Appendix B. Moreover, for the utilization of this research, a web-based discussion forum platform has been applied for the purpose of inviting a focused group of crowds from the pharmaceutical related sector in Addis Ababa, Ethiopia.

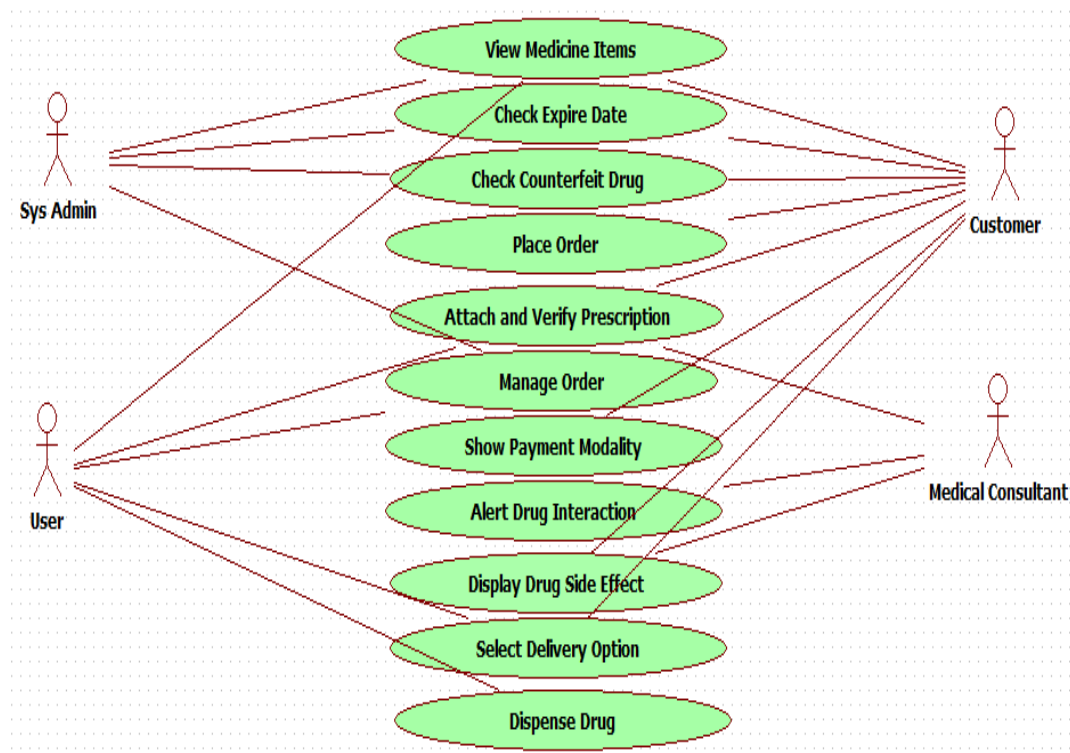


Figure 6. Use case diagram that shows “customer requests medicine items online requirement”.

5.3. Features Selection on User Feedback Analysis

Before analyzing user feedback, there is a need to identify features from initial requirements, business rules, and objectives of the company. As a result, effective analysis of requirement related comments from the feedback shall be facilitated. These features have potential capabilities to express for stakeholders what the intended application can accomplish. In order to find valuable data features for CASCARE, the process of capitalizing domain knowledge will be elaborated in this section. To that end, NLP operations, such as extracting features from sentences, selection, and associated optimization will be carried out. Therefore, there is a need to first identify the group of characteristics and associated representation in the process of extracting user feedback from a particular scenario. One of the approaches for the feature selection is the use of keywords as they have useful features for text classification [33]. Accordingly, keywords extraction process in this research allows for the process of identifying and correlating input from initial requirements content, domain experts’ knowledge, and users’ feedback on discussion forum.

The core features enable the proposed model to detect some patterns from participants’ feedback. These patterns are expected to embrace descriptions which contain continuous and repeated texts that describe the features and purposes of the intended application. Accordingly, identification of keywords pattern that frequently describe features was performed. Therefore, the mutual use of the specified keywords from initial requirements and the selected central features enable CASCARE to consolidate generated features for the intended application. These core feature selection and document comparison operations have been performed using NLP methods, such as RAKE_NLTK, Word Cloud, TextRank, Textrazor API, and associated string operations, such as read (), split (), concat (), cmp (), drop (), intersection ().

Word embedding model [34] has been also implemented in the experiment in order to apply the appropriate representation for texts where words that have the same meaning will have a similar representation. As a result, there is a possibility to facilitate categorization of requirements with similar concepts together. Consequently, this boosts the realization of the framework in detecting similar requirements which are written in different expressions.

For instance, Drug name, medicine name, and tablet name represent a similar concept and are mapped to Drug name. Similarly, treatment date, diagnosis date, and examination date can be gathered from participants in the requirements gathering process. Then, this input can be mapped to “treatment date” that can serve as input for the online pharmaceutical system, which is integrated with a hospital system.

