Requirement Engineering Challenges for Social Sector Software Development: Insights from Multiple Case Studies

VARUN GUPTA, Centro de Competências em Cloud Computing (C4-UBI), Universidade da Beira Interior, Portugal

Diversity is a great challenge for software engineers in the social sector context. The objective of this paper is to contribute to the identification of the RE processes and associated challenges in releasing the software in the social sector markets for which an exploratory case study is conducted. The outcome of the case study indicates that the diversity limits the ability to involve the representative samples of user populations using the same set of RE tools and techniques as one size fits all solution for all segments. The diverse user base must be partitioned into different segments, with each segment triggered using a suitable set of RE techniques i.e., traditional and crowd-based RE. The diverse perspectives learned as a result of the interaction with each segment, must be merged together into a single perspective about the software meant to be used in the social sector. There is a need for a new RE process specially designed for handling the complexities of the social sector, which this paper terms as **Social Sector Requirement Engineering (SSRE)**. There is an increased need for collaboration between government social sector institutions and software engineers to get access to diverse customers to improve software quality.

CCS Concepts: • **Software and its engineering** → **Requirements analysis**;

Additional Key Words and Phrases: Social Sector Requirement Engineering, social sector, Requirement Engineering

ACM Reference format:

Varun Gupta. 2021. Requirement Engineering Challenges for Social Sector Software Development: Insights from Multiple Case Studies. *Digit. Gov.: Res. Pract.* 2, 4, Article 34 (December 2021), 13 pages.
<https://doi.org/10.1145/3479982>

1 INTRODUCTION

Mahiti Infotech Pvt. Ltd., a firm based in India with a focus on technological social innovation, implemented a project called “OurCrop” in the year 2012. OurCrop is a Free and Open Source Software for the agricultural institutes to manage farming activities. This social innovation project imposed several challenges to the firm, the major ones included accurately exploring the problem domain, translating understandings into software requirements, and ensuring the continuous active user engagement throughout the software development

This work was supported by Operação Centro-01-0145-FEDER-000019 – C4 – Centro de Competências em Cloud Computing, co-financed by the Programa Operacional Regional do Centro (CENTRO 2020), through the Sistema de Apoio à Investigação Científica e Tecnológica – Programas Integrados de IC&DT.

Author’s address: V. Gupta, Centro de Competências em Cloud Computing (C4-UBI), Universidade da Beira Interior, Estrada Municipal 506, 6200-284 Covilhã, Portugal; email: varun.iit13@gmail.com.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2021 Copyright held by the owner/author(s).

2639-0175/2021/12-ART34

<https://doi.org/10.1145/3479982>

process [1]. These difficulties arose because the main stakeholders of the system were the marginalised segments of the society (i.e. farmers), which were hard to be involved in for knowledge transfer with the **Requirement Engineering (RE)** team. There are many other social sector applications which are used by marginalised as well as technology savvy citizens (for instance, *Seguranca Social Direta* in Portugal), the diversity which makes RE much harder to execute.

RE is the software engineering sub activity that tries to identify the real needs of the customers which are then documented as requirements after a careful analysis, prioritization, and validation. The main reason for project failures in big and small companies (like startups) is the inability to satisfy its intended users i.e., delivering the features that are not required by the users [2–4]. This means that the requirement team must interact with the users to identify their needs and validate the set of requirements with them before starting to implement them into working software. User involvement is thus a necessary condition for project success [2, 5–6].

Traditional RE includes the activities that require users to be in close contact with the requirement team at the same place and same time, with the focus on a limited number of users [6]. The **crowdsourcing based RE** (called **Crowd based RE**) [7–9] and the Gamification versions of RE (for fostering motivation levels for continuous user participation [2, 10–14]) require the users to have skills to access the crowdsourcing infrastructure.

The social sector market is made up of the citizens as the users, which are too diverse in terms of their demographics, expectations, habits, technological experience, and so on. Further, they are distributed in far flung areas of the country. This diversity limits the applicability of traditional RE, as face to face participation is not feasible with diverse users. The crowd based RE, and gamification based RE are also limited, as it is hard to involve the citizens with no or little technological expertise on online platforms. This limits the ability of the RE team to make RE activity participatory by involving diverse users, which lowers the chances of the social software to be successful in the market. The software solution thus fails to meet society needs and fails to deliver social goods (benefits to the society).

One major aspect in reaching out to the users for exploring the problem domain is how well the requirement team understands the market of the product (in other words, possible users), and how easily they can manage to establish communications with them? To make accessibility easier, the role of government is a prime factor in fostering the knowledge transfer between citizens and the requirement team. This is because the government has complete details about the possible users of the social sector applications, personal details about these users (for instance, data available with social institutions as provided by citizens themselves), and resources to motivate them to participate in RE activities. The success of social sector applications depends on the ability to meet product/market fit, which is possible only when the citizens are accessible to the requirement team. The government processes are ICT enabled which allows them to store, process and utilize the big data about the customers. The strong focus of the government to provide better services to the citizens by reducing bureaucracy using digital tools (for instance, blockchain based system for land registry) is a strong motivator for investing more in developing the software applications for the social sector of the nation. This allows them to show their focus on promoting transparency, auditability, accountability, and effortless and fast services. However, the adoption of these social sector applications by the citizens is probabilistic as not only they need to be convinced to use such systems but also the system should be able to match their needs. Government could act as a bridge between the requirement team and citizens rather than acting as the customer representative.

