Association for Information Systems

AIS Electronic Library (AISeL)

AMCIS 2020 Proceedings

Virtual Communities and Collaboration (VCC)

Aug 10th, 12:00 AM

Collaborating with the Crowd for Software Requirements Engineering: A Literature Review

Pascal Vogel
University of Hamburg, pascal.vogel@uni-hamburg.de

Christian Grotherr
University of Hamburg, christian.grotherr@uni-hamburg.de

Collaborating with the Crowd for Software Requirements Engineering: A Literature Review

Completed Research

Pascal Vogel
University of Hamburg
pascal.vogel@uni-hamburg.de

Christian Grotherr
University of Hamburg
christian.grotherr@uni-hamburg.de

Abstract

Requirements engineering (RE) represents a decisive success factor in software development. The novel approach of crowd-based RE seeks to overcome shortcomings of traditional RE practices such as the resource intensiveness and selection bias of stakeholder workshops or interviews. Two streams of research on crowd-based RE can be observed in literature: data-driven approaches that extract requirements from user feedback or analytics data and collaborative approaches in which requirements are collectively developed by a crowd of software users. As yet, research surveying the state of crowd-based RE does not put particular emphasis on collaborative approaches, despite collaborative crowdsourcing being particularly suited for joint ideation and complex problem-solving tasks. Addressing this gap, we conduct a structured literature review to identify the RE activities supported by collaborative crowd-based approaches. Our research provides a systematic overview of the domain of collaborative crowd-based RE and guides researchers and practitioners in increasing user involvement in RE.

Keywords

Crowdsourcing, requirements engineering, crowd-based requirements engineering, literature review.

Introduction

Requirements engineering (RE) represents a decisive success factor in software development projects (Hofmann and Lehner 2001). Traditional approaches to RE, which frequently involve techniques such as interviews of stakeholders by requirement analysts, joint workshops, participant observation or focus groups, "are inherently complex in nature – both effort and time intensive" (Sharma and Sureka 2017, p. 1) and quickly turn costly with rising stakeholder numbers. As they require co-presence, these techniques struggle to involve groups of stakeholders which are large in number or geographically distributed (Lim and Finkelstein 2012), fail to represent the diversity of heterogenous stakeholder groups (Snijders et al. 2015) and are prone to selection bias (Fernandes et al. 2012). However, the involvement of stakeholders in the RE process has been shown to be an important determinant of requirement quality and in turn overall software development project success (Zowghi et al. 2015). To overcome these shortcomings, the novel approach of crowd-based RE leverages a crowd of software users to derive software requirements using data-driven and collaborative variants or a combination of both (Groen et al. 2017). This approach enables a representative involvement of large groups of stakeholders, increases the volume and diversity of elicited requirements and enables a continuous user involvement in the RE process (Hosseini et al. 2015). In data-driven crowd-based RE, requirements are extracted by analyzing user reviews, bug reports or logfiles (Maalej et al. 2016). In collaborative crowd-based RE, stakeholders of a specific software product propose and jointly develop software requirements supported by tools such as a web-based crowdsourcing platform (Snijders et al. 2015). This collaborative approach goes beyond eliciting uni-directional feedback from software users and enables users to interact with each other and each other's proposed requirements for their joint specification or prioritization. As these collaborative approaches can be conducted asynchronously, remotely, and scale to large numbers of participants, they harbor the potential for cost and time savings.

Generally, crowdsourcing enables access to heterogenous, valuable knowledge and facilitates ideation, distributed decision making and problem-solving tasks (Ye and Kankanhalli 2013), representing a good fit for the task of defining software requirements which requires the contextual expertise and experience of software users. Collaborative crowdsourcing shows particular promise as efficient collaboration among stakeholders is also shown to lead to higher requirement quality and accuracy (Dalpiaz et al. 2017; Unkelos-Shpigel and Hadar 2015). Implementing new software features based on the wishes of software users has been shown to positively impact acceptance and user satisfaction (Fleischmann et al. 2015). In practice, software companies already leverage crowdsourcing for user involvement in RE. For example, Microsoft uses the software-as-a-service UserVoice to collect feedback from its customers on the Office 365 suite of software products[1]. As yet, research which surveys the state of crowd-based RE (Khan et al. 2019; Wang et al. 2019) does not focus specifically on collaborative approaches but focuses on data-driven variants instead. However, collaborative crowdsourcing seems particularly suited for RE as it excels at generating innovative ideas and enables multidisciplinary groups to work jointly on complex, knowledge-intensive tasks (Nakatsu et al. 2014). We address this research gap by formulating the following guiding research question:

RQ: How does collaborative crowdsourcing support requirements engineering activities?

To answer this research question, we conduct a structured literature review aimed at identifying (1) the RE activities supported by collaborative crowdsourcing and (2) the artifact design features affording this support and discuss our findings in light of extant research on collaborative crowd-based RE. Our findings show that while there is broad support for the RE activities of elicitation and analysis, specific functionality for requirements specification and validation is scarce. We further able to identify several artifact design features such as game elements or decision support tools which highlight different intended goals and audiences of collaborative approaches. In the following section, we briefly introduce the topics at the center of this review and present related research. Subsequently, we provide a detailed account of our applied methodology. Next, we present the results of our review and discuss their implications and identified research opportunities. We conclude this paper with a summary of results, contributions and limitations.