5.4. Data Analysis on User Feedback

In order to analyze users’ feedback, the following approaches have been applied. Firstly, users have been given brief information about the local drug importer, distributor, and seller PLC. Secondly, feedback has been gathered from 32 pharmacists in the focused crowd of various pharmaceuticals. Moreover, after participants’ feedback has been gathered from the GUI, it has been exported to CSV file. From there, Python was used to process the CSV file in order to identify related concepts regarding the most discussed requirements. Consequently, a total of 250 requirements related feedback were gathered and about 1400 keywords have been generated from the requirements.

Thirdly, we have exploited NLP algorithms together with the implementation code for the purpose of checking the performance of the framework. These performance measures have been used to detect keywords which are accurately suggested by users’ feedback that align with available test keywords already generated from the existing service dataset and organizational business objectives. Therefore, after performing the experiment, the result shows that it is possible to identify requirements related concepts for the intended application.

In order to assist system analysts during the selection of most important requirements from user reviews, CASCARE ranks and extracts most relevant keywords from users’ feedback. Figure 7 has depicted keywords with associated frequency and ranks them from top to bottom. For instance, percentage frequency of the phrase “drug might cause additional side effects” is 33.5, frequency of the phrase “authorized users can override previously recorded directives of Drug” is 33.1, and frequency of the phrase “search and display drug items by category” is 23.9.

Keywords: [(33.5, 'drug might cause additional side effects '), (33.08333, ' authorized users can override previously recorded disclosure directives of Drug and patient records '), (26.07765, 'system must show drug side effect'), (25.4166666, 'search and display drug items by category'), (23.87765567765568, 'system must show drug expire date'), (23.220512820512823, 'system must register drug expire date'), (20.41666, 'filter target medicinal items based'), (19.82051282051282, 'system must register expire date'), (18.41666, 'electronic pharmacy records based upon'), (18.0, 'must support e-signature capture capability'), (16.8205, 'system must alert effect'), (16.0, 'signal minimum reorder level'), (15.82051, 'system must change units'), (15.82051, 'system must alert users'), (15.66666, 'must provide unambiguous direction'), (15.65, 'adjust drug dose based'), (15.37606, 'safe prescription managements system'), (15.22222, 'improperly filling prescription evidence'), (14.66666, 'check confirmed medicinal producer'), (14.0, 'Food and Drug control agency '), (13.83333, 'real time management reporting'), (13.566666, 'edit medicinal drug items'), (13.4, 'check drug storage location'), (13.288888, 'buy drug without prescription'), (13.0, 'rx image scanning feature').

Figure 7. Sample screen shot of most discussed requirements from Python shell.

Consequently, it allows experts to specify their interest in the requirements of intended application from keywords which are already sorted out. Then, the framework exploits those keywords to apprehend the most pertinent feedback on a discussion forum, expecting them to contain opinions that essentially correlate with the specified interests of stakeholders.

However, even if the output of keywords (Figure 7) is mostly related and exhibit general characteristics, it might be difficult for system analysts to search each related word

for further analysis. For example, both keywords “encode drug” and “record drug” could be used to describe the pharmaceutical system’s “Register-Drug” use case. Therefore, CASCARE clusters these related concepts into smaller fragments in order that system analysts can decide on more general representative keywords. In order to facilitate specified clustering, a vector-based representation of words, Word2Vec, in combination with K-means algorithm, has been applied. The approach characterizes texts in the file as a high-dimensional vector, thereby analyzing and adapting texts from corpus and assigning a similar vector value for texts with related semantic meanings.

5.5. Questionnaire to Experts for Assessing the Practical Applicability of the Framework

In order to augment the evaluation process for the research study, a mixed approach of qualitative and quantitative methods has been utilized for the verification process. Therefore, in addition to the evaluation approach that utilizes criteria for selecting RE methods, as discussed in Section 4.2, case study evaluation has been applied for verification of the output of the case study to reach triangulation.

Accordingly, the result of the case study has been demonstrated to the experts who have already participated in the evaluation process of this study (Section 4.1). In order to assess the experts view, the questionnaires depicted in Appendix D have been provided to experts. Based on the opinion of the experts that participated in the assessment, it was indicated that 93% of gurus have strongly agreed on the applicability of CASCARE in real projects. In addition, 90% of the collected requirement concepts reflect the pharmaceutical domain. Moreover, 86% of the feedback of the participants revealed that implementation of proposed ideas facilitated the generation of requirements that represent concepts of related services and innovative ideas. Furthermore, 82% of participants agreed that the output of the research can complement the task of requirements prioritization.

6. Discussion

In this study, we have developed a conceptual framework, CASCARE, for the purpose of enhancing the RE process. Evaluation of the framework has been conducted in Section 4 and corresponding key findings were also pin pointed. Consequently, enhancement of CASCARE has been exhibited in RE tasks, such as completeness, recommendation, prioritization, and additional innovative features compared to utilizing individual approaches. Then, in Section 5, a case study has been conducted for the purpose of demonstrating the practical applicability of the proposed framework. Accordingly, discussion and recommendation will be given by reconsidering the corresponding research questions in the paper.