The objective of this paper is to investigate the various challenges experienced by the social sector software developing companies and to study their RE practices. This objective is fulfilled through an exploratory case study conducted with the two Multinational Software companies located in India and the USA. Exploratory study is conducted because RE in the social sector context is an unexplored issue in the literature. The studied companies have multiple subsidiaries located globally and are involved in delivering software products for the mass markets, including social sectors. (For Instance, company A product portfolio has 50% social sector product contribution and company B product portfolio has 28% social sector software products).

The study aims to answer the following research questions:

RQ1. How is RE carried out in the context of social sector applications?

RQ2. What are the challenges in doing RE in the context of social sector applications?

This paper is structured as follows: the brief background of requirement engineering techniques applicable in the social sector context is presented in Section 2 followed by case study research protocol in Section 3. Section 4 highlights the result of the case study which is further discussed through a real user case (Section 5.1), framework for future research (Section 5.2), result assessment (Section 5.3), and implications to government and software engineers (Section 5.4). Finally, the paper is concluded along with directions for future work in Section 6.

2 BACKGROUND

Gamification and crowdsourcing had been widely employed in software engineering activities (especially **Requirement Engineering (RE)**) to leverage the power of diverse software related knowledge distributed amongst a crowd of motivated stakeholders. These technologies require technological platforms (for instance, social networks) to actively involve crowds for undertaking outsourced software engineering tasks. In the context of RE, crowdsourcing helps the requirement analysts to involve the globally distributed stakeholders (especially the users) to provide their feedback (or requirements). The requirements provided by the stakeholders are then prioritized by them through their social interactions in the form of voting, comments, etc. [16–17]. Crowdsourcing platforms help to reach a large number of users, which are globally distributed and hence not approachable through face-to-face interactions at the same physical space. Their motivation levels are boosted using gamification.

The literature provides good references to the work already reported in the domain of crowdsourcing based RE [18–30]. The overall analysis of the published literature suggests that the crowdsourcing had been applied in generic settings without being focused on specific application domains (for instance, social sector or public administration). The crowd based RE had been witnessing the challenges like ability of the crowdsourcing platform to integrate feedback from different feedback channels (for instance, combining user feedbacks across App stores and social networks), efficiently processing the highly unstructured and noisy user feedbacks expressed in natural languages, ability to trust the feedback provider without compromising his privacy in crowdsourcing platforms (and filtering out those expressed by illegitimate users), and aggregating the feedback responses (collective intelligence) of the providers to support RE decisions. Building over these limitations, the applicability of the crowdsourcing techniques for RE in the social sector application domain is still in its infancy stage.

The literature has limited studies pertaining to requirement engineering in the social sector context. The limited studies do address the requirement engineering in the social sector context, but they do not provide rigorous solutions that address the unique challenges of social software applications.

Authors in [31] proposed Requirements Bazaar, a browser-based social software platform for **Social Requirements Engineering (SRE)** that allows community members to have idea generation, idea selection and realisation collaboratively. The authors in [32] provided an opinion about participation of the user masses in requirement engineering activities using delegated voting and structured refinement. Authors in [33] provided an opinion about interrelation between requirement engineering and society; with requirement engineering uncovering the needs of society. For instance, it is important to analyse the unique needs of the elder people using social media, which will facilitate the technology design decisions aligned with their needs. An author in [34] applied motivational modelling to develop a live system called Ask Izzy, a system that assists homeless Australians to identify information about the services that they need. The team was successful in understanding the needs of the special user group—those facing homelessness and conducting requirement engineering activities by overcoming the socio-technical gap. The outcome has good lessons for the requirement engineers that are involved in designing social sector solutions by involving diverse users.

Table 1. Details of the Social Sector Related Studies

Focus	Research Type	Suitability for Social Sector				Limitation	Study Reference
		Diversity of User	Geographical Access	Computational Intelligence from diverse perspectives	Non Functional Aspects like Scalability, Ease of Use etc.		
Collaborative mass user participation in Requirement Engineering.	New working Method	No.	Yes.	No.	No.	Key issues remain unaddressed: 1. How can diverse users (especially less technology savvy users) can be involved in requirement engineering? 2. How the perspectives of different groups will be merged (if different mechanisms are used to reach different user groups).	[16]
	Vision	No.	No.	No.	No.		[17]
Requirement Engineering and Society.	Vision	No.	No.	No.	No.		[33]
Motivational modelling to reduce socio-technical gap.	New Working method	No.	No.	No.	No.		[34]

Table 1 gives the focus and limitations of the studies with respect to their applicability for social sector requirement engineering i.e., requirement engineering for the software applications that are intended to be used by diverse citizens of the country.

Table 1 highlights that there is a need to better understand the challenges faced by requirement analysts in exploring the problem domain of social sector applications. The understanding will help drive the research addressing the social challenges in requirement engineering in the future.

3 CASE STUDY PROTOCOL

The case study is executed as per the guidelines proposed in [15]. This involves five steps i.e., case study design, data collection procedures, collecting evidence, analysis of collected data, and reporting. These steps are executed considering the validity and ethical issues.

3.1 Research Design

The case study is Embedded Multiple case study which studies the RE practices (**units of analysis**) in the software companies (**cases**) involved in the software development for the social sector (**context**).

The data was collected in two phases, namely:

- Direct interviews conducted using video conferencing, and
- Observation of the documents shared by the companies.

The insights shared in the first phase are elaborated in the second phase to address doubts and elaborate abstract details, and address new perspectives that emerge after analysis of the collected data. The details of the interviewed companies and their representatives are given in Section 3.2. The collected data was transcribed verbatim and subjected to the grounded theory for necessary analysis.

To ensure the validity of the results, the four types of threats i.e., construct validity, internal validity, external validity, and reliability, are addressed while designing and executing the case study protocol. The collected data after the end of interviews of two phases and final results were shared with the case study participants to ensure that their perspectives are accurately captured thereby addressing construct validity. The case study is

Table 2. Case Details

S. No.	Company Name	Main Location of cases	Other Locations	Software Products
1.	A	India	Global	Large Portfolio
2.	B	USA	Global	Large Portfolio

exploratory in nature so internal validity is not a threat. The use of multiple levels of data collections through two methods i.e., interviews and archival analysis and providing the chain of evidence addresses reliability.

The data shared by the company representatives comes from their wide experience in developing social sector applications, which could be meaningful for other companies as well. As the companies are diverse in terms of their working context and resources, the findings could not be very interesting in those marginal cases and hence external validity is marginally impacted.

3.2 Case Details

The cases in this case study were selected because of the following reasons:

- (a) the companies had been in the market from past many years (and thus have a consortium of product release experiences as their strategic assets).
- (b) they have a trace record of successful market software projects.
- (c) they have been successful in delivering software in the social sector (as a part of their social corporate responsibility and for product diversifications in social sector markets).

These companies have multiple subsidiaries located globally and are involved in delivering software products for the mass markets, including social sectors. For instance, company A product portfolio has 50% social sector product contribution, and the percentage is 28% for the company B. Table 2 gives the characteristics of such companies without revealing their identity. The company representatives agreed to participate on the condition of anonymity, so company names are replaced with A and B.

The case study involved two representatives from each company. The representatives were senior managers who had diverse experience in leading the social sector project commercialization. Their participation brought diverse perspectives driven by their experiences with different projects exhibiting different roles.

Table 3 gives the brief information about the social sector projects undertaken by the studied companies. The number of social sector projects implemented by the companies counts each incremental version of the software delivered to the society as a single individual project.

4 RESULT ANALYSIS

The data collected from the cases are subjected to analysis to generate answers to the formulated research questions. The RE challenges specific to the social sector application are individually mentioned under individual research questions below:

RQ1. How is RE carried out in the context of social sector applications?

The software companies usually undertake bespoke (single client like NGO or the Government) and/or mass market software development (wider market) of the social sector software. Both studied companies perform the RE activities in flexible, light weight processes, jointly with the users. The users are highly diverse in terms of their technical expertise, language, ages, geographical locations, etc. This limits the involvement of the crowd of users throughout the RE activities. Further, the user segments are beyond the reach of the organisation, as they could be globally dispersed. For example, one of the respondents from company A said, “*Social sector application may be intended to be offering services to the senior citizens. Their health issues, large ages, technical expertise etc.*”

Table 3. Project Details of Studied Companies

S. No.	Company Name	End Users	Problem domain targeted	Total Projects in portfolio	Number of Social Sector Projects	% Social Sector projects
1.	A	Social Sector Institutions	Social Media analytics software. Ideas evaluation. Automation of activities. Tax management.	68	20	50%
		Citizens	Child welfare. Medical support to poor families. Education. Woman entrepreneurship		14	
2.	B	Social Sector Institutions	Automation of activities. Funding request applications. Audit reports. Funding management. Crowdfunding campaign management. Scheme social Impact analytics. Tax management.	142	32	28%
		Citizens	Education Homeless people management. Sustainability (environment).		08	

Table 4. RE Activities

S. No.	RE Activity	Process Involved
1.	Requirement Gathering	Informal Discussions and Observations.
2.	Requirement Analysis	Manually by requirement analysts.
3.	Requirement Prioritization	By team of requirement analysts and product manager on the basis of their gut feeling (understanding of the problem domain).
4.	Requirement Validation	Through prototypes (animations) which are validated with the sample of users.
5.	Requirement Documentation	Done informally using computer notes or sticky notes on the wall.
6.	Requirement Evolution	Discussion and observation of the set of users who requested for the changes in the requirements (by expressing their problems).

limits their continuous involvement throughout RE. However, the best we could do is to interact to the maximum with them to convert the interaction into the knowledge that drives the engineering process”.

