Data-Driven Agile Requirements Elicitation through the Lenses of Situational Method Engineering

Xavier Franch
Univ. Politècnica de Catalunya
Barcelona, Spain
franch@essi.upc.edu

Aron Henriksson
Stockholm University
Kista, Sweden
aronhen@dsv.su.se

Jolita Ralyté
University of Geneva
Geneva, Switzerland
jolita.ralyte@unige.ch

Jelena Zdravkovic
Stockholm University
Kista, Sweden
jelenaz@dsv.su.se

***Abstract*— Ubiquitous digitalization has led to the continuous generation of large amounts of digital data, both in organizations and in society at large. In the requirements engineering community, there has been a growing interest in considering digital data as new sources for requirements elicitation, in addition to stakeholders. The volume, dynamics, and variety of data makes iterative requirements elicitation increasingly continuous, but also unstructured and complex, which current agile methods are unable to consider and manage in a systematic and efficient manner. There is also the need to support software evolution by enabling a synergy of stakeholder-driven requirements elicitation and management with data-driven approaches. In this study, we propose extension of agile requirements elicitation by applying situational method engineering. The research is grounded on two studies in the business domains of video games and online banking.**

Keywords—data-driven requirements elicitation, situational method engineering, agile system development, design science.

I. Introduction

Agile methodologies have been guiding rapid development of minimum viable products, which evolve through team-based practices that facilitate timely improvements [1]. Relevant to requirements engineering are continuous communication with the customer, rapid customer's feedback, late and limited documentation, frequent system releases, and quick reaction to change.

Agile practices fit well when big data is considered as the source of requirements: many data are continuously emerging, enabling thus timely changes and refinements of requirements in later releases; and once collected they may be stored for different processing at any time; upon processing is completed, the data can be condensed by grouping similar information. As a result, there are ongoing efforts to support and enrich requirements elicitation activities by automatically mining and processing digital data for information about requirements, and support requirements engineers in their decisions about which requirements to include in upcoming system releases [2].

There is a wide spectrum of digital data sources that may be exploited. In contrast to human sources, i.e. stakeholders, digital sources continuously generate data that is often not originally created for the purposes of requirements elicitation, e.g. on forums, microblogs, machine-generated trace logs, and sensor data. Streams of large volumes of data can be exploited to enable automation of a continuous requirements elicitation process using AI techniques that, for instance, combine natural language processing and machine learning [3]. On the other hand, the characteristics of big data (in terms of volume, velocity, variety, and veracity) present numerous challenges in effectively extracting and aggregating requirements-pertinent information, which requires a combination of various algorithms and data processing techniques, as well as the identification of appropriate analytical tasks. These decisions are highly dependent on the type of digital source that is targeted. Furthermore, as the data is not created explicitly for requirements elicitation, it tends to be limited in terms of completeness and correctness with respect to a canonical requirements format that is understandable and feasible to develop and implement. Consequently, data-driven requirements elicitation outcomes risk to be of low effectiveness and practical use [4], or requiring a substantial manual effort [5].

While many companies are increasingly interested in exploiting the potential value and opportunities that these data sources provide for requirements engineering, they face the challenge of understanding *how to collect, analyse, and aggregate data using existing techniques and apply data-driven requirements elicitation principles such that actionable insights concerning requirements can be generated*. This is a highly complex task, not considered by current requirements elicitation methods, which still focus on traditional stakeholder-oriented approaches [6].

We therefore aim at organizing the existing body of research on data-driven requirements elicitation into a holistic method that we build by applying Situational Method Engineering [7]. The aim is thereby to take into consideration the various situations that organisations can face and to put together different intentions, conditions, and alternatives to guide them in effectively using their heterogenous digital data and processing it with suitable techniques and algorithms for elicitation of requirements.

II. Background

Data-Driven Requirements Elicitation (DDRE). DDRE is motivated by the increasing inflow of user feedback, both in the form of natural language (e.g., discussion forums, online and app reviews, microblogs) and machine-generated sources (e.g., usage data, sensor data) [2][8]. Most of efforts has focused on identifying and classifying requirements-related information from few sources such as user feedback and machine-generated data. A systematic literature review of DDRE techniques showed that there is still a lack of methods to combine more diverse types of data sources [9]. Recently, more holistic views on DDRE have been presented, which integrate information from different sources, based on domain ontologies [10], metamodeling [11] or contextual user feedback gathering [12]. This information is the basis upon which requirements engineers may elicit requirements of different kinds, e.g. quality requirements [13]. Although these studies provide applicable solutions, they do not address the fact that DDRE highly depends on contextual factors such as available data sources or the expertise of the company, and thus adoption of DDRE is still an ad-hoc process.

 DOI 10.1109/RE51729.2021.00045

Situational Method Engineering (SME). SME provides theory and guidance to build situation-specific, adaptable and configurable methods [7]. The key to success lies in the modularity of the methods, defined in terms of building blocks, and the ability to handle various situations. In this work we apply the SME approach using method chunks as method building blocks [14]. A method chunk is an autonomous part of a method supporting the achievement of a particular software engineering goal. It provides guidelines to reach the goal and defines the artefacts to be used and produced by applying the method chunk. The fitness of the method chunk to a particular situation is specified with a set of context criteria [14]. The composition of method chunks is based on the Map approach [15], which allows to express the method process model in terms of engineering intentions and strategies to achieve the intentions. Several strategies can be defined to achieve an intention, thereby expressing variability and flexibility in the method specification and implementation. Each strategy has to be implemented by at least one method chunk. At each step of the method, the choice of a strategy and method chunks is driven by context criteria. SME has already been used in disciplines as software evolution [16] and open-source adoption [17]. In this paper, we apply a similar approach to DDRE.

III. Research Approach

Our research follows Peffers' *et al.* incremental and iterative approach to Design Science [18], as explained below.

Problem Identification. Many business organizations are increasingly interested in exploiting the potential value and opportunities that digital data sources provide for eliciting requirements continuously and in near real-time. While considering data in the requirements engineering process can provide a number of advantages, it also requires solving complex challenges on the way. We present in Section IV two case studies facing the challenge of understanding how to collect, analyse, and aggregate data using existing techniques and apply DDRE principles such that a) actionable requirements-related insights can be generated, and b) they adhere to the development practices in use. This is a highly complex task and not methodologically considered by current stakeholder-oriented elicitation approaches.

Design. Building on top of the case studies, we synthesise the objectives of the solution. On the one hand, we realized that different companies, or even projects inside a company, build their own approaches to DDRE combining existing techniques and methods in different ways. Furthermore, we also got acquainted that the configuration of the DDRE approach depends on contextual factors whose explicit identification and analysis would add rationale to the definition of the method. Therefore, we argue that SME is the right approach to define the most adequate DDRE method for an organization. We present in Section V the fundamentals of the application of SME for our purposes, introducing a process and a metamodel with correctness conditions.

Development. Once the conceptual approach has been decided to be SME, we elaborate a particular solution that we present in Section VI. We first present the method process map composed of four main intentions, a number of strategies to achieve them and a representative sample of context criteria, elicited from the two case studies. We then provide insights for every intention, illustrating them with some method chunks that are described in terms of the metamodel presented in Section V.

Demonstration and Evaluation. One relevant activity of Design Science is the demonstration of how the designed artefact can be used to address the explicated problem. We provide a demonstration by showing how the problems of the two case studies can be addressed by the designed solution, i.e. how the SME method allows complementing an agile stakeholder-driven approach to data-driven. We also conducted a preliminary, ex-ante evaluation of the proposed solution by doing explorative interviewing with several agile development experts and data scientists of the two companies to describe the envisioned method for the elicitation process and its contextualization, and to get the feedback of how the proposed application of SME can solve or alleviate the practical problems outlined in Section IV.