6.1. What Will Be the Components of the CASCARE Framework? And What Are the Complementary Relationships That Exist between Agents and Services, Specifically in the Context of Software RE? (RQ1)

To answer this research question in terms of identifying components of CASCARE, target participants have been identified from academic institutions and enterprises as elaborated in Section 3.2. Subsequently, survey questions have been designed by investigating issues, such as purpose of the research, boundary of the research, research questions, contribution of the research, as well as detailed literature review.

Consequently, major components that are discovered and incorporated in the logical model are: Specify Goals, Assess Features of Services, Post Issue, Requirement Feedback Preprocessing, Requirement Feedback Filter, Requirement Recommender, Prioritize Requirement, and Requirement Knowledge Base.

Accordingly, in terms of complementary relationship, the following issues have been identified:

- **Common platform:** As it has been discussed in Section 3.4, the main concept of CASCARE is to gather requirements related specification for an application with both agents and services engaged through a common SNS platform. Prominent SNS platforms have been also discussed by Robinson [35] and Munante et al. [36].

- **Responsibility:** At an abstract level, the agent concept is responsible for representing elements of the system that makes a decision in the case of possible interaction with other system elements. Moreover, it facilitates users' interaction with each other in terms of goals that the organization anticipates to achieve. Related agent responsibility issues have been also discussed in [17,37–39]. On the other hand, the service concept is responsible for reusing key features from associated services-based applications, which are distributed on SNS platforms. Correlated service role issues have been also discussed in [40–42].
- **Complementary relationships:** The reusability feature of the service paradigm provides better alignment between business requirements and SW products. To this end, at a higher level, service protocols check the interface of service descriptions from RE perspective, but they may not have the capability to update themselves dynamically. To fill this limitation, the agent wraps up the specified service functionality in a way that becomes adaptable. Moreover, the service approach makes use of agent characteristics, since, when some feature of a service is reused, the organizational goal is taken into consideration. This idea has been also reflected by the work of Verlaine et al. [43]. On the other hand, the agent approach also gains some characteristics of services since the decision making process to recommend key requirements concept is carried out by taking into consideration reused service interactions and business rules.

6.2. Case Study for Demonstration of Core Parts in the Proposed Framework (RQ1)

Empirical study has demonstrated that the combined impact of services and agents can significantly enhance the practice of software RE. Organizations can improve the efficiency, effectiveness, and overall quality of the RE process by utilizing service-oriented architectures (SOA) and intelligent agents. One specific case study has focused on employing intelligent agents to facilitate stakeholder collaboration and communication during requirement elicitation. Acting as intermediaries, these agents have gathered stakeholder input and preferences from SNS, streamlining consensus-building and decision-making processes. This approach effectively reduced misunderstandings, increased stakeholder satisfaction, and ensured accurate requirement capture. Moreover, the case study has explored the application of SOA to support requirement modularization and reuse. Organizations can achieve greater flexibility, maintainability, and scalability by breaking down requirements into more minor, independent services. SOA enables organizations to leverage existing services and components, reducing redundancy and development effort.

Furthermore, combining intelligent agents and services can significantly enhance traceability and impact analysis. Intelligent agents can automate requirement tracking and establish relationships with other artifacts, such as design documents and test cases. Services can provide real-time access to requirement related information, empowering stakeholders to assess the impact of changes and make well-informed decisions. In summary, the case study highlights the potential benefits of integrating agents and services in SNS-based software RE. By leveraging intelligent agents for collaboration and communication and adopting SOA for modularization and reuse, organizations can substantially improve the effectiveness and efficiency of the RE process, ultimately leading to the development of higher-quality software products.

6.3. How Can Intelligence of Users in Social Media Assist the Task of RE Processes? (RQ2)

As discussed in Section 3.4 and as demonstrated in the case study in Section 5, user's intelligence in SNS improves tasks of core RE features which are mentioned below. The SNS exhibited such a vital role in collaboration with synergy of agents and services.

In terms of completeness feature, it has been already demonstrated in Section 4 how the proposed approach has enhanced this feature compared to individual approaches. We have also observed related improvement in our case study analysis. For instance, in our case study, 250 user comments regarding requirements have been gathered from several participants and analyzed. As a result, 1400 keywords were generated. This number of

requirements and keywords, which is gathered within 2 weeks, is a promising factor for the proposed approach, which it is capable of gathering almost complete requirements within the specified period.

In addition, we have examined the frequency of requirements related concepts and keywords in order to decide on their priority. This is facilitated by the keyword rank module as has been demonstrated in the case study. In addition, this facilitates the RE process of the proposed method in terms of prioritization. Moreover, the recommendation agent utilizes these frequently discussed and identified keywords and concepts, in the alignment of user requirements and business requirements.

Furthermore, besides the identification of core features in our pharmaceutical case study, the clustering module helps to pin point an innovative idea, such as the users interest in the application that it has to support “Amharic” local language, filling prescriptions through barcodes, detecting whether a drug is counterfeit or not from users smart phone, ability of the application to remotely order drug while the customer is at the hospital or at home, and receive corresponding physical items by physical agents.