The users specify the changes or make new requests by specifying the problems they are facing (i.e. the pain points). The RE team prefers to observe and interact with the users to enhance the understanding of the problem domain. This understanding helps them to map the problems into software requirements, which are validated in an informal discussion with the users. As per one of the respondents from company B, *“Due to diversity in the users and lack of financial resources for social projects, we need to strike a balance between value to be offered and the efforts to be invested for such offerings. That is the reason why we execute RE activities as light weight processes driven by accurate & validated understanding of the customer problem domain”.*

The manner the RE activities are conducted by the companies, are mentioned in Table 4.

The level of involvement of the users for different RE activity varies as shown in Figure 1 (number of users employed vs individual RE activities). The graphical representation is drawn based on the average project data of the two companies (14 projects implemented by company A and 8 projects implemented by company B).

Fig. 1. Number of users for individual RE activities.

Fig. 2. Variation of User Involvement throughout RE for both companies.

Figure 2 is another equivalent representation of Figure 1, which graphically shows the variation of the user involvement for each RE activity, for the studied companies only. In Figure 2, the Y axis denotes the number of users employed for different RE phases, represented by the X axis.

RQ2. What are the challenges in doing RE in the context of social sector applications?

RE is the co-creation process that must involve users throughout the whole process. However, the RE team must cope up with numerous issues while executing the RE activity as mentioned below:

- (a) **Complex and unexplored problem domain:** The problem domain is unclear initially. This is because the end users (citizens) do not know exactly their needs. Finding access to good representatives of these end users (because many may not be known initially) and making them agree to participate in RE, is a greater challenge for the requirement team. Understanding the social problems is a harder exercise, the expertise which companies lack otherwise (for instance, social science researchers). The support of secondary data is minimal as they could be less accurate and may not fulfil the purpose. The companies collect user pain points from the NGOs and the volunteers, to get an understanding of the problem domain.
- (b) **Diversity of Users:** User Diversity is the source of diverse perspectives about the solution, provided that establishing crowdsourcing infrastructure is possible. This is however a major challenge in the case of the social sector. This is because social sector applications must deliver benefits to all eligible sections of the society irrespective of any bias. Minority segments have equal representation in the solution and the overall success of the system. The diverse segments are further harder to be involved in RE activities both in the company premises or through crowdsourcing platforms. For instance, crowdsourcing will not be feasible with less technology savvy people especially the uneducated, senior citizens, and so on. The diversity thus inhibits the co-creation RE process.
- (c) **Financial restrictions:** The social sector applications are either purchased by a single customer (bespoke) or purchased by many customers (mass market). However, their objective is to purchase software for social good i.e., to provide benefits to the social sector (i.e. people) without any sake of the profit. Thus, the purchases are made through grants or donation funds, which are limited. This puts pressure on the software development process as the cost overrun could result in project failures.

The financial restrictions will be the greatest inhibitor for the small software companies compared to the big companies, to deliver software for the social good. The reason is that RE is a costly activity as it requires frequent interactions with the globally dispersed and diverse customers, on a continuous basis. The area of RE in social sector context is yet to sufficient attention of researchers so to succeed, the companies should continuously perform experimentations with the users to get validated learning and invest heavily in Research & Development activities, which seems to be feasible in big companies compared to the small ones.

- (d) **Limited scope for crowdsourcing of diverse perspectives:** Due to large diversity in the users of social sector applications in terms of their age, education, and experience, it is difficult to access their understanding of the problem domain through crowdsourcing platforms. Further, it is hard to motivate them to continuously use crowdsourcing platforms because of their ignorance of the benefits to be offered by the system (because of lack of understanding of the system domain).

Further, if the crowdsourcing seems to be promising with few user segments (for instance, college students), the feedback will be noisy and diverse in the way it is expressed, which makes their automated analysis using natural language processing techniques quite difficult.

- (e) **Role of Nonfunctional requirements:** Social sector applications are to be used by the citizens for solving their social problems. The user base is highly diverse and for the solution to be absorbed among them, it should satisfy nonfunctional requirements like easy to use, high performance, security, privacy, etc. The ability of the software to satisfy the users not only depends on their functional utilities but also on the nonfunctional aspects of the software solution, which actually determine the absorption of the innovation among citizens. Further, identifying nonfunctional aspects is based on accurate analysis of the user segments (for instance, their familiarity with the use of smart phones), which is hard to be made in geographically located diverse user segments.

As per one of the respondents from company B, “We were designing a social sector application for the senior citizens. Our observation during the requirement gathering stage helped us realize that 50% of them do not know how to use smart phones and could use their mobiles for simple functions only. This helped us realize that the offered solution should have less keyboard based inputs. We offered authentication based on fingerprints and the functionality using simple touch based Graphical User Interface”.