IV. Problem Identification

We carried out two case studies to identify the problems concerning methodological support for DDRE. In both cases, the success of offered products and services relies heavily on the preferences of customers and end-users, whose number is up to several millions per software product. The case studies showed a high orientation towards the use of digital data for acquisition of requirements and for integrating these into agile development approaches, while each company has its own situation.

A. Case Study: Game Development

Context: The company focuses primarily on developing "endless" historical and grand-themed strategy video games, role-playing games and management simulators.

Methodology: We conducted semi-structured interviews and focus meetings that engaged the company's CTO, as well as Product Owner, Data Analyst, and Developer roles. Closed questions were used to obtain the information presented in Table 1, while open discussion led to insights about data sources, challenges of data collection and analysis, as well as perceptions about the value of automating these activities to increase efficiency of processing and mapping the data to requirements.

Observations: As playfulness and attractiveness of the games is vital, the company is continuously deciding on the right evolution directions for their games. Concerning NL sources for the acquisition of requirements, popular games have a large player base that continuously comments on the game and its features, primarily using online gaming forums and Twitter. A small group consisting of the most dedicated players (approx. 100) are also able to report issues or "bugs" using an issue tracking system. As for machine-sourced data, the playing sequences from the user logs are continuously collected and analysed in order to discover dominant player types, e.g. action-oriented, social, or "lone wolfs". An external eye-tracker service is moreover used for obtaining statistics on player behaviour, i.e. concerning eye gazes, moves, and fixation. Other digital sources, e.g. policy documents on ethics and privacy, are consulted in relation to the development of game features. However, this data is currently processed manually, in an ad-hoc manner. Product owners and developers frequently meet to read and discuss gathered data with the aim of creating request-for-change items corresponding to several user stories, or to improve existing features.

Results: The main business concern is to please the players' requests regarding new features and to minimize negative feedback. Much time is therefore spent on (manually) analysing the collected data: what is requested, whether it comes from influential players, from many players, etc. The company is eager to automate analysis and to create methods that would enable them to gain insights with respect to needed changes and desired features so that they can be timely pushed to development.

B. *Case Study: Online Banking*

Context: Financial services are offered according to various business models; a recent one is online banking, where customers are able to conduct financial activities via the Internet.

Methodology: We carried out interview sessions with the Engineering Manager (following the same structure as in Case Study A), who provided insights concerning the gathering of requirements from digital data sources, reflecting a lack of a structured method for supporting the company's need to optimize development resources, removal of individual biases for requirements prioritization, and the need for more rapid system releases.

Observations: The efficiency of the online baking platform is the major concern to satisfy and grow the customer base. It is therefore critical to obtain customer feedback, which is currently collected by the company's online support unit from emails, telephone calls, and chats. The agile development team also reviews data from social media data (Twitter) on a regular basis. As requests for changes emerge, different disclaimers need to be checked and added from regulatory documents of the company. The machine-sourced data are used for analysing the outcomes of event log monitoring, mainly for discovering suspicious behaviour and patterns related to financial fraud, but not for the assessment of system functionality. As in Case Study A, heterogeneous data are processed manually towards possible tickets for adding or changing user stories.

Results: The company is concerned by a slow and subjective interpretation of the raw data due to a lack of routines and mechanisms for analysing large and similar data, often due to missing or incomplete information on data importance. Another concern is an increasing tendency of refining requirements too upfront (also discussed in [19]), hence the company is eager for more utilization of digital data for requirements evolution.

C. *Summary and Research Question*

We summarize the insights on DDRE from the two cases as:

- be able to automatically collect large amounts of digital data from the sources that are relevant to the business to support a systematic and objective requirements elicitation;
- know how to implement data processing and modelling depending on which digital data sources are targeted;
- collect as complete data as possible to enable identification of data sources, desired behaviour and benefits, to support filling the requirements and motivations behind;
- fully automate the initial processing of the data, such as sentiment analysis and classification, i.e. the tasks where automation would provide high accuracy ;
- use semi-automated processing in further processing (i.e. in aggregation and mapping to requirements) with human-expert involvement and control to ensure that new requirements follow the established product strategy;
- continuously collect, process, and map data to enable refinement of requirements in short iterations to support minimal initial software release and frequent evolution;
- visualize, i.e., create the inventory of the activities being applied in the process of digital data acquisition to be able to analyse it, reuse it, and improve it.

This list of issues concerns the lack of a holistic approach to collecting, analysing, and aggregating relevant big data such that it can be incorporated into their agile approach to requirements development. Hence the main research question addressed in our paper is: *how software development companies can methodologically augment their current agile RE approach in place to complement elicitation by human stakeholders with data-driven, while leveraging skills and resources according to the relevance of the big data?*

TABLE I. DESCRIPTION OF THE TWO CASE STUDY COMPANIES

Business domain	Game development	Online banking
Products	Over 20 strategy games	Savings and investments, loans and pensions
Scope	Global	Serving 4 Northern European countries
Company size	~500 employees	~500 employees
Number of customers	~12 000 000 players	~1 000 000 customers
Data sources	Online internal reviews ("Steam"), forums (Paradox, Reddit, etc.), Twitter, user/system logs, eye-tracking systems	Social media (Twitter, Facebook, Instagram, LinkedIn), online customer service (mails, chatbots, phone-desk), transaction logs, compliance regulations (e.g., privacy)
Types of data	NL- and machine-sourced	NL- and machine-sourced
Relevance of data	Focus on reviews and online forums for obtaining data on bugs and suggestions for improvements; eye-tracking service used for deep issues; Twitter mainly used to promote games (~50,000 followers per game)	Focus on system features from comments on social media
Tools for collection of digital data	Combination of in-house and off-the-shelf tools; collection mainly automated	External data, e.g., from social media, is collected manually
Tools for analysis of data	Primarily manual data analysis, except for eye-tracking; no ML models in production; bug reports, feature requests and change requests	Analysis is done manually; ML techniques are desired, but not yet developed/in use
Method for generating requirements from digital data	No automated processing and classification of different players' data, nor structured mapping to requirements; measures are also manually monitored; team meetings are used to create "requirements items".	No structured or automated processing of digital data and of different data sources; Product Owner calls team meetings to create "tickets" that are mainly epics, leading to requirements for the backlog.

V. Design

We adopt Basili's et al. experience factory concept [20] in our approach. Two main processes arise: *(i)* definition and evolution of the experience base, in our case the method process map and associated method chunk catalogue; *(ii)* application of the experience base in a particular situation. Both processes rely on the conceptual model depicted in Fig. 1, based on [11][16]. The salmon-shadowed part represents the experience base, the blue-shadowed part its application in a company, and the purple-shadowed part is shared by both. Process *maps* are composed of *intentions* connected by *strategies*. *Method chunks* implement a strategy applicable in a certain context represented by the possible *values* of context *criteria*. Chunks produce output artefacts by executing an activity over input artefacts, and are exerted by a number of roles (for clarity, artefacts, activities and roles are represented as *method elements*). Chunks can exhibit *relations*, e.g. conflict and requires, and also structural ones such as specialization and composition. Meanwhile, the *situation* of a *company* is represented by a set of *value assignments* to context *criteria*. As a result, the *method* for a company is composed by a set of method chunks, fulfilling two correctness conditions:

CC1. The value assignment to context criteria in the situation is compatible with the values assigned in the method chunks.
CC2. The selected method chunks fulfil their relations (e.g., all chunks required by selected chunks are also selected).

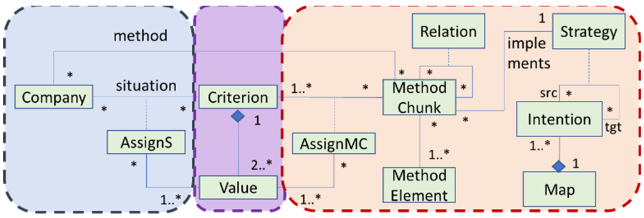

Fig. 1. Conceptual model for the SME approach to DDRE

When a company chooses to apply our SME-based approach to build a method for DDRE, it should apply the following steps:

1) Understanding the company's overall goals and current way of working. The main outcome of this step is the description of the situation (i.e., values that apply to the context criteria).

2) Identification of method chunks already in place in the company. In the general case, the company will be already collecting and processing data, and possibly even aggregating them somehow to support the requirements elicitation process. By exploring the experience base, the method chunks that correspond to these strategies already in place can be identified.

3) Selection of additional method chunks to compose a complete DDRE model. With the help of a recommender system, this step will suggest the method chunks that satisfy the remaining intentions and fulfil correctness conditions CC1 and CC2.

In all these steps, new or evolved method chunks and context criteria can be added to the experience base.

VI. Development

The goal of our method is to generate different requirements items (user stories, epics, quality criteria) in the product backlog (i.e. requirements specification) by semi-automatically collecting, analysing and aggregating big data coming from different sources. We build our method based on recent work [11][16], the literature review reported in Section II, and especially the data collected from companies (Section IV). Although our case study companies are following the agile practices in requirements elicitation, the method is envisioned to be compatible with even other elicitation approaches.

To deal with the modularity, consistency and completeness of the method, we built a multi-dimensional taxonomy of digital data sources, see Fig. 2. Each data source is described on three dimensions: *Producer type*, *Collection means* and *Data format*, which are farther refined into categories and sub-categories. We do not claim it is complete, but rather it proposes an outline for future development; furthermore, it is easily extendible.

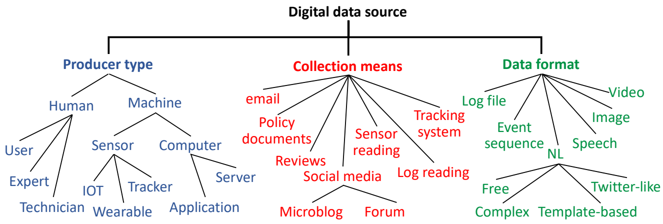

Fig. 2. Taxonomy of digital data sources

We identify four main intentions to be reached to attain the above goal, depicted in Fig. 3 as a process map and described below. We also present in TABLE II. a representative sample of context criteria emerging from our case studies, ready to specify the situation of the involved companies and guide the selection of the method chunks best suited to their situation. Some of them apply at the level of the company (e.g., C1-C4). Others are specific of a subset of intentions (e.g., C5 applies to Collect, while C6 to both Collect and Analyse). Others are applied repeatedly, like C7, which in fact is a family $\{C7_d\}$ for every data source *d* identified in C5 (same for C8). Further experiences will help identifying new criteria, *cf.* Section V.

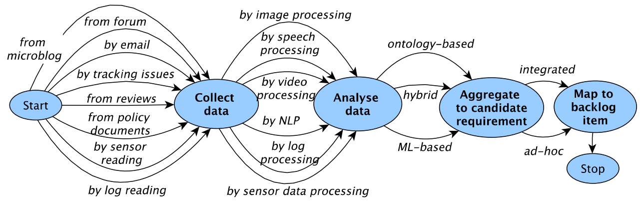

Fig. 3. Method process map

TABLE II. Context Criteria for Elicitation Method Construction

Context criterion	Value
C1. Data relevance towards elicitation	Essential, Valuable, Marginal
C2. Expertise required in the company	High, Medium, Basic
C3. Resources assigned to adopt DDRE	High, Medium, Low
C4. Frequency of releases	Continuous, Frequent, Staged
C5. Available data collection means	Emails, Reviews, … (see Fig. 2)
C6. Format of available data	Log file, Image, … (see Fig. 2)
C7. Amount of data per data source	Very high, High, Medium, Low
C8. Data quality per data source	Excellent, High, Fair, Poor

Collect Data. As there are many different digital data source types, each of them requiring a particular data collection approach, the method process map includes one strategy per data *Collection means* (see Fig. 2). Therefore, the method should provide at least one method chunk per data collection means; see

TABLE III. for an example. If a given means can collect several *Data formats*, a method chunk per data format will be required.

TABLE III. EXAMPLE OF METHOD CHUNK FOR DATA COLLECTION

Method chunk	Content
Name	Collect forum posts
Intention	Collect data
Strategy	From forum
Goal	Extract actionable information from users' comments in online forums
Input	Stream of forum posts, including: forum user, time stamp, number of likes, the reaction score of the user
Activity	Collect data programmatically using the location and the API of the forum; automated and continuously
Output	Filtered stream of posts mentioning the company
Roles	Data Engineer
Criteria	C2 = Medium, C3 = Low, C5 = Forum, C6 = Free NL
Related chunks	--

Analyse Data. The aim here is to provide a tool-assisted data analysis. Different data types require different analysis methods and tools. Based on the classification of digital data source types shown in Fig. 2, we define the strategies to analyse collected data, one per *Data format*, e.g., *by NLP*, *by log processing*, … (see Fig. 3). Each strategy potentially offers several method chunks, as data processing and analysis depends not only on the data type but also on its quality and format. For example, as a consequence of data variety, not all NL data can be analysed in the same way. Forum posts will likely require using different ML models than policy documents. This, in turn, entails that several specializations of the NLP method chunk have to be defined. ML techniques are commonly used for analysing the collected data. This is currently emphasized in the method process map, but one could apply different types of approaches for processing the collected data, e.g. supervised/unsupervised ML or rule-based (heuristic) approaches.

TABLE IV. EXAMPLE OF METHOD CHUNK FOR DATA PROCESSING

Method chunk	Content
Name	Sentiment analysis
Intention	Analyse data
Strategy	By ML
Goal	Understand the sentiment behind a given opinion
Input	NL text, forum post type, a paragraph size
Activity	Use (i.e. create if it does not exist) a pre-trained ML model to automatically identify, extract, and quantify expressions of sentiment in free-text
Output	NL text labelled with emotions
Roles	Data Scientist (NLP expert)
Criteria	C2 = High, C3 = Low, C6 = NL, C7 = Very high, C8 = High
Related chunks	<<requires>> Collect NL data

Aggregate to Candidate Requirement. This step requires to align data obtained from different sources. For example, NL data from reviews can be combined with the behaviour logs recorded from an eye tracker using named entity recognition for identifying top-used [21]. Existing techniques for data aggregation are based on ontologies and/or ML, yielding to *ontology-based*, *ML-based* and *hybrid* strategies. Beyond the technical approach adopted, other aspects may be considered, for instance, whether the two data sources are merged [10] or used sequentially [12], or whether the approach provides some facilitation aspects as software analytics tools [22]. All these aspects should be captured by new strategies and method chunks.

Map to Backlog Item. The aggregated candidate requirements have to be mapped into the product backlog items as defined in agile methods, i.e. user stories, epics, quality criteria, constraints. We see here two mapping strategies, *integrated* and *ad-hoc*. The *integrated* strategy smoothly aligns with the previous steps so that all the process can be contemplated as a holistic cycle. Examples of method chunks are given in [13][23], where the emerging requirements are finally stored in a backlog. The *ad-hoc* strategy refers to usual requirements elicitation techniques that may benefit from the collected and aggregated data but still are classical in this sense. For instance, [24] proposes gamification involving key stakeholders into a collaborative endeavour to transform and prioritize users' feature requests into requirements. Automation of the mapping task is demanding for achieving and will probably require human interaction [25].

VII. EVALUATION

In the first Design Science cycle, we evaluated our SME-based approach to DDRE in two different ways (*cf.* Section III).

First, we re-examined the two case studies described in Section IV to evaluate the feasibility of the different elements of our solution, presented in sections VI and VII. Our case studies showed that not all companies use the same types of digital data source; probably no single company uses all of them. The modularity of SME allowed selecting only those method chunks that are pertinent for each company, i.e. fitting the company's situation. Given the context criteria that apply to data collection, remarkably C6 to C8, per each data source type we recommend to assess the relevance of the data towards requirements elicitation and whether the amount and quality of data is sufficient.