6.4. How Can an Evaluation Scheme for the Proposed Framework Be Carried Out? (RQ3)

In order to undertake the evaluation process, we have first customized and organized 21 evaluation criteria to facilitate the comparison between the specified methodologies, thereby demonstrating the impact of the integrated approach over individual approaches. After the criteria have been identified, we have consulted senior system analysts and the corresponding weight is assigned as shown in Table 3, depending on the significance impact of each criterion.

Consequently, after participants have filled in their response concerning the evaluation criteria, the response has been compiled and findings have been summarized in Table 3. In particular, the specified table recaps the results based on participants’ conclusions about how the three approaches strongly, partially, or less likely react to the evaluation criteria. Two distinct expert groups, each with 12 participants, have been formed based on their previous experience in AORE and SORE, respectively in order to perform the evaluation.

Participants in both groups stated that CASCARE has performed better than individual approaches. In particular, those experts in the first group have reported that CASCARE with a score of 93.7% is found to be better than that of AORE with a score of 88.7%, and experts in the second group have declared that CASCARE with a score of 92.3% is better than SORE with a score of 83.7%.

Regarding the evaluation criteria “Assist gathering of complete requirements”, both AORE and CASCARE have similar scores. The reason for the same value could be that both approaches have the “check all goals accomplishment” role in their respective models. On the other hand, CASRE performed better than SORE concerning the specified criteria. In terms of the criteria “Support detection of innovative idea”, CASCARE’s enhancement has been observed with respect to SORE and AORE. The reason for this improvement in both cases is that, in addition to the collaboration of agents and services, the crowd competence in SNS has supported the detection of creative requirements for the intended software system.

In relation to the evaluation criteria “assist prioritization of requirements”, both CASCARE and AORE have exhibited equal performance. However, both have outclassed SORE and both CASCARE and AORE examine the frequency of requirements related concepts to decide on their priority. This is facilitated by a dedicated agent that comprises the keyword rank module in their respective framework. With regard to the “facilitate recommendation” evaluation criteria, both CASCARE and AORE demonstrate equal performance. However, both have surpassed the SORE method. The expert’s viewpoint could be that both support intelligent software components in the form of dedicated agents which assist the RE task.

In terms of “providing tool support” criteria, SORE has performed better than both AORE and CASCARE. The experts’ outlook could be that a tool for CASCARE is under

construction. Moreover, compared to AORE-based tools, SORE exploits various types of industry standard UML tools.

In addition to the above evaluation scheme, experts have also evaluated the research after attending the demonstration concerning the result of the case study. Consequently, experts have strongly agreed on the applicability of CASCARE in real projects. Furthermore, the majority of the collected requirement concepts reflect the pharmaceutical domain. In addition, participants have agreed that the output of the research can complement the task of requirements prioritization, innovation, and reusability.

6.5. Limitations

The implementation demonstrated in the case study did not consider more than one case study to ensure the practical applicability of the framework. In subsequent research work, this will be demonstrated with another case study in another domain. Moreover, input to the current mini-prototype is possible only using one language. Therefore, there will be an effort to introduce local language in the implementation, in order to support participants elaborate their interest using native local language.

7. Conclusions and Future Work

This research demonstrates how the collaborative impact of agents and services on top of SNS better support RE for SW projects, compared to individual approaches. Subsequently, research questions were devised. To address the questions, a detailed literature review has been conducted, and then assessments about the anticipated framework have been gathered from a group of experts. Accordingly, a high-level framework has been designed. For the actualization of the evaluation, a customized list of evaluation criteria for SW RE methodologies has been prepared.

After performing the necessary group-based assessment based on the specified criteria, those experts in the first group have reported that CASCARE with 93.7% result is found to be better than that of AORE with 88.7% outcome. Moreover, experts in the second group have declared that CASCARE with 92.3% result is better than SORE with 83.7% outcome. In both cases, improvements have been observed, which reveals that the synergy of CASCARE has a better impact on the RE process than that of utilizing individual approaches.

As a result, to demonstrate the practical applicability of the framework, we have practically demonstrated a case study. The experimental demonstration comprises tasks, such as extracting keywords from users' feedback, prioritizing and clustering the specified keywords based on their semantic similarity. It has been also observed that the proposed approach is suitable for a data-driven development approach, where the role of end users is crucial. Consequently, the case study demonstrates the capability of the proposed approach in terms of generating innovative ideas from multiple users, facilitating priority, assisting completeness feature, and recommendation of related requirements.

There is an intention to expand this research in the future with perspectives, such as supporting the implementation with local language features, deploying the framework concept in another case study for better visualization, incorporating and testing change management and traceability characteristics of requirements. Moreover, investigating issues for developing a comprehensive tool and demonstrating the tool on a different case study indifferent domain are planned future endeavors.

Author Contributions: Conceptualization, M.S. and D.M.; methodology, M.S. and D.M.; software, M.S.; validation, M.S., T.M.G. and A.L.; formal analysis, M.S., T.M.G. and A.L.; investigation, D.M.; resources, A.L.; data curation, M.S.; writing—original draft preparation, M.S. and A.L.; writing—review and editing, M.S., A.L. and D.M.; visualization, M.S., T.M.G. and A.L.; supervision, D.M.; project administration, T.M.G.; funding acquisition, M.S., T.M.G. and A.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset for the case study and associated code fragment is available at https://github.com/mekugit/pharma_dataset, accessed on 15 June 2023.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Questionnaire Filled for Construction of Conceptual Framework

Table A1. Questionnaires provided to a group of experts for conceptual framework design.

S.No	Survey Question	Remark
1	Utilizing CASCARE enables the crowd to give complete requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree	
2	Applying CASCARE is difficult to organize and coordinate for eliciting the right requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree	
3	Applying CASCARE makes it hard to reach a consensus/agreement on requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree	
4	Applying CASCARE increases meaningfulness of requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree	
5	By applying CASCARE, the credibility of the elicited information cannot be guaranteed 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree	
6	By applying CASCARE, the crowd competence supports obtaining the right requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree	
7	In using CASCARE, the crowd competence supports detecting and gathering creative requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree	
8	Using CASCARE enables us to gather more relevant requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree	
9	Applying CASCARE makes the requirements engineering process difficult to manage 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree	
10	Collaboration obtained from CASCARE helps requirement engineers in applying validation process of the elicited requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree	
11	CASCARE supports obtaining more adaptive requirements to upcoming related applications or new version of current application 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree	
12	CASCARE might provide a chance for malicious users to enter invalid input and affect the overall quality of elicited requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree	
13	CASCARE enables us to capture tacit knowledge about requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree	
14	CASCARE can provide the function for the notification of new requirement changing which is not yet specified 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree	
15	CASCARE can provide the possibility for conducting various types of discussions with customers regarding requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree	
16	CASCARE enables us to gather requirements information timely 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree	
17	The proposed approach, CASCARE, can have an impact on reusing requirement from related previous systems 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree	
18	CASCARE can help us in prioritizing requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree	

Table A1. Cont.

S.No	Survey Question					Remark
19	The proposed approach, CASCARE, enables us to detect a saturation point regarding the elicited requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree					
20	The proposed approach, CASCARE, can facilitate change management tasks related to requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree					
21	CASCARE can help us in predicting or recommending requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree					
22	CASCARE can help us in keeping track of consistency 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree					
23	The proposed approach, CASCARE, can have an impact in order to better understand customer needs 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree					
24	CASCARE can help us in identifying redundant requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree					
25	Using CASCARE, the requirement gathering process may not be accomplished on time 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree					
26	CASCARE can be adopted for different case studies 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree					
27	In applying CASCARE, since we gather requirements from large-size stakeholders, it may be difficult to refine requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree					
28	CASCARE can be applicable both for centralized and distributed application development 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree					
29	By applying CASCARE, it is possible to detect conflicting requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree					
30	To ensure the reliability, comparison of CASCARE must be made with respect to the traditional approaches 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree					

Appendix B. Additional Use Cases and Corresponding Descriptions for the Case Study

Table A2. Use cases and brief descriptions for major functionalities.

S.N	Use Case Name	Description	Primary Actor
1	Return Drug	Used to capture returned drug, in the case that there is a defect on the medicine and the customer is unwilling.	User, Customer
2	Display Category	Used to categorize drugs, for instance, narcotic and antibiotic. Moreover, prescribed and non-prescribed drug items.	User, Medical Consultant
3	Receive Feedback	Used to gather requests from customers in relation to drug.	Customer, User
4	Send Response	Helps to send suggestions to customer's request.	Medical Consultant
5	View Order	Used to observe users order.	User
6	Generate Report	Used to prepare various types of reports.	User
7	Add to cart	Helps to let customers add multiple items selected for purchase.	Customer, User
8	Send SMS Notification	Used to send information to customers, such as drug interaction and new arrivals.	Customer, Medical Consultant
9	Calculate Drug Dose	It suggests the amount of drug proposed per age and weight.	Medical Consultant
10	Alert Minimum Reorder	Alerts the pharmaceutical owner, in order to purchase and import drugs before the specified item has vanished.	User

Table A2. *Cont.*

S.N	Use Case Name	Description	Primary Actor
11	Filter Medical item	Used to identify selected items based on parameters.	User
12	Record drug Provider information	Helps to preserve provider status for frequent communication and to retain quality of service.	User, Sys Admin
13	Register Invoice	Used to capture transaction data concerning finance.	Accountant, User
14	Search Drug Items	Used to find target drug based on specific criteria.	User
15	Print inventory list	Used to send information to auditors and drug control administration offices.	User, Medical Consultant
16	Receive E-Signature	Used to capture a signature from the physician, in the case that the signature on the prescription is not visible.	User, Sys Admin
17	Link with Hospitals	Helps to capture information related to drug usage in hospitals, thereby assessing the demand and supply.	User, Medical Consultant
18	Link with SNS	Assists the process of capturing and receiving customers' suggestion from social network sites.	User, Sys Admin
19	Detect Frequently requested Drug	Helps to capture information about frequently requested items in order that it will have a role during purchase decision.	User, Medical Consultant
20	Register and Edit Items	Used to record and update medical items.	User, Sys Admin
21	Update Items Price	Helps to automatically updatethe price of medical items.	Accountant, User
22	Fill Insurance Form	Used to connect customer fee with insurance company.	Accountant, User
23	Filter Expired Drugs for Destruction	Helps to eradicate expired medical items.	User, Sys Admin
24	Show Sales Transaction by Date	Displays transaction by date and sendsa report to concerned body.	Accountant, User
25	Register Drug Consumption Approach	Registers drug consumption way such as by oral or injection.	User, Medical Consultant
26	Display Storage Location	Helps to show the location of each medical item.	User
27	Link with Provider	To make an online connection with provider order entry.	User
28	Attach Drug Image	Used to associate an image to medical items.	User, Consultant

Appendix C. Customized Evaluation Criteria Form for Comparing RE Methodologies

Separate form is filled for each of the three cases: AORE, SORE, and CASCRE.

Each person (expert) fills all 21 criteria with (H or M or L) whose numeric value is (3, 2, 1), respectively. And (H = "highly supportive", M = "medium supportive", L = "less supportive").

Table A3. Customized evaluation criteria form for comparing RE methodologies.

[illegible]

Table A3. Cont.

S.N	Evaluation Criteria	Cr. Weight (%)	Evaluation for CASCRES											
			Person											
			P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
5	Assist identification of reusable requirements	5												
6	Assist identification of ambiguous requirements	4												
7	Assist identification of stakeholders	5												
8	Facilitate communication and negotiation	6												
9	Assist understanding of social issues	4												
10	Assist gathering of implicit users interest	5												
11	Assist modeling of requirements	5												
12	Assist detection of non-functional requirements	5												
13	Support decomposition of complex requirements	6												
14	Assist prioritization of requirements	5												
15	Practical applicability of the method	4												
16	Facilitating good learning curve	5												
17	Facilitating recommendation feature	4												
18	Facilitates verification	4												
19	Providing tool support	5												
20	Integration with other methodologies	4												
21	Gathering requirements from distributed teams	4												

Appendix D. Questionnaire Provided to Experts for Assessing the Result of the Case Study

Table A4. Questionnaire provided to experts for assessing the result of the case study.

S.No	Survey Question	Remark				
1	The concept of CASCRES can be practically applicable 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree					
2	The case study has demonstrated that CASCRES can gather adequate requirements and supplement previous methods 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree					
3	Applying CASCRES makes it hard to reach a consensus (agreement) on requirements 5: Strongly Agree 4: Agree 3: Undecided 2: Disagree 1. Strongly Disagree					

Table A4. Cont.

S.No	Survey Question					Remark
4	The case study demonstrated that it is possible to detect and gather creative requirements	5: Strongly Agree	4: Agree	3: Undecided	2: Disagree	1: Strongly Disagree
5	The case study has demonstrated that it is possible to reuse existing service dataset	5: Strongly Agree	4: Agree	3: Undecided	2: Disagree	1: Strongly Disagree
6	The demonstration showed that the crowd competence supports obtaining the right requirements	5: Strongly Agree	4: Agree	3: Undecided	2: Disagree	1: Strongly Disagree
7	The demonstration showed that CASCARE supports prioritization of requirements	5: Strongly Agree	4: Agree	3: Undecided	2: Disagree	1: Strongly Disagree
8	The case study revealed that CASCARE enables us together more relevant requirements	5: Strongly Agree	4: Agree	3: Undecided	2: Disagree	1: Strongly Disagree
9	CASCARE makes the requirements engineering process easier to manage by clustering keywords	5: Strongly Agree	4: Agree	3: Undecided	2: Disagree	1: Strongly Disagree
10	The demonstration revealed that CASCARE makes the RE process difficult to manage	5: Strongly Agree	4: Agree	3: Undecided	2: Disagree	1: Strongly Disagree
11	The demonstration showed that CASCARE cannot support prioritization of requirements	5: Strongly Agree	4: Agree	3: Undecided	2: Disagree	1: Strongly Disagree
12	The case study has demonstrated that it is difficult to reuse existing service dataset	5: Strongly Agree	4: Agree	3: Undecided	2: Disagree	1: Strongly Disagree