For instance, the social sector applications access the private personal information of the citizens and are associated with the monetary benefits. This needs strong security features to avoid any breach of the system security mechanisms. Also, due to different expertise levels of the citizens and different devices they use, it is very challenging to find a solution that works optimally across all the platforms.

- (f) **Technological solution Integration Issues:** During Requirement Engineering, the efforts are being made to understand the software requirements including nonfunctional ones. The major challenge is that in many cases, the software solution architecture requires integration with other applications or databases managed by the government. This requires designing the system on the assumption of a positive outcome of the bureaucratic processes involved in gaining permission to access the information from centralized databases.
- (g) **Software evolution:** The social sector application must be evolved continuously because in the social sector context, the environment is highly volatile. For instance, the government policies always change. This needs the highly flexible RE process to map new perspectives into software requirements and ensure that further evolutions are not limited by the presence of the technical debts.

5 DISCUSSION

5.1 Real USE CASE

The software application in the problem domain of “Medical support to poor family’s application” aims to connect the poor families (called beneficiaries) with a single centralized system that could take care of their medical treatments. This software application is implemented by the company A.

The user base was too diverse as this included poor families, poor senior citizens (with no children), poor orphan children, etc. Further, the number of users is too large quantitatively and dispersed across the nation, which limits the ability of the requirement team to approach them for requirement gathering. The bad financial

situation prevailing among these users signifies the lack of ownership of the smartphones and high-speed internet facilities, which limits the ideas of using crowdsourcing for requirement gathering.

The requirement team decided to contact the local NGOs to gain better understanding of the problem domain. This proved to be beneficial as diverging perspectives about the problem domain came out of the interactions. Based on the interactions, the team found access to the few samples of the user base, which were interviewed through face-to-face traditional requirement gathering techniques, with a focus on understanding their pain points.

More interactions brought more insights like difficulty in gaining access to the specialized treatments, too many bureaucratic processes in claiming financial support for medical treatment, etc., which were not mentioned by the NGOs. This highlights the gaps between the perspectives of the stakeholders including actual users. As per one of the respondents from company A, *“The single customer never understands the actual needs of the intended users and in fact they strongly trust their understanding about the problem domain, which are just unvalidated assumptions”*.

Some of the user’s segments were also interacted using online video conferencing techniques, with remote NGO taking responsibility for arranging such meetings. The support of the volunteers in observing the poor people and interacting with them, helped the requirement team to broaden their understanding about the problem domain. However, it was difficult to interact with them initially because they were suspicious about the activities of the requirement team, and they lacked motivation because they were not confident that the solution would change their lives. It was also hard to motivate the users to participate in the requirement gathering activities. As per one of the respondents of company A, *“The hardest part is to continuously involve the users in the RE activities because they do not see value in the product in the first instant which limits their motivation levels”*.

The software solution for the identified problem was confronted with numerous challenges. These include (a) the present mobile phones used by users were simple phones and did not support the mobile applications, (b) Users have low expertise with accessing phones for special tasks, i.e. making an access to the mobile apps, which means that solutions should provide a simple “touch and select” type access, (c) the eligibility of the beneficiaries is to be verified from government records which require access to the central database that incurs privacy concerns, (d) security is to be provided to avoid breach of financial information or wrong debits of the funds, and (e) application should support nonfunctional features also to promote their adoption among the users.

Finally, the requirements team decided that the software solution will register the beneficiary’s validated information with the help of the local NGOs and volunteers. The software solution will be mobile apps which will be accessible to the local NGOs (except the privacy information of the beneficiary). The mobile number of the beneficiary will be uniquely linked to his financial and medical records. A simple call to the number or simple message, will issue him/her a ticket, which means that he is registering for the medical payment related requests. The request is handled by local NGO and volunteers. The system has support for crowdfunding for managing the funds for the charitable cause.

5.2 Framework

The results of the case study bring a useful framework that could set the boundaries for the future research in **Social Sector Requirement Engineering (SSRE)**-Requirement Engineering for Social Sector Applications. The SSRE Framework is represented by Figure 3.

The framework represents the following elements:

- (a) Processes: This represents the RE processes which is the consortium of diverse techniques customized for each user segment. The mapping process converts the non-technical perspectives of the users into software requirements (technical). The diverse perspectives of user segments as gathered during RE is merged into system understanding by the integrative processes.

Fig. 3. SSRE Framework.

- (b) **People:** The people involved in the SRE include diverse user segments, requirement analysts and domain experts. Involvement of domain experts helps the RE analysts to better understand the perspectives shared by some user segments (for instance, by farmers in their local languages) and enhance their understanding about the social sectors.
- (c) **Platforms:** To cater to the needs of the diverse users, SRE will have Non-IT based mechanisms (for instance, face to face interactions), IT Platforms (for instance, online web application with crowdsourcing and gamification features), and Hybrid mechanisms (for instance, web application with online interactions for non-technology savvy users and social interactions for technology savvy users).

The overall outcome of SSRE is the better understanding of the social sector problem domain (which helps in future software evolutions) and a ranked list of requirements that have the capacity to satisfy the diverse user segments.

5.3 Result Assessment

To ensure that the data collected in the case study is both valid and accurate, the member checking was performed with the company A and B. A total of eight employees of both companies participated in the result assessment (more than the number of case study participants). This ensures that data is correctly collected and analysed as shared by the company representatives and the company representatives share correct data.

The post case study questionnaire was shared with the company employees. The rating scale of 1 (not agree) to 5 (strongly agree) was used to specify the employee's ratings about the shared information. The aggregated collected responses are given in Table 5, which shows that a majority of the employees agreed to the validity of the results.

5.4 Implications to Government and Software Engineers

The social sector reforms had traditionally been the responsibility of the governments. Public sector institutions are held responsible for providing their services for removing social problems in order to benefit the community. Government institutions undertake social innovation through the adoption of technology for which they outsource their needs to third party software development companies (including public undertakings). The software firms have the best expertise in undertaking technical work, but they lack the deeper understanding of the societal challenges (except those that they observe in their daily lives).

Table 5. Result Assessment

Question.	1	2	3	4	5	Agreed?
Diversity is the biggest threat	0	0	0	1	7	Yes
Blended Requirement Engineering techniques is the solution in social sector.	0	0	0	0	8	Yes
All constraints are correctly mentioned.	0	0	1	2	5	Yes
Requirement Engineering practices are correctly shared.	0	0	0	0	8	Yes
User involvement decreases from Requirement gathering activity onwards.	0	0	0	0	8	Yes

The accessibility of the community will be a challenge for the requirement analysts. High quality software will provide value to the citizens that indirectly will be valuable to the governments in meeting their objectives. The ability of the solution to meet the needs of the community and get accepted by them will require a requirement analyst to be familiar with social problems. For this to happen, the role of government is quite important. The government support will not only help software firms to develop good quality software, but it will provide the research community with access to diverse experiences that will drive future research resulting in rigorous requirement engineering solutions.

6 CONCLUSION & FUTURE DIRECTIONS

The exploratory case study identified the RE process executed by the software development companies and the associated challenges in the continuous delivery of the high quality software in the social sector markets. RE must be able to capture the diverse user perspectives continuously to keep innovating the software for the social good. Maintaining continuous interactions with the social sector user segments is a must for RE but their diversity is a big challenge for their continuous involvement for evolutionary RE. The tools and techniques used in traditional or crowd based RE will not work for all user segments in the social context. In fact, a blended mix of traditional & emerging techniques should be used to converge the divergent perspectives of diversely large users into a single perspective about solution. The success of the SSRE depends on joint coordination between software engineers and the government institutions.

The diversity and size of the users of the social sector applications is a challenge for evolving the software product. It is well known that the software must be continuously innovated in order to keep satisfying the users because of the changes in the user needs and change in the environments (for instance, government regulations, policies, etc.). The changes in the environment happen very frequently, which proves a litmus test for the evolution of the software. Mapping the diverse user perspectives in the software requirements requires face to face interactions with the users and strong abilities to map non-technical aspects (user insights) into technical ones (technical requirements).

The emerging problem solving paradigms like crowdsourcing and gamified crowdsourcing (with game elements to boost motivations) is not a one size fits all solution for all segments of the users. These paradigms could work perfectly fine for the segments that are technology savvy or have experience using the technological platforms. For the remaining segments, different traditional RE techniques involving face to face interviews, nontechnical prototypes, and observations is a must. This means that RE must be executed in the blended way, involving the traditional RE and the one involving emerging problem solving paradigms. The efforts need to be made to merge the perspectives brought by traditional and new RE processes, which is another litmus test.

In the future, the following two aspects are worth investigating. First is to investigate the new RE processes suitable for those user segments that are not accessible through the latest technologies. This provides several challenges to address, including:

- (a) Efficient ways to map nontechnical perspectives into technical perspectives that are suitable for software development teams to work with.
- (b) Identification of suitable prototype solutions that could extract as much as possible information about the problem domain followed by the validation.
- (c) Identifying the ways of involving the users to prioritize the software requirements. This includes a challenge to present requirements in a nontechnical way to the users to get their preferences and then to map these preferences into ranking decisions. Ranking decisions should also resolve trade-offs with the software development team perspectives.

The second aspect is to address issues with the user segments that are accessible through the latest technologies (or techniques) like Crowd based RE, including the following:

- (a) Designing gamification that motivates different users in the same segment. For example, socialisers will be more motivated if game elements provide them the ability to enhance social interactions.
- (b) Designing crowdsourcing platforms with the ability to collect the user ideas, promote social interaction and convert the interactions into RE decisions like a prioritised list of software requirements. The automated natural language solutions for analysing the crowd inputs and social interactions will affect the accuracy of the final solutions. This is further hindered because the users will provide their feedback in different expressions, different terminologies, and at different levels of details. On one hand user privacy needs to be respected and on the other, the fake user's need to be identified with their contributions filtered out.
- (c) Merging the perspectives of the users with those that could not be accessed using technologically advanced RE platforms, is computationally hard.