Related Work

Collaborative Crowdsourcing

Crowdsourcing can be defined as "the act of a company or institution taking a function once performed by employees and crowdsourcing it to an undefined (and generally large) network of people in the form of an open call" (Howe 2006, p. 1). Its process generally consists of (1) a requestor identifying a task to be performed, (2) broadcasting of this task online, (3) performance of the task by a crowd and (4) the selection of the best solution or synthetization of crowd-provided solutions by the requestor (Nakatsu et al. 2014). For organizations, crowdsourcing has demonstrated a variety of benefits such as the ability to remain specialized in core areas of business, cost and time reduction for task execution, access to heterogenous creative knowledge and the externalization of risk (Ye and Kankanhalli 2013). Crowdsourcing has been applied for a variety of use-cases, including the generation of innovative ideas via idea competitions, acquisition of capital via crowdfunding or for analyzing large amounts of data via microtask crowdsourcing (Brabham 2013). Among these use-cases, Nakatsu et al. (2014) describe the crowdsourcing archetype of reciprocal collaboration, a form of crowdsourcing which requires members of the crowd to cooperate beyond simple coordination by interacting with each other and sharing their individual contributions. According to them, reciprocal collaboration is suited for addressing unstructured and interdependent tasks which cannot be completed by members of the crowd single-mindedly pursuing their own self-interests. This type of crowdsourcing corresponds with the general premise of collaboration: to create value which the individual effort a of a member is not able to achieve (Briggs et al. 2009). Examples of this form of crowdsourcing include open source software and hardware development or open content projects (Nakatsu et al. 2014). Similar and related archetypes in extant literature include open collaboration (Blohm et al. 2017) or solution crowdsourcing (Prpic and Shukla 2016).

[1] *https://office365.uservoice.com/*

Crowd-Based Requirements Engineering

Requirements engineering can be defined as "the subset of systems engineering concerned with discovering, developing, tracing, analyzing, qualifying, communicating and managing requirements that define the system at successive levels of abstraction" (Hull et al. 2011, p. 8). RE comprises the two core activities of requirements development and requirements management (Wiegers and Beatty 2013). While the goal of requirements development is the definition of novel requirements, requirements management is concerned with the lifecycle of existing requirements post definition. Involving users in the RE process has been demonstrated to positively impact factors such as requirements quality, user satisfaction and the overall success of software projects (Kujala 2003; Pagano and Bruegge 2013; Zowghi et al. 2015). Crowd-based RE represents an approach for user involvement in RE which comprises "automated or semiautomated approaches to gather and analyze information from a crowd to derive validated user requirements" (Groen et al. 2017, p. 1). In case of crowd-based RE, the crowd consists of a specific group of individuals, the "current and potential users of a software product who interact among themselves or with representatives of a software company" (Groen et al. 2017, p. 1). Crowd-based RE aims to alleviate some of the challenges faced by traditional approaches to RE which rely on costly and time-consuming interviews, workshops and fail to involve large, diverse and geographically distributed groups of stakeholders (Law et al. 2012). Crowd-based RE follows a data-driven or collaborative approach. Dalpiaz et al. (2018) propose a data-driven approach by outlining the usage of natural language processing for the extraction of software requirements from a variety of data sources. In contrast, Snijders et al. (2015) present an approach for collaborative crowd-based RE by developing a method and web-based software artifact which enable stakeholders, software engineers and requirements analysts to collaboratively develop software requirements. Research on collaborative-crowd based RE is closely aligned with the definition of collaborative crowdsourcing presented above, with the definition of individual software requirements representing the unstructured and interdependent tasks to be performed by the crowd. As data-driven approaches rely on usage data, they are predominantly applied in the post-implementation phase of software products, i.e., when software users have access to a finished version of the software product. Conversely, considering the current body of research, we observe collaborative approaches to crowd-based RE to be focused on the pre-implementation phase of software products.

Methodology

We conduct our structured literature review based on established guidance by Webster and Watson (2002), vom Brocke et al. (2015) and Bandara et al. (2015), aiming to produce a systematic, transparent and reproducible review (Cram 2019). We commence with the extraction of relevant literature from a selection of databases which includes ABI/INFORM Complete (via ProQuest), ACM Digital Library, AIS Electronic Library (AISeL), Business Source Complete (via EBSCO), IEEE Xplore, ScienceDirect and Springer Link, covering important publications in IS and related fields such as software engineering. Based on an iterative scoping search (Booth et al. 2012), we identify the combination of *CROWD* AND REQUIREMENT** as the most promising search term. Other terms and term combinations such as social or collaborative RE lacked an IT-mediated focus or did not lead to relevant results. Where possible, we filter for peer-reviewed results only. Searching the fields title, abstract and keyword, we arrive at 770 hits across our selected databases. Table 1 presents our raw search results ahead of any filtering and deduplication while our final selection of included articles is available as part of a literature matrix presented in Table 2.

Database	ABI/ INFORM	ACM DL	AISeL	EBSCO BSC	IEEE Xplore	Science Direct
Hits (relevant)	107 (0)	14 (3)	23 (1)	165 (1)	319 (7)	142 (1)

Table 1. Raw search hits per database (relevant articles in parenthesis, incl. duplicates)

Based on the previously presented definitions and related work, our search is targeted at scholarly articles discussing approaches or design artifacts which (a) support one or more sub-activities of RE, (b) leverage crowdsourcing, (c) follow a collaborative approach and (d) are IT-mediated. Filtering the identified articles based on these inclusion criteria and after the removal of duplicates, we are presented with 11 relevant articles. Additionally, we conduct a backward search by analyzing each paper's references and a forward search using both the Google Scholar and Web of Science citation indexing services, adding 5 relevant articles. The last iteration of our search was conducted in December of 2019. In the analysis phase of our literature review, we perform qualitative data analysis (Bandara et al. 2015) using MAXQDA. Initially, we perform a deductive coding phase based on the RE sub-activities elicitation, analysis, specification and validation. In a second phase, we follow the Wolfswinkel et al. (2013), conducting of open, axial and analytical coding iterations. For open coding, we analyze each article and induce themes based on artifact design features, for instance "comments" or "votes". In the context of this work, we define design features as the individual capabilities of a design artifact (Hevner et al. 2004). During axial coding, we identify interconnections between codes and group them into higher-level categories, for instance "tags" and "full-text search" into a "information retrieval" sub-category. Finally, using selective coding, we integrate and refine the identified sub-categories, for example grouping "game roles" and "game mechanics" into the "game elements" category. We aim to ensure the validity and reliability of our research by each coding step being performed independently by two researchers and in two iterations before results are integrated.

Results

In the following, we present the results of our qualitative data analysis regarding both (1) the RE activities supported by approaches and artifacts presented in the articles included in our review as well as (2) significant artifact design features for collaborative crowd-based RE. Table 2 provides an overview of the included articles and addressed aspects in the form of a literature concept matrix.

Requirements Engineering Activities

Requirements **elicitation** describes the activity of identifying the needs and constraints of a software product's stakeholders as well as the relevant stakeholders themselves (Wiegers and Beatty 2013). As functionality for eliciting requirements was defined as a criterium for inclusion in our review, all presented artifacts support the elicitation of requirements from a crowd of software users. However, Lim et al. (2010) and Lim and Finkelstein (2012) do not directly support the elicitation of requirements but the collaborative identification of relevant stakeholders which subsequently can be queried for requirements. Elicitation in case of our analyzed artifacts takes place via software users actively and autonomously communicating their needs and requirements on a crowdsourcing platform. Most analyzed articles implement a custom web-based software artifact as the crowdsourcing platform. Seyff et al. (2015) represent a notable exception as they leverage private groups on the online social network Facebook as the basis for their crowd-based approach. Rashid et al. (2008) complement their web-based platform with a software annotation tool which allows users to directly relate requirements to software screenshots.

Requirements **analysis** is concerned with the refinement of requirements including ensuring their quality, decomposing or merging requirements into appropriate levels of detail, identifying dependencies between requirements and negotiating their priority (Wiegers and Beatty 2013). We identify a number of quality assurance mechanisms among the analyzed artifacts. Vogel et al. (2019) prompt software users to perform a self-assessment survey during requirement submission which considers requirements quality properties such as feasibility or completeness and provides users with feedback depending on their answers. Ninaus et al. (2014) leverage model-based diagnosis (Reiter 1987) to identify inconsistencies among sets of requirements and recommended actions for restoring consistency. A number of artifacts implement functionality for branching out one requirement into several (sub-)requirements (Dalpiaz et al. 2017; Lim and Finkelstein 2012; Snijders et al. 2015) with similar functionality for merging overlapping requirements into one. In this regard, Lohmann et al. (2009) afford users the functionality to map relationships between requirements such as one requirement's correct implementation being dependent on another requirement being implemented first. Ninaus et al. (2014) use dependency detection to indicate relationships between requirements, albeit the concrete assertion of a semantic relationship falls to the stakeholders. Most artifacts support the qualitative negotiation of implementation priorities in a set of requirements via a commenting or discussion system. Offering a more structured form of negotiation, we also observe a variety of crowd-based prioritization systems. Snijders et al. (2015) and Dalpiaz et al. (2017) implement a simple binary agree/disagree scheme while Seyff et al. (2015) leverage the like mechanism of a popular online social network in similar fashion. Fernandes et al. (2012), Lim and Finkelstein (2012) and Lohmann et al. (2009) use a five-star rating system for crowd-based requirement prioritization. Klamma et al. (2011) complement a five-star rating with a price-finding mechanism which allows software users to pledge financial rewards for the implementation of a requirement. Closely related to this price-finding approach, Law et al. (2012) allocate each member of the crowd a limited budget of abstract votes which can be distributed on one or more requirements. Both approaches share resemblance to enterprise crowdfunding (Feldmann and Gimpel 2016). Other articles present a multi-criteria prioritization system: Kolpondinos and Glinz (2017) and Kolpondinos and Glinz (2019) let users rate requirements on a three-point-scale for each of the two criteria relevance (relevant, neutral, irrelevant) and popularity (like, neutral, dislike). Similarly, Yang et al. (2008) poll their users on the criteria business importance and ease of realization for each requirement.

During requirements **specification**, software requirements are documented in a structured, standardized and accessible manner that is readily understandable by all stakeholders (Wiegers and Beatty 2013). In case of Kolpondinos and Glinz (2019), requirements are submitted using a simple user story template. Vogel et al. (2019) pre-filled their submission form with a template for submitting requirements based on standard requirement specification documents. As opposed to the majority of analyzed cases where artifacts are submitted and subsequently edited only by their initial author, Lohmann et al. (2009) and Yang et al. (2008) enable a wiki-style collaboration where multiple users can work on specifying a single requirement in parallel. Most articles do not provide examples of requirements specified using the presented approaches but based on the available information it can be assumed that these requirement do not meet the standards of traditional requirements specification documents (Wiegers and Beatty 2013). This is not surprising, considering the target crowd for most artifacts are stakeholders untrained in requirements specification who cannot be expected to produce such fully specified requirement specification documents. In our literature matrix in Table 2, we only mark those articles which offer support for requirements specification which goes beyond the most basic specification of requirements in form of free text as to highlight the most insightful contributions among the analyzed artifacts. Requirements **validation** entails the reviewing and testing of requirements by various stakeholders, the definition of acceptance criteria and ensuring stakeholder satisfaction in general (Wiegers and Beatty 2013). It can be argued that in case of the analyzed collaborative platforms, a constant implicit review process takes place by members of the crowd which are able to screen and comment on the requirements submitted by their peers. Adepetu et al. (2012) implement a requirements review method which enables a client user role to perform an acceptance review on requirements submitted by a crowd of users. Klamma et al. (2011) and Law et al. (2012) provide another form of acceptance validation by linking requirements on their platform to implemented software features and providing monitoring data on their usage. As with the requirements specification, we only mark articles which explicitly present functionality targeted towards requirements validation as providing support for this activity in our concept matrix in Table 2.

Artifact Design Features

Besides offering crowd-based prioritization, some articles present further **decision support tools** aimed at facilitating the selection of high impact requirements. Klamma et al. (2011) implement a dashboard which aggregates several indicators per requirement such as user-provided ratings, communication and development activity, usage statistics for implemented requirements as well as its estimated financial value. Ninaus et al. (2014) apply a majority voting strategy based on group decision heuristics (Ninaus 2012) in order to combine the diverging priorities of individual users into one consistent and actionable prioritization result. In doing so, they not only let users determine the priority of the requirement but multiple additional criteria such as feasibility, risk and relevance. Vogel et al. (2019) allow product owners to compare submitted requirements using impact and effort matrices based on user-provided votes and an effort estimation on the Planning Poker Scale (Cohn 2005). Identifying and implementing suitable motivational mechanisms represents a central challenge for any crowdsourcing initiative (Blohm et al. 2017), collaborative crowd-based RE being no exception. Several of the analyzed articles leverage **game elements** as a motivational mechanic, albeit with varying degrees of comprehensiveness. Snijders et al. (2015) and Dalpiaz et al. (2017) complement the RE approach with game mechanics such as points, leaderboards and resources. Other artifacts such as those presented by Fernandes et al. (2012) and Kolpondinos and Glinz (2019) exhibit a deeper integration of gamification, adding additional game elements such as multiple types of points, progress bars, levels, rewards and game roles with associated capabilities. Beyond game elements, most articles explicitly or implicitly describe the opportunity to gain social recognition as a further motivational mechanic. In their internal crowdsourcing setting, Vogel et al. (2019) afford users to position themselves as domain experts and thereby gain the recognition of peers.

Similarly, Adepetu et al. (2012) describe a potential "sense of prestige" while Dalpiaz et al. (2017) identify social influence and reciprocal benefit as motivators.

Several analyzed artifacts implement functionality for **stakeholder relationship management**. This includes features which enables developers or product owners to communicate the status of requirements to users, provide reasoning for implementation decisions, or for managing stakeholder expectations. For example, Rashid et al. (2008) or Law et al. (2012) allow for the assignment of statuses such as Backlog, Implemented or Closed to requirements in order to communicate its lifecycle state to software users. In case of Law et al. (2012), software users are informed of status changes via a notification system, triggered for example once a requirement has been implemented by developers. The artifact presented by Vogel et al. (2019) further affords product owners to illustrate the lifecycle status of requirements to software users via a configurable software roadmap and forces product owners to provide reasoning when a requirement status is changed. Features of this category are relevant only in specific role constellations, i.e. when there is a clear distinction between the crowd of stakeholders as proposers of requirements and an organization or its representatives deciding on their implementation.

	Supported RE activities				**RE-relevant design features**				
REFERENCES	Elicitation	Analysis	Specification	Validation	Decision support tools	Game elements	Stakeholder relationship management	Recommender systems	Retrievability & traceability
Adepetu et al. (2012)	✓			✓					✓
Dalpiaz et al. (2017)	✓	✓				✓			✓
Fernandes et al. (2012)	✓	✓				✓			✓
Klamma et al. (2011)	✓	✓		✓	✓				✓
Kolpondinos and Glinz (2019)	✓	✓	✓			✓			✓
Kolpondinos and Glinz (2017)	✓	✓	✓			✓			✓
Law et al. (2012)	✓	✓		✓			✓		✓
Lim and Finkelstein (2012)		✓						✓	
Lim et al. (2010)		✓							
Lohmann et al. (2009)	✓	✓	✓					✓	✓
Ninaus et al. (2014)	✓	✓			✓			✓	
Rashid et al. (2008)	✓						✓		✓
Seyff et al. (2015)	✓	✓							✓
Snijders et al. (2015)	✓	✓				✓			✓
Vogel et al. (2019)	✓	✓	✓		✓		✓		✓
Yang et al. (2008)	✓	✓	✓						
	14	14	5	3	3	5	3	3	12
	(88%)	(88%)	(31%)	(19%)	(19%)	(31%)	(19%)	(19%)	(75%)

Table 2. Literature concept matrix

Recommender systems are a useful building block in automating and scaling RE practices to large crowds of software users (Castro-Herrera et al. 2008). Ninaus et al. (2014) use content-based filtering to recommend requirements based on user preferences. Similarly, the system proposed by Lim and Finkelstein (2012) recommends relevant requirements to users based on their preferences and previous rating behavior. Lohmann et al. (2009) use a recommender system during the requirement submission process which alerts users to similar existing requirements to prevent submission of duplicates and stimulate their interest in collaboration. Several analyzed artifacts implement user-generated tagging systems to enable the **retrievability and traceability** of requirements. In case of Lohmann et al. (2009), project managers provide an initial taxonomy as the basis for classifying submitted requirements in addition to a user-generated tag system. This results in both a top-down taxonomic as well as bottom-up folksonomic classification of requirements. By prompting users to add a definition for their newly created tags, Lohmann et al. (2009) further build up a shared glossary of project relevant terms. Most artifacts such as Klamma et al. (2011), Adepetu et al. (2012) or Kolpondinos and Glinz (2017) implement search functionality for the retrieval of requirements.

Discussion and Future Research

Our review shows that approaches to collaborative crowd-based RE as yet do not address the entire spectrum of activities which comprises the RE process. While there is comprehensive support for elicitation and analysis activities, specification and validation are not universally supported with specific functionality However, one must acknowledge that the intended goal of the majority of included articles is to increase user involvement in the RE process in the hopes of improving requirements quality (Zowghi et al. 2015). This goal can be achieved via crowd-based elicitation of user needs and requirements as well as crowd-based analysis, e.g. via crowd-based prioritization. Therefore, the comprehensive and in-depth specification of requirements or their exhaustive validation could be deemed as out of scope for some artifacts depending on their underlying goals and intentions. Considering the RE-relevant design features reveals, a number of artifacts seek to gamify the RE process (Snijders et al. 2015) or go even further resulting in RE-as-a-game approaches (Kolpondinos and Glinz 2017). Other artifacts put emphasis on the software product owner role as the intended audience, making intelligent decision making and communication with stakeholders the primary goals (Vogel et al. 2019). The gaps in RE process coverage and feature support can serve as a blueprint for developing more comprehensive approaches and artifacts for collaborative crowd-based RE.

As a general observation, most analyzed articles regard RE as detached or isolated from the greater software engineering practices it is embedded in. However, the effectiveness of the proposed approaches for collaborative crowd-based RE likely depends heavily on the ability of the software-providing organization to deliver the requirements elicited from software users, necessitating a smooth interplay between the development of requirements and their implementation in software. This issue becomes even more significant when viewing the elicitation of software requirements not as a one-off step in a waterfall-style development model but as a continuous process which accompanies agile software development such as Scrum, Kanban or Extreme Programming. Among our analyzed papers, we can see some initial efforts towards achieving this goal. Snijders et al. (2015) and Dalpiaz et al. (2017) envision a development sprint as the last step in their crowd-based RE method. Likewise, Rashid et al. (2008) specify a final 'Implement' process step as part of their process. Law et al. (2012) integrate their collaborative crowd-based RE artifact with the JIRA issue tracking software via its application programming interface. In this regard, previous research has demonstrated that crowd-based approaches cannot simply be "bolted onto existing systems" (Gleasure et al. 2017, p. 339) underlining the need for an adequate embedding of novel artifacts and practices into existing contexts (Orlikowski and Iacono 2001). Future research should focus on investigating possible avenues for tightly integrating collaborative crowd-based RE with software engineering practices as well as the organizational context as a whole.

Considering the type of contribution made by the articles included in our review from a design science perspective (Gregor and Hevner 2013), all articles present situated implementations of artifacts in the form of instantiations of software products and in some cases associated processes. Kolpondinos and Glinz (2019) and Vogel et al. (2019) go one step further towards providing more abstract design knowledge and present design principles for one specific domain of crowd-based RE. Assessing the set of analyzed articles as a whole, we determine a lack of explicit design knowledge such as constructs, methods, design principles or even design theories (Gregor and Hevner 2013) which guide researchers and practitioners in designing their own collaborative crowd-based RE approaches. Although this lack of formalized design knowledge may be explained by most included articles not originating from information systems literature, future research should nevertheless seek to develop mature design knowledge which guide researchers and practitioners in designing approaches to collaborative crowd-based RE.

Only a very limited number of analyzed articles combines collaborative and data-driven variants to RE as envisioned by Groen et al. (2017). For example, Klamma et al. (2011) integrate a usage data monitoring of implemented requirements which originate from their collaborative crowd-based RE artifacts. As can be observed in our literature matrix, some artifacts also complement the collaborative approach with data-driven recommender systems. Generally, from the perspective of collaborative crowd-based RE, data-driven elements could be integrated following two avenues. First, needs identified automatically via data analysis could serve as an input to and starting point for collaborative requirements development. Second, data-driven elements could be used to validate collaboratively developed requirements by monitoring their real-world usage and related feedback. In practice, this combination of collaborative and data-driven approaches is more pronounced, again looking at for example the platform UserVoice or other offerings such as UseResponse[2] which both integrate qualitative and quantitative approaches. Future research should expand its focus beyond extant literature and survey existing platforms for collaborative crowd-based RE, for example using taxonomy development.

Conclusion

Collaborative crowdsourcing's ability to effectively address ideation and complex problem-solving tasks has drawn the attention of crowd-based RE researchers. However, studies surveying the state of crowd-based RE do not focus on collaborative but on data-driven variants. We conduct a structured literature review of extant research on collaborative crowd-based RE to identify both the supported RE activities and significant RE-relevant design features of artifacts presented by included studies. Among 16 relevant articles, we observe support for the elicitation and analysis of requirements while the specification and validation are beyond the scope of the majority of presented artifacts. We identify the integration of collaborative crowd-based RE with software development processes, development of design knowledge and the integration of data-driven and collaborative variants as avenues for future research. The contribution of our work is twofold. We contribute to research on crowdsourcing and crowd-based RE by providing a systematic overview of literature in the domain of collaborative crowd-based RE. Second, our work can guide researchers and practitioners in designing their own collaborative crowd-based approaches by providing an overview of existing design features and identifying research gaps for novel ones. Our review is faced with some limitations. Our keyword selection strongly determined the included articles and thereby the results of our work. Due to the relative novelty of the topic we include a wide variety of publication in our review and are not able to exclusively limit our search to established, peer-reviewed outlets. Furthermore, despite our data analysis being independently performed by two researchers and following a standardized and structured approach, human error and selection bias may have influenced the outcome of our analysis.

REFERENCES

Adepetu, A., Ahmed, K.A., Al Abd, Y., Al Zaabi, A., and Svetinovic, D. 2012. "Crowdrequire: A Requirements Engineering Crowdsourcing Platform," in: *Association for the Advancement of Artificial Intelligence Spring Synopsium*. Stanford, CA, USA: pp. 2-7.

Bandara, W., Furtmueller, E., Gorbacheva, E., Miskon, S., and Beekhuyzen, J. 2015. "Achieving Rigor in Literature Reviews: Insights from Qualitative Data Analysis and Tool-Support," *Communications of the Association for Information Systems* (37).

Blohm, I., Zogaj, S., Bretschneider, U., and Leimeister, J.M. 2017. "How to Manage Crowdsourcing Platforms Effectively?," *California Management Review* (60:2), pp. 122-149.

Booth, A., Papaioannou, D., and Sutton, A. 2012. *Systematic Approaches to a Successful Literature Review*.

Brabham, D.C. 2013. *Crowdsourcing*. Cambridge, MA: MIT Press.

Briggs, R.O., Kolfschoten, G., de Vreede, G.-J., Albrecht, C., Dean, D.R., and Lukosch, S. 2009. "A Seven-Layer Model of Collaboration: Separation of Concerns for Designers of Collaboration Systems," in: *International Conference on Information Systems (ICIS)*. Phoenix, AZ, USA.

Castro-Herrera, C., Duan, C., Cleland-Huang, J., and Mobasher, B. 2008. "Using Data Mining and Recommender Systems to Facilitate Large-Scale, Open, and Inclusive Requirements Elicitation Processes," *2008 16th IEEE International Requirements Engineering Conference*, pp. 165-168.

Cohn, M. 2005. *Agile Estimation and Planning*. Upper Saddle River, NJ, USA: Prentice Hall.

Cram, W.A. 2019. "(Re)Considering the Concept of Reproducibility of Information Systems Literature Reviews," in: *Americas Conference on Information Systems*. Cancun, Mexico.

[2] *https://useresponse.com*

Dalpiaz, F., Ferrari, A., Franch, X., and Palomares, C. 2018. "Natural Language Processing for Requirements Engineering: The Best Is yet to Come," *IEEE Software* (35:5), pp. 115-119.

Dalpiaz, F., Snijders, R., Brinkkemper, S., Hosseini, M., Shahri, A., and Ali, R. 2017. "Engaging the Crowd of Stakeholders in Requirements Engineering Via Gamification," in *Gamification: Using Game Elements in Serious Contexts,* S. Stieglitz, C. Lattemann, S. Robra-Bissantz, R. Zarnekow and T. Brockmann (eds.). Cham, Germany: Springer International Publishing, pp. 123-135.

Feldmann, N., and Gimpel, H. 2016. "Financing Projects through Enterprise Crowdfunding: Understanding the Impact of Proposal Characteristics on Funding Success," in: *European Conference on Information Systems (ECIS)*. Istanbul, Turkey.

Fernandes, J., Duarte, D., Ribeiro, C., Farinha, C., Pereira, J.M., and Silva, M.M.d. 2012. "Ithink: A Game-Based Approach Towards Improving Collaboration and Participation in Requirement Elicitation," *Procedia Computer Science* (15:1), pp. 66-77.

Fleischmann, M., Grupp, T., Amirpur, M., and Benlian, A. 2015. "When Updates Make a User Stick: Software Feature Updates and Their Differential Effects on Users' Continuance Intentions," in: *International Conference on Information Systems (ICIS)*. Forth Worth, TX, USA.

Gleasure, R., O'Reilly, P., and Cahalane, M. 2017. "Inclusive Technologies, Selective Traditions: A Socio-Material Case Study of Crowdfunded Book Publishing," *Journal of Information Technology* (32:4), December 01, pp. 326-343.

Gregor, S., and Hevner, A.R. 2013. "Positioning and Presenting Design Science Research for Maximum Impact," *Management Information Systems Quarterly* (37:2), pp. 337-356.

Groen, E.C., Seyff, N., Ali, R., Dalpiaz, F., Doerr, J., Guzman, E., Hosseini, M., Marco, J., Oriol, M., Perini, A., and Stade, M. 2017. "The Crowd in Requirements Engineering: The Landscape and Challenges," *IEEE Software* (34:2), pp. 44-52.

Hevner, A.R., March, S.T., Park, J., and Ram, S. 2004. "Design Science in Information Systems Research," *Management Information Systems Quarterly* (28:1), pp. 75-105.

Hofmann, H.F., and Lehner, F. 2001. "Requirements Engineering as a Success Factor in Software Projects," *IEEE Software* (18:4), pp. 58-66.

Hosseini, M., Shahri, A., Phalp, K., Taylor, J., Ali, R., and Dalpiaz, F. 2015. "Configuring Crowdsourcing for Requirements Elicitation," *2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS)*, pp. 133-138.

Howe, J. 2006. "The Rise of Crowdsourcing," *Wired Magazine* (14:6), pp. 1-4.

Hull, E., Jackson, K., and Dick, J. 2011. *Requirements Engineering*. London, UK: Springer.

Khan, J.A., Liu, L., Wen, L., and Ali, R. 2019. "Crowd Intelligence in Requirements Engineering: Current Status and Future Directions," in: *Requirements Engineering: Foundation for Software Quality*. pp. 245-261.

Klamma, R., Jarke, M., Hannemann, A., and Renzel, D. 2011. "Der Bazar Der Anforderungen - Open Innovation in Emergenten Communities," *Informatik-Spektrum* (34:2), pp. 178-191.

Kolpondinos, M.Z., and Glinz, M. 2019. "Garuso: A Gamification Approach for Involving Stakeholders Outside Organizational Reach in Requirements Engineering," *Requirements Engineering*), June 07.

Kolpondinos, M.Z.H., and Glinz, M. 2017. "Behind Points and Levels — the Influence of Gamification Algorithms on Requirements Prioritization," *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pp. 332-341.

Kujala, S. 2003. "User Involvement: A Review of the Benefits and Challenges," *Behaviour & Information Technology* (22:1), 2003/01/01, pp. 1-16.

Law, E.L.-C., Chatterjee, A., Renzel, D., and Klamma, R. 2012. "The Social Requirements Engineering (Sre) Approach to Developing a Large-Scale Personal Learning Environment Infrastructure," *21st Century Learning for 21st Century Skills,* A. Ravenscroft, S. Lindstaedt, C.D. Kloos and D. Hernández-Leo (eds.), Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 194-207.

Lim, S.L., and Finkelstein, A. 2012. "Stakerare: Using Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation," *IEEE Transactions on Software Engineering* (38:3), pp. 707-735.

Lim, S.L., Quercia, D., and Finkelstein, A. 2010. "Stakesource: Harnessing the Power of Crowdsourcing and Social Networks in Stakeholder Analysis," *2010 ACM/IEEE 32nd International Conference on Software Engineering*, pp. 239-242.

Lohmann, S., Tramp, S., Heim, P., and Heino, N. 2009. "A Web Platform for Social Requirements Engineering," in: *German Informatics Societey: Software Engineering Conference*. Kaiserslautern, Germany.