References

- Standish Chaos Report Group. Major Causes of Software Failure. Available online: www.ukessays.com (accessed on 21 June 2022).
- Kingsman, M. 68 Percent of IT Projects Fail. 2012. Available online: <https://www.zdnet.com/article/study-68-percent-of-it-projects-fail/> (accessed on 12 October 2022).
- Mike, P.; Van den Heuvel, W.-J. Service oriented architectures: Approaches, technologies and research issues. *VLDB J.* **2010**, *16*, 389–415.
- Engels, G.; Hess, A.; Humm, B.; Juwig, O.; Lohmann, M.; Richter, J.-P.; Willkomm, J. Service-Oriented Architecture: Performance Issues and Approaches. ISBN: 978-3-89864-506-5. Available online: <https://www.cscan.org/openaccess/?id=175> (accessed on 16 July 2021).
- Jennings, N.R.; Sycara, K.; Wooldridge, M. A Roadmap of Agent Research and Development. *Auton. Agents Multi-Agent Syst.* **2009**, *1*, 7–38. [CrossRef]
- Odell, J.; Parunak, H.V.D.; Bauer, B. Extending UML for Agents. In Proceedings of the Agent-Oriented Information Systems (AOIS) Workshop at the 19th National Conference on Artificial Intelligence, Trier, Germany, 27 September–1 October 2010; Available online: <http://www.jamesodell.com/ExtendingUML.pdf> (accessed on 16 July 2021).
- Keating, M.; Furberg, R. A Methodological Framework for Crowd Sourcing in Research. Available online: https://nces.ed.gov/FCSM/pdf/H1_Keating_2013FCSM_AC.pdf (accessed on 7 March 2018).
- Ikram, M.; Ashraf, H. A Review of RE Issues and Challenges. Available online: https://researchgate.net/publication/272863222_A_Review_of_requirements_engineering_issues_and_challenges_in_various_software_development_methods.pdf (accessed on 16 October 2017).
- Chemuturi. More Misunderstanding than Right Understanding about Requirements. Available online: <https://www.springer.com/content/chemuturi2013.pdf> (accessed on 12 July 2018).
- Lim, S.L.; Finkelstein, A. Stakerare: Using social networks and collaborative filtering for large-scale requirements elicitation. *IEEE Trans. Softw. Eng.* **2012**, *38*, 707–735.
- Lohmann, S.; Dietzold, S.; Heim, P.; Heino, N. *A Web Platform for Social Requirements Engineering*, Software Engineering Conference; KöllenDruck Verlag GmbH: Bonn, Germany, 2012; Volume 150, pp. 309–315.
- Jacek, D.; Emmanuel, L.; Anna, M. *Analyzing App Reviews for Software Engineering: A Systematic Literature Review*; Springer: Berlin/Heidelberg, Germany, 2022; Available online: https://www.researchgate.net/publication/357987040_Analysing_app_reviews_for_software_engineering_a_systematic_literature_review (accessed on 14 September 2019).
- Hosseini, M.; Moore, J.; Almaliki, M.; Shahri, A.; Phalp, K.; Ali, R. Wisdom of the Crowd within enterprises: Practices and Challenges. *Comput. Netw.* **2015**, *90*, 121–132. [CrossRef]
- Adepetu, A.; Khaja, A.A.; Al Abd, Y.; Al Zaabi, A.; Svetinovic, D. *CrowdREquire: A Requirements Engineering Crowd-Sourcing Platform*; Wisdom of the Crowd, AAAI Spring Symposium: Stanford, CA, USA, 2013.
- Lam, N.; Sano, H.; Robin, M.E.; Shun, S. A Web Agent Based on Exploratory Event Mining in Social Media. In Proceedings of the 2012 IIAI International Conference on Advanced Applied Informatics, Fukuoka, Japan, 20–22 September 2012; Available online: <https://ieeexplore.ieee.org/abstract/document/6337195> (accessed on 17 May 2018).

16. ElSayed, I.A.; Ezz, Z.; Nasr, E. Goal Modeling Techniques in Requirements-Engineering: A Systematic Literature Review. Available online: <https://thescipub.com/pdf/10.3844/jcssp.2017.430.439> (accessed on 21 March 2022).
17. Cheah, W.; Shafreena, A.; Puad, A.; Hong, P.C. Agent-Oriented Requirement Engineering for Mobile Application Development. *Int. J. Interact. Mob. Technol.* **2015**. [CrossRef]
18. Bano, M.; Ikram, N. Issues and Challenges of RE in Service Oriented Software Development. In Proceedings of the Fifth International Conference on Software Engineering Advances (ICSEA), Nice, France, 22–27 August 2010; pp. 64–69.
19. Seixas, K.; Fantinato, M.; Medeiros, M. A framework for requirements monitoring of service based systems. In Proceedings of the 2nd International Conference on Service Oriented Computing, New York, NY, USA, 15–19 November 2004; Available online: <https://dl.acm.org/doi/abs/10.1145/1035167.1035181> (accessed on 9 March 2019).
20. Mario_Gomez. Available online: https://www.researchgate.net/publication/230625704_Integrating_Knowledge_Modeling_and_Multi-Agent_Systems (accessed on 27 January 2019).
21. Novak, M.; Švogar, I. Current usage of component based principles for developing web applications with frameworks. *Interdiscip. Descr. Complex Syst.* **2016**, *14*, 253–276.
22. Carreño LV, G.; Winbladh, K. Analysis of User Comments: An Approach for Software Requirements Evolution. In Proceedings of the 35th International Conference on Software Engineering, San Francisco, CA, USA, 18–26 May 2013; pp. 582–591.
23. Perini, A.; Susi, A.; Avesani, P. A machine learning approach to software requirements prioritization. *IEEE Trans. Softw. Eng.* **2013**, *39*, 445–461. [CrossRef]
24. Abualhajja, S.; Arora, C.; Sabetzadeh, M.; Briand, L.C.; Vaz, E. A machine learning-based approach for demarcating requirements in textual specifications. In Proceedings of the IEEE International Conference on Requirements Engineering, Jeju, Republic of Korea, 19–24 September 2019; pp. 51–62. [CrossRef]
25. Hevner, Design Science Research Cycles (Hevner, 2007). Available online: https://www.researchgate.net/figure/Design-Science-Research-Cycles-Hevner-2007_fig1_312627356 (accessed on 21 June 2021).
26. Xu, S. Empirical research methods for software engineering. In Proceedings of the 2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA) (2017 IEEE), London, UK, 7–9 June 2017. [CrossRef]
27. Stol, K.J.; Fitzgerald, B. Uncovering theories in software engineering. In Proceedings of the 2013 2nd SEMAT Workshop on a General Theory of Software Engineering (GTSE), San Francisco, CA, USA, 26 May 2013; pp. 5–14. [CrossRef]
28. Wieringa, R.J. *Design Science Methodology for Information Systems and Software Engineering*; Springer: Berlin/Heidelberg, Germany, 2014. [CrossRef]
29. Engström, E.; Storey, M.A.; Runeson, P.; Höst, M.; Baldassarre, M.T. How Software Engineering Research Aligns with Design Science. *Empir. Softw. Eng.* **2020**, *25*, 2630–2660. [CrossRef]
30. Type-Form. Online form Builder. Available online: <https://forms.app/en/templates/research-surveys> (accessed on 3 June 2022).
31. Garcia, E.; Giret, A.; Botti, V. Evaluating software engineering techniques for developing complex systems with multi agent approaches. *Inf. Softw. Technol.* **2011**, *53*, 494–506. [CrossRef]
32. Hickey, A.M.; Davis, A.M. Elicitation technique Selection: How do experts do it? In Proceedings of the 11th IEEE International Requirements Engineering Conference (2010), Monterey Bay, CA, USA, 12 September 2003.
33. Tadist, K.; Najah, S.; Nikolov, N.S.; Mrabti, F.; Zahi, A. Feature selection methods and genomic big data: A systematic review. *J. Big Data* **2019**, *6*, 79. [CrossRef]
34. Ruder, S.; Vulić, I.; Søgaard, A. A Survey of Cross-lingual Word Embedding Models. *J. Artif. Intell. Res.* **2019**, *65*, 569–631. Available online: <https://www.jair.org/index.php/jair/article/view/11640> (accessed on 19 February 2019). [CrossRef]
35. Robinson, R. *The 7 Top Social Media Sites*; IBM Press: Indianapolis, IN, USA, 2020; Available online: <https://www.adobe.com/express/learn/blog/top-social-media-sites> (accessed on 5 January 2020).
36. Munante, D.; Siena, A.; Kifetew, F.M.; Susi, A.; Stade, M.; Seyff, N. Gathering Requirements for Software Configuration from the Crowd. In Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW), Lisbon, Portugal, 4–8 September 2017.
37. Yu, I. Main References in AORE, a Framework that Uses Actors, Beliefs, Commitments, and Goals to Model Organizational Environments. Available online: <http://www.agentlink.org> (accessed on 16 December 2022).
38. Miller, T.; Pedell, S.; Sterling, L.; Lu, B. Engaging Stakeholders with Agent-Oriented-Requirement Modeling. Available online: <https://people.eng.unimelb.edu.au/tmiller/pubs/aose2010a.pdf> (accessed on 7 March 2021).
39. Miller, T.; Lu, B.; Sterling, L. Requirements Elicitation and Specification Using the Agent Paradigm. University of Wollongong Research Online. 2015. Available online: <https://ro.uow.edu.au/cgi/viewcontent.cgi?referer=&httpsredir=1&article=4131&context=eispapers> (accessed on 5 May 2020).
40. Verlaine, B.; Jureta, I.; Faulkner, S. Towards conceptual foundations of requirements engineering for services. In Proceedings of the 5th IEEE International Conference on Research Challenges in Information Science [RCIS], Gosier, France, 19–21 May 2011; IEEE Computer Society: Washington, WA, USA, 2012.
41. Penserini, L.; Perini, A.; Sus, A.; Mylopoulos, J. From stakeholder needs to service requirements. In Proceedings of the Service-Oriented Computing: Consequences for Engineering Requirements, Minneapolis, MN, USA, 12 September 2009.

42. Bano, M.; Zowghi, D. Service Oriented Requirements Engineering: Practitioner's Perspective. Available online: <https://www.researchgate.net/publication/260290632> (accessed on 6 October 2018).
43. Dheepa, V.; Vijayalakshmi, C.; Naganathan, E. A Novel Method for Large Scale Requirement Elicitation. Available online: https://iraj.in/journal/journal_file/journal_pdf/3-18-13908227667-12.pdf (accessed on 12 April 2021).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.