The overall objective is to provide software companies the ability to execute the accurate and trustworthy RE process model that is lightweight and flexible. The need is to address research challenges with the emerging RE solutions and with the traditional processes, to provide a blended solution that has the ability to make RE a pure co-creation process with the users.

REFERENCES

- [1] P. Bhatt, A. J. Ahmad, and M. A. Roomi. 2016. Social innovation with open source software: User engagement and development challenges in India. *Technovation* 52 (2016), 28–39. <https://doi.org/10.1016/j.technovation.2016.01.004>.
- [2] R. Snijders, F. Dalpiaz, S. Brinkkemper, M. Hosseini, R. Ali, and A. Ozum. 2015. REfine: A gamified platform for participatory requirements engineering. *2015 IEEE 1st International Workshop on Crowd-Based Requirements Engineering (CrowdRE)*, Ottawa, ON, 2015, 1–6. DOI: [10.1109/CrowdRE.2015.7367581](https://doi.org/10.1109/CrowdRE.2015.7367581)
- [3] N. Paternoster, C. Giardino, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson. 2014. Software development in startup companies: A systematic mapping study. *Inf. Softw. Technol.* 56, 10 (2014), 1200–1218. [10.1016/j.infsof.2014.04.014](https://doi.org/10.1016/j.infsof.2014.04.014)
- [4] M. Cantamessa, V. Gatteschi, G. Perboli, and M. Rosano. 2018. Startups' roads to failure. *Sustainability* 2018, 10, 2346. <https://doi.org/10.3390/su10072346>.
- [5] F. Dalpiaz, R. Snijders, S. Brinkkemper, M. Hosseini, A. Shahri, and R. Ali. 2017. Engaging the crowd of stakeholders in requirements engineering via gamification. In *Gamification*, Springer, 2017, 123–135. https://doi.org/10.1007/978-3-319-45557-0_9.
- [6] A. Menkveld, S. Brinkkemper, and F. Dalpiaz. 2019. User story writing in crowd requirements engineering: The case of a web application for sports tournament planning. *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, Jeju Island, Korea (South), 2019, 174–179. DOI: [10.1109/REW.2019.00037](https://doi.org/10.1109/REW.2019.00037)
- [7] E. C. Groen, N. Seyff, R. Ali, F. Dalpiaz, J. Doerr, E. Guzman, M. Hosseini, J. Marco, M. Oriol, A. Perini, and M. Stade. 2017. The crowd in requirements engineering: The landscape and challenges. *IEEE Software* 34, 2 (2017), 44–52. DOI: [10.1109/MS.2017.33](https://doi.org/10.1109/MS.2017.33)
- [8] J. A. Khan, L. Liu, L. Wen, and R. Ali. 2019. Crowd intelligence in requirements engineering: Current status and future directions. In *Requirements Engineering: Foundation for Software Quality*. E. Knauss, M. Goedicke (eds) REFSQ 2019. *Lecture Notes in Computer Science*, vol 11412. Springer, Cham. https://doi.org/10.1007/978-3-030-15538-4_18
- [9] R. Sharma and A. Sureka. 2017. CRUISE: A platform for crowdsourcing requirements elicitation and evolution. *2017 Tenth International Conference on Contemporary Computing (IC3)*, Noida, 2017, 1–7. DOI: [10.1109/IC3.2017.8284308](https://doi.org/10.1109/IC3.2017.8284308)