Maalej, W., Nayebi, M., Johann, T., and Ruhe, G. 2016. "Toward Data-Driven Requirements Engineering," *IEEE Software* (33:1), pp. 48-54.

Nakatsu, R.T., Grossman, E.B., and Iacovou, C.L. 2014. "A Taxonomy of Crowdsourcing Based on Task Complexity," *Journal of Information Science* (40:6), pp. 823-834.

Ninaus, G. 2012. "Using Group Recommendation Heuristics for the Prioritization of Requirements," in: *Proceedings of the sixth ACM conference on Recommender systems*. Dublin, Ireland: Association for Computing Machinery, pp. 329–332.

Ninaus, G., Felfernig, A., Stettinger, M., Reiterer, S., Leitner, G., Weninger, L., and Schanil, W. 2014. "Intellireq: Intelligent Techniques for Software Requirements Engineering," in: *Proceedings of the European Conference on Artificial Intelligence (ECAI)*. Prague, CZ: pp. 1161-1166.

Orlikowski, W.J., and Iacono, C.S. 2001. "Research Commentary: Desperately Seeking the "It" in It Research—a Call to Theorizing the It Artifact," *Information Systems Research* (12:2), pp. 121-134.

Pagano, D., and Bruegge, B. 2013. "User Involvement in Software Evolution Practice: A Case Study," in: *International Conference on Software Engineering (ICSE)*. San Francisco, CA, USA: pp. 953-962.

Prpic, J., and Shukla, P. 2016. "Crowd Science: Measurements, Models, and Methods," in: *Proceedings of the 49th Hawaii International Conference on System Sciences (HICSS)*. Kauai, HI, USA: pp. 4365-4374.

Rashid, A., Wiesenberger, J., Meder, D., and Baumann, J. 2008. "Bringing Developers and Users Closer Together: The Openproposal Story," in: *Proceedings of the Multikonferenz Wirtschaftsinformatik (MKWI)*. Munich, Germany.

Reiter, R. 1987. "A Theory of Diagnosis from First Principles," *Artificial Intelligence* (32:1), 1987/04/01/, pp. 57-95.

Seyff, N., Todoran, I., Caluser, K., Singer, L., and Glinz, M. 2015. "Using Popular Social Network Sites to Support Requirements Elicitation, Prioritization and Negotiation," *Journal of Internet Services and Applications* (6:1), April 24, p. 7.

Sharma, R., and Sureka, A. 2017. "Cruise: A Platform for Crowdsourcing Requirements Elicitation and Evolution," *Proceedings of the 10th International Conference on Contemporary Computing*, pp. 1-7.

Snijders, R., Dalpiaz, F., Brinkkemper, S., Hosseini, M., Ali, R., and Ozum, A. 2015. "Refine: A Gamified Platform for Participatory Requirements Engineering," in: *International Workshop on Crowd-Based Requirements Engineering (CrowdRE)*. Ottawa, Canada.

Unkelos-Shpigel, N., and Hadar, I. 2015. "Inviting Everyone to Play: Gamifying Collaborative Requirements Engineering," in: *International Workshop on Empirical Requirements Engineering (EmpiRE)*. Ottawa, Canada: pp. 13-16.

Vogel, P., Grotherr, C., and Semmann, M. 2019. "Leveraging the Internal Crowd for Continuous Requirements Engineering – Lessons Learned from a Design Science Research Project," in: *European Conference on Information Systems*. Stockholm-Uppsala, Sweden.

vom Brocke, J., Simons, A., Riemer, K., Niehaves, B., Plattfaut, R., and Cleven, A. 2015. "Standing on the Shoulders of Giants: Challenges and Recommendations of Literature Search in Information Systems Research," *CAIS* (37:9).

Wang, C., Daneva, M., van Sinderen, M., and Liang, P. 2019. "A Systematic Mapping Study on Crowdsourced Requirements Engineering Using User Feedback," *Journal of Software: Evolution and Process* (31:10), p. e2199.

Webster, J., and Watson, R.T. 2002. "Analyzing the Past to Prepare for the Future: Writing a Literature Review," *MIS Quarterly* (26:2).

Wiegers, K.E., and Beatty, J. 2013. *Software Requirements*. London, UK: Pearson Education.

Wolfswinkel, J.F., Furtmueller, E., and Wilderom, C.P.M. 2013. "Using Grounded Theory as a Method for Rigorously Reviewing Literature," *European Journal of Information Systems* (22:1), pp. 45-55.

Yang, D., Wu, D., Koolmanojwong, S., Brown, A.W., and Boehm, B.W. 2008. "Wikiwinwin: A Wiki Based System for Collaborative Requirements Negotiation," *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, pp. 24-24.

Ye, H., and Kankanhalli, A. 2013. "Leveraging Crowdsourcing for Organizational Value Co-Creation," *Communications of the Association for Information Systems* (33), pp. 225-244.

Zowghi, D., Rimini, F.d., and Bano, M. 2015. "Problems and Challenges of User Involvement in Software Development: An Empirical Study," in: *Proceedings of the 19th Conference on Evaluation and Assessment in Software Engineering (EASE)*. Nanjing, China: ACM, pp. 1-10.