As illustration, the gaming company is, as described for that case, collecting the data from several sources, while players' forum blogs, and eye-tracking logs are essential for them; i.e. for the company relevance is a vital factor due to the strategic focus of satisfying the user base. In contrast, for the banking company, the cost of implementing in addition to relevance is important, and the company chooses to set the essential focus to microblogs, and policy artefacts; and e-mails are also important because the customers are used to this means for asking or reporting some issues. The gaming company would need NLP in combination with ML for efficient processing and analysis of massive forum posts, while different programmatic techniques for the analysis of the quantitative data obtained from eye-tracking systems for recognizing playing behaviour patterns.

In more detail, for the gaming company, it is very important to, upon forum complaints as "...to find the button for a detailed map is a 'mission impossible'...", or "I am trying...but the button is simply not there!", understand by the eye-tracking logs how long it takes to find the button; therefore, the aggregation of these data becomes relevant to objectively decide for going towards a request for requirement change or not. SME allows combining the appropriate strategies to implement this intention.

In a second step, we conducted exploratory interviews with the studied companies to obtain an ex-ante opinion about our proposal. After we presented them the purpose of SME, we decided to focus the interviews on the adequacy of the context criteria (see Table II), because they are key to success for the approach: if context criteria are unclear or do not discriminate significantly among different strategies to achieve intentions, our approach will not deliver appropriate results. Below we summarize the main highlights emerging from our interviews:

- All the suggested criteria are relevant;
- C1 is the first criterion that should be used to determine if DDRE is valuable for a company/software product;
- C2 and C3 could be further refined in terms of possible categories of skills and types of resources;
- C4 is important to know to be able to catch "hot" requirements on time;
- C5 is very important to determine as a number of data sources are not (directly) available, or they have some constraints for access;
- C6 provides valuable information on the techniques and tools that should be prioritized;
- C7, C8 relate to essential parameters; e.g. volume of data determines the needed processing and storage resources.

As some possibly missing aspects, the respondents emphasized that the determination of the relevant data sources could be extended from the company to the entire business sector to be able to collect even the data of similar businesses. We also got some interesting remarks, e.g., *(i)* sources with high volume [of data] are more important typically than those with smaller volume because they say "more", *(ii)* the need to handle constraints in some data sources in terms of privacy and other legal issues.

VIII. Conclusions and Future Work

In this emerging research paper, we have laid the foundations for the systematic design of context-aware DDRE methods, for guiding the collection of data from heterogeneous digital sources, processing, aggregation, and mapping to system requirements. The research is technically based upon SME principles, and methodologically based upon design science, for which we have presented the first cycle.

Our study aims to provide business organizations *a systematic automated aid* for dealing with the complexity of digital data management, according to different business conditions, available and prioritized data sources, and different techniques for data processing, analysis, and predictions. Applying SME for this aim makes our research novel and, most importantly, adaptable to a variety of business contexts. Despite the fact that our case study companies are following the agile practices in requirements elicitation, the method is envisioned and planed for further development to be compatible with even other elicitation approaches and frameworks.

Future work is framed in the second design science cycle. We plan to evolve the current proof-of-concept artefact by: *(i)* involving more software companies both to better scope the problem and evaluate the solution, *(ii)* elaborating a more complete experience factory (method chunks and context criteria), *(iii)* developing a recommender system to apply our method in an organization (and thus conduct a proper evaluation).

Acknowledgment

The work presented in this paper is partially funded by the DOGO4ML Spanish research project, PID2020-117191RB-I00.

References

[1] K.S. Rubin: Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley, Upper Saddle River, 2012.

[2] W. Maalej, M. Nayebi, T. Johann, G. Ruhe: Toward Data-Driven Requirements Engineering. IEEE Software 33(1), 2016, pp. 48–54.

[3] F. Dalpiaz, N. Niu: Requirements Engineering in the Days of Artificial Intelligence. IEEE Software 37(4), 2020, pp. 7–10.

[4] J. Dąbrowski, E. Letier, A. Perini, A. Susi: Mining User Opinions to Support Requirement Engineering. CAiSE 2020, pp. 401–-416.

[5] M. van Vliet, E.C. Groen, F. Dalpiaz, S. Brinkkemper: Identifying and Classifying User Requirements in Online Feedback via Crowdsourcing. REFSQ 2020, pp. 143–159.

[6] S. Bühne et al. IREB Certified Professional for Requirements Engineering – Foundation Level – Syllabus. Available at https://www.ireb.org, 2020.

[7] B. Henderson-Sellers, J. Ralyté, P. Ågerfalk, M. Rossi: Situational Method Engineering. Springer, 2014.

[8] X. Franch: Data-Driven Requirements Engineering: A Guided Tour. CCIS 1375, 2021, pp. 83-105.

[9] S. Lim, A. Henriksson, J. Zdravkovic. Data-Driven Requirements Elicitation: A Systematic Literature Review. SN Computer Science 2/16, 2021.

[10] M. Oriol *et al.*: FAME: Supporting Continuous Requirements Elicitation by Combining User Feedback and Monitoring. RE 2018, pp. 217-227.

[11] A. Henriksson, J. Zdravkovic: A Data-Driven Framework for Automated Requirements Elicitation from Heterogeneous Digital Sources. PoEM 2020, pp. 351-365.

[12] D. Wüest, F. Fotrousi, S. Fricker: Combining Monitoring and Autonomous Feedback Requests to Elicit Actionable Knowledge of System Use. REFSQ 2019, pp. 209-225.

[13] M. Oriol *et al.* Data-driven and tool-supported elicitation of quality requirements in agile companies. Software Quality Journal 28, 2020, pp. 931–963.

[14] I. Mirbel, J. Ralyté: Situational Method Engineering: Combining Assembly-based and Roadmap-driven Approaches. Requirements Engineering Journal 11, 2006, pp. 58–78.

[15] C. Rolland, N. Prakash, A. Benjamen: A Multi-Model View of Process Modelling. Requirements Engineering Journal, 4, 1999, pp. 169–187.

[16] X. Franch *et al.*: Situational Approach for the Definition and Tailoring of a Data-Driven Software Evolution Method. CAiSE 2018, pp. 603-618.

[17] L. López *et al.*: OSSAP - A Situational Method for Defining Open Source Software Adoption Processes. CAiSE 2016, pp. 524-539.

[18] K. Peffers, T. Tuunanen, M. Rothenberger, S. Chatterjee: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 24(3), 2007, pp. 45–77.

[19] R. Kasauli, E. Knauss, J. Horkoff, G. Liebel, F. de Oliveira: Requirements Engineering Challenges and Practices in Large-Scale Agile System Development. Journal of Systems and Software 172, 110851, 2021.

[20] V. Basili, G. Caldiera, D. Rombach: Experience Factory. In Encyclopedia of Software Engineering, J.J. Marciniak (Ed.), 2002.

[21] J. R. Finkel and C. D. Manning: Nested named entity recognition. EMNLP 2009, pp. 141-150.

[22] S. Martínez-Fernández *et al.*: Continuously Assessing and Improving Software Quality With Software Analytics Tools: A Case Study. IEEE Access 7, 2019, pp. 68219-68239.

[23] G. Williams, A. Mahmoud: Mining Twitter Feeds for Software User Requirements. RE 2017, pp. 1–10.

[24] P. Busetta *et al.*: Tool-Supported Collaborative Requirements Prioritisation. COMPSAC 2017, pp. 180-189.

[25] J. Zdravkovic, E.-O. Svee, C. Giannoulis: Capturing Consumer Preferences as Requirements for Software Product Lines. Requirements Engineering Journal 20, pp. 71–90.