- [10] M. Z. Kolpondinos and M. Glinz. 2020. GARUSO: A gamification approach for involving stakeholders outside organizational reach in requirements engineering. *Requirements Eng* 25 (2020), 185–212. <https://doi.org/10.1007/s00766-019-00314-z>
- [11] R. Cursino, D. Ferreira, M. Lencastre, R. Fagundes, and J. Pimentel. 2018. Gamification in requirements engineering: A systematic review. *2018 11th International Conference on the Quality of Information and Communications Technology (QUATIC)*, Coimbra, 2018, 119–125. DOI: [10.1109/QUATIC.2018.00025](https://doi.org/10.1109/QUATIC.2018.00025)
- [12] M. Z. Huber Kolpondinos and M. Glinz. 2017. Behind points and levels — the influence of gamification algorithms on requirements prioritization. *2017 IEEE 25th International Requirements Engineering Conference (RE)*, Lisbon, 2017, 332–341. DOI: [10.1109/RE.2017.59](https://doi.org/10.1109/RE.2017.59)
- [13] F. M. Kifetew et al. 2017. Gamifying collaborative prioritization: Does pointsification work?, *2017 IEEE 25th International Requirements Engineering Conference (RE)*, Lisbon, 2017, 322–331. DOI: [10.1109/RE.2017.66](https://doi.org/10.1109/RE.2017.66)
- [14] F. Kifetew, D. Munante, A. Perini, A. Susi, A. Siena, and P. Busetta. 2017. DMGame: A gamified collaborative requirements prioritisation tool. *2017 IEEE 25th International Requirements Engineering Conference (RE)*, Lisbon 2017, 468–469. DOI: [10.1109/RE.2017.46](https://doi.org/10.1109/RE.2017.46)
- [15] P. Runeson and M. Höst. 2009. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering* 14, 2 (2009), 131.
- [16] V. Gupta. 2021. Comment on a social network based process to minimize in-group biasedness during requirement engineering. In *IEEE Access* 9, (2021), 61752–61755. DOI: [10.1109/ACCESS.2021.3073379](https://doi.org/10.1109/ACCESS.2021.3073379)
- [17] S. Mughal, A. Abbas, N. Ahmad, and S. U. Khan. 2018. A social network based process to minimize in-group biasedness during requirement engineering. In *IEEE Access* 6 (2018), 66870–66885. DOI: [10.1109/ACCESS.2018.2879385](https://doi.org/10.1109/ACCESS.2018.2879385)
- [18] U. S. Ghanyani, M. Murad, and W. Mahmood. 2018. Crowd-based requirement engineering. *International Journal of Education and Management Engineering* 1, 3 (2018), 43–53.
- [19] E. C. Groen, J. Doerr, and S. Adam. 2015. Towards crowd-based requirements engineering a research preview. In *Requirements Engineering: Foundation for Software Quality. REFSQ 2015*. S. Fricker., K. Schneider (eds) *Lecture Notes in Computer Science*, vol 9013. Springer, Cham. https://doi.org/10.1007/978-3-319-16101-3_16
- [20] Soonh Taj, Qasim Arain, Imran Memon, and Asma Zubedi. 2019. To apply data mining for classification of crowd sourced software requirements. In *Proceedings of the 2019 8th International Conference on Software and Information Engineering (ICSIE'19)*. Association for Computing Machinery, New York, NY, USA, 42–46. DOI: <https://doi.org/10.1145/3328833.3328837>
- [21] A. Menkveld, S. Brinkkemper, and F. Dalpiaz. 2019. User story writing in crowd requirements engineering: The case of a web application for sports tournament planning. In *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*. IEEE, 174–179.
- [22] D. A. P. Sari, A. Y. Putri, M. Hanggareni, A. Anjani, M. L. O. Siswondo, and I. K. Raharjana. 2021. Crowdsourcing as a tool to elicit software requirements. In *AIP Conference Proceedings 2329*. AIP Publishing LLC, 1 (2021), 050001.
- [23] A. Adepetu, A. A. Khaja, Y. Al Abd, A. Al Zaabi, and D. Svetinovic. 2012. Crowdrequire: A requirements engineering crowdsourcing platform. In *2012 AAAI Spring Symposium Series*.
- [24] V. Gupta. 2019. Crowdsourcing and probabilistic decision-making in software engineering: Emerging research and opportunities. IGI Global, 1–182.
- [25] M. Hosseini, A. Shahri, K. Phalp, J. Taylor, R. Ali, and F. Dalpiaz. 2015. Configuring crowdsourcing for requirements elicitation. *2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS)*, 2015, 133–138. DOI: [10.1109/RCIS.2015.7128873](https://doi.org/10.1109/RCIS.2015.7128873)
- [26] R. Snijders, F. Dalpiaz, M. Hosseini, A. Shahri, and R. Ali. 2014. Crowd-centric requirements engineering. In *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*. IEEE, 614–615.
- [27] C. Li, L. Huang, J. Ge, B. Luo, and V. Ng. 2018. Automatically classifying user requests in crowdsourcing requirements engineering. *Journal of Systems and Software* 138, 108–123.
- [28] T. Ambreen. 2019. Handling socio-technical barriers involved in crowd-based requirements elicitation. *2019 IEEE 27th International Requirements Engineering Conference (RE)*, 2019, 476–481. DOI: [10.1109/RE.2019.00065](https://doi.org/10.1109/RE.2019.00065)
- [29] P. K. Murukannaiah, N. Ajmeri, and M. P. Singh. 2017. Toward automating crowd RE. *2017 IEEE 25th International Requirements Engineering Conference (RE)*, 2017, 512–515. DOI: [10.1109/RE.2017.74](https://doi.org/10.1109/RE.2017.74)
- [30] E. C. Groen. 2015. Crowd out the competition. In *2015 IEEE 1st International Workshop on Crowd-Based Requirements Engineering (CrowdRE)*. IEEE, 13–18.
- [31] D. Renzel et al. 2013. Requirements bazaar: Social requirements engineering for community-driven innovation. *Proc. of IEEE RE 2013*, 326–317.
- [32] T. Johann et al. 2015. Democratic mass participation of users in requirements engineering? *Proc. of IEEE RE 2015*, 256–261.
- [33] G. Ruhe et al. 2017. The vision: Requirements engineering in society. *Proc. of IEEE RE 2017*, 478–479.
- [34] R. Burrows et al. 2019. Motivational modelling in software for homelessness: Lessons from an industrial study. *Proc. of RE 2019*, 298–307.

Received March 2021; revised May 2021; accepted August 2021