A Requirement Engineering Model for Big Data Software

Hamza Hussein Altarturi, Keng-Yap Ng *, Mohd Izuan Hafez Ninggal, Azree Shahrel Ahmad Nazri, Abdul Azim Abd Ghani
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia, Selangor, Malaysia
altarturih@gmail.com, kengyap@upm.edu.my, mohdizuan@upm.edu.my, azree@upm.edu.my, azim@upm.edu.my

Abstract— **Most prevailing software engineering methodologies assume that software systems are developed from scratch to capture business data and subsequently generate reports. Nowadays, massive data may exist even before software systems are developed. These data may also be freely available on Internet or may present in silos in organizations. The advancement in artificial intelligence and computing power has also prompted the need for big data analytics to unleash more business values to support evidence-based decisions. Some business values are less evident than others, especially when data are analyzed in silos. These values could be potentially unleashed and augmented from the insights discovered by data scientists through data mining process. Data mining may involve overlaying and merging data from different sources to extract data patterns. Ideally, these values should be eventually incorporated into the information systems to be. To realize this, we propose that software engineers ought to elicit software requirements together with data scientists. However, in the traditional software engineering process, such collaboration and business values are usually neglected. In this paper, we present a new requirement engineering model that allows software engineers and data scientists to discover these values hand in hand as part of software requirement process. We also demonstrate how the proposed requirement model captures and expresses business values that unleashed through big data analytics using an adapted use case diagram.**

Keywords—big data requirement; requirement model; actionable intelligence; big data system; data scientist model.

I. INTRODUCTION

The focus on data in information systems is rapidly increasing nowadays due to the vast amount of data generated by software systems. Youtube, for example, had more than a billion users in 2015, who watched 4 billion videos every day and uploaded 300 hours of video every minute [1]. In this hurricane of data generated from the usage of software applications, software engineers need to be aware of what Big Data is, how to develop Big Data software and what the requirements of Big Data software are. As a general definition, big data are data with 5V's; variety, velocity, volume, veracity, and value [2]. Taking big data into account in software engineering will enhance the quality of software and exploit the values of such data that would otherwise be wasted. When appropriately exploited, the business values (or simply values) of big data can affect the systems positively in several ways. For example, it enhances the quality of big data software and helps make accurate predictions regarding day to day business which may assist in informed decisions to maximize the quality of services and profit. Moreover, with the advent of Internet of things (IoT), cloud computing and machine learning techniques have also indirectly enabled software systems to process big data, and subsequently unleash and exploit business values by learning from their previous activities in order to manage future activities better, and subsequently optimize their business goals. This is evident in the case study explained in section IV.

Despite being well-established, the traditional software engineering is facing both technical and organizational challenges when big data systems are in question. The challenges are: (1) technical challenges in developing big data systems that address the 5Vs of big data (i.e., Volume, Velocity, Variety, Veracity and Value) and (2) organizational challenges surrounding how data scientists can work with software engineers effectively to discover and maximize the values of big data [4]. In this paper, we address the organizational challenges. We propose a new model that organizes and coordinates the process of requirement engineering in a collaborative manner between the software engineers and the data scientists. The model is further explained in Sections III and IV. Specifically, we describe how the conventional use case model can be merged with the newly proposed data scientist model. This integrated model explains how to specify requirements on business values in big data software systems.

Having established the importance of big data in software systems, we address the following questions: how different are traditional systems (i.e., systems that do not generate and implement big data analytics) from big data systems? From the software engineer perspective, does the difference necessitate a new model for the software engineering process or does the traditional software engineering process suffice? How to define the requirements of big data systems and are they different from the requirements of traditional systems? How can the requirements be documented and described using UML (Unified Modelling Language) diagrams? These questions are answered in the remainder of this paper. We also highlight that big data systems are in fact different from traditional systems. Moreover, we illustrate how the proposed big data requirement model is capable of spotting and exploiting the values in big data. We also describe the how the traditional requirement process is adapted to account for big data.

II. HOW DIFFERENT BIG DATA SOFTWARE ENGINEERING IS?

The challenges that big data software engineering facing has rendered some existing software engineering models insufficient to fulfil the requirement process. With respect the organizational aspect, some traditional software engineering methodologies may fail in (1) coordinating and organizing work between the data scientist and the software engineer [4], and (2) adding the requirements for business values of big data in the model. These methodologies are usually user centric rather than data centric. In other words, most traditional requirement methodologies focus on requirements

978-1-5386-0790-9/17/$31.00 ©2017 IEEE

apparent to users rather than potential features proposed by data scientists after big data analytics to discover patterns and insights. Often, these features are not apparent to users, however, sometimes they could be a potential game changer in business.

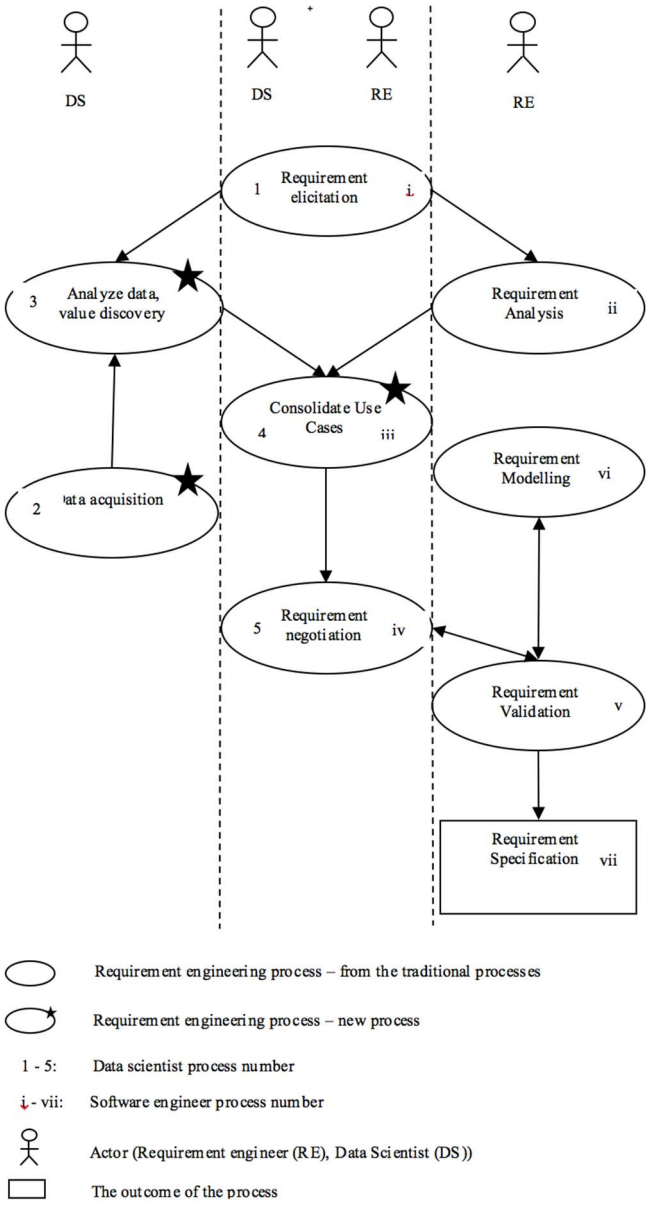

Figure 1: Illustration of the proposed big data requirement engineering processes

Ideally, a software developer should discover, analyze, and document the business values of big data requirements, then incorporate requirements in system design. Both requirement engineers and data scientists are equipped with different sets of skills. Hence, requirement engineers require the help from data scientists to discover those business values before they can be documented. Unfortunately, the traditional requirement engineering process does not support this activity. Moreover, in the requirement elicitation process, usually software engineers gather requirements from users whom may not be aware of the potential of business values from big data.

In our proposed model, we assume that big data requirements are discovered in the early stage of the software requirement process by data scientists. The process will result in the identification of business values that it can be gained from big data and subsequently produce actionable intelligence that should be implemented in the software system. The technical how will be described in the next section.

III. BIG DATA REQUIREMENT ENGINEERING PROCESS

"Requirements Engineering is, as its name suggests, the engineering discipline of establishing user requirements and specifying software systems" [4]. Requirement engineering involves analyses to identify what users want from a software system, and to understand what their needs mean in terms of software design and architecture. However, "It's not the customer's job to know what they want" – Steve Jobs [24]. It should be the data scientists' in our context. Apart from asking customers or users what they do want and expect in their software, big data software requirements may discover business values which can be unleashed by big data analytics and then duly pitched to them.

To elicit requirements for big data software, it is necessary to define the objectives of the project at earlier stages. Time and cost reduction, finding insights from data and optimizing decision making are the most common goals that make an organization to appreciate big data software projects. In addition, to understand user requirements, these projects may also need other technologies like machine learning. Most of these technologies are defined in a technology driven manner which makes a little input from users [5]. As compared to the conventional requirement engineering, big data requirement engineering consists of additional processes such as data acquisition, finding business values and consolidating use cases as shown in Figure 1.

The proposed big data requirement model has three different types of processes. The first type will be done by the requirement engineer. The second type will be done by the data scientist. The third type will be done collaboratively by both the requirement engineer and the data scientist. The major organizational challenge that the big data software engineering facing is the third type [3]. All these processes are illustrated in detail as follows:

A. Requirements elicitation

In the traditional requirement processes, requirement elicitation is an activity that software engineers carry out together with customers and software end-users to define the application domain, what services the software should provide, the required performance of the software, hardware constraints, and so on [7]. In order to elicit requirements effectively, good communication between users and the requirement engineer should take place [13]. A role for the data scientist should be injected in this communication to define requirements that comply with big data requirements. Meaning that, to elicit big data requirements, the data scientist should also join the requirement engineer and the customers to identify data in user requirements which can be used to generate business values and subsequently enhance the user requirements.

B. Data acquisition (New Process)

The need for big data collection cannot be fulfilled simply by applying the traditional methods of data collection because of the unique characteristics of big data (volume, velocity, variety, veracity, and value) as well as the cost of conversion between data formats [8]. In this step, the data scientist decides on what data must be kept and what data must be

discarded because part of the data will be useless and should therefore be dismissed. These decisions will be done by the following three data acquisition methods which are: collecting method of system log, collecting method of network unstructured data, and other data acquisition methods. These methods are proposed by Liu, Yang and Zhang [9]. The output from these processes is a definition of data to be used for knowledge discovery. This process shall be performed by the data scientist.

C. Requirement analysis

The purpose of the Requirements Analysis Process is to transform stakeholder's requirement-driven view of desired services into a technical view of a required product that could deliver those services [9]. Conflicts between the requirements will be detected and resolved in this process. The boundaries of the system should also be defined in this process. The environment that interacts with the system is important to be defined and explained in this process. In the proposed big data requirement engineering model, this process will be done in parallel with the process of finding business values and data acquisition by the requirement engineer.

D. Data analysis and value discovery (New Process)

Big data will be useful only if it has the potential for revealing information or discovering knowledge. This is why big data have to be analyzed. Knowledge discovery, in short, is a nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [14]. Using advanced techniques such as machine learning, data mining, text analytics, predictive analytics and natural language processing of big data allows analysts and managers to make timely and informed decisions. Knowledge discovery consists of multiple and varying processes and analytical steps (refer to Fayyad [14] for a detailed description). The data scientist is the one who is responsible for this process, including the selection of the data analysis techniques.

The result of this process will be represented as a data scientist model, which is described in Section IV. This model forms the collaboration part between the data scientist and the software engineer. The collaboration process is described in the next section (Use Cases Consolidation).

E. Use cases consolidation (New Process)

Consolidation is the process of merging the part of the model done by the software engineer with those done by the data scientist. The result of this consolidation will be represented as an Actionable Use Case Diagram, which is explained in detail in Sections IV. The actionable use case diagram allows business values unleashed from data to be depicted in the data scientist's model, together with their roles in the software and their interactions with other software components. Once the values are defined and presented, the consolidated model is ready to be presented and discussed with the customers. This process will be done by both the software engineer and the data scientist.

F. Requirement negotiation

In the traditional requirement engineering process, the requirement negotiation happens at the beginning of the requirement engineering process [10], followed by the elicitation and analysis processes [7]. In big data requirement engineering, however, this process comes after the models by the requirement engineer and the data scientist are consolidated. Performing the processes in this particular order helps the customer to understand the actionable intelligence that can be achieved using the big data. The result of this process is a set of negotiated requirements. The requirements can be validated after this process. This process will be performed by the software engineer and the data scientist with the customers.

G. Requirement modelling

In this process, the consolidated model will be finalized to yield the finalized Actionable Use Case diagram before validation. This helps the software engineer to document the requirements during the specification step. The requirement modelling will be performed by the requirement engineer.

H. Requirement validation

Requirement validation is the process of checking whether requirements indeed define what the customer really wants from the system [7]. In the proposed model, this process comes after negotiating the requirements and consolidating the use case diagram. This process will be performed by the requirement engineer.

I. Requirement Specification

Once the requirements have been validated, the user and software requirements can be written in a requirements document [7]. In this process, unlike the traditional requirement engineering model, the values of the data will be specified and documented using an Actionable Use Case diagram. In this way, the values of big data (one of the 5'Vs characteristics of big data) in the software being engineered can be defined and presented. This process will be performed by the requirement engineer.

IV. ACTIONABLE USE CASE FOR BIG DATA SOFTWARE

In the previous section, we have shown how some new requirement engineering processes could help finding values for the big data software. These values, which can be found by data scientist, will help to improve the software capacity and features. However, after finding the values, significant questions should be answered.

Firstly, how to represent the values found? The reason to represent values is to ensure that they can be delivered in the requirement engineering phase to the stakeholders. This step is crucial to allow customers to understand those values in their software. After that, the customer can study this value in an economical manner. Moreover, this representation of the values will help the data scientist, software engineer, and the customer to negotiate the values and their roles in the big data software. Secondly, how to document these values? The documentation of these values is not less important than other documentations in software engineering. Documentation is essential for any software [7]. In particular, good documentation would be handy in the maintenance process.

After identifying the business values, Actionable Use Case (AUC) diagrams for big data software will be presented. These diagrams play an important role in representing the values and document them collaboratively by both of the data scientist and the software engineer. The AUC diagram is a consolidation of data scientist model and the traditional use case. The consolidated model explains the following items:

1. Actor (manager, user, system or website).
2. Actionable intelligence to be presented.
3. Data scientist model, which contains:
    a. Actionable intelligence.

   b. The algorithms and the analysis methods to get the actionable intelligence.
   c. The data source used in the analysis and algorithm
   d. The "Rely on" relationship between the actionable intelligence and data scientist model.
4. The relationships between actor and actionable intelligence.
5. The relationships between data source and analysis and algorithm.

Figure 2 shows an example of the new model including all the aforementioned items:

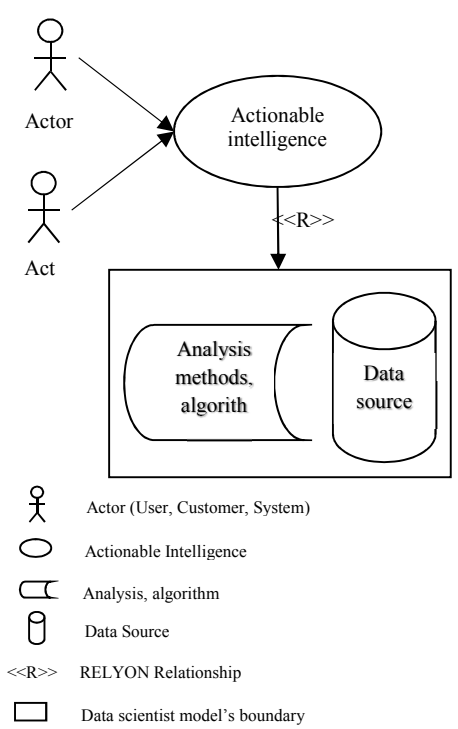

Figure 2: an actionable use case diagram

V. CASE STUDY: RECOMMENDATION SYSTEM

In this section, a recommendation system is introduced and studied. Recommendation systems are software tools and techniques that provide suggestions for items which are potentially of use for a certain user [11]. These suggestions rely on big data mining. A recommendation system is chosen because it is easy to understand and analyzed as compared to other big data systems.

A. Over view of the case study

Most of recommendation systems are personalized. Personalized recommendation systems allow different users to receive different suggestions. These suggestions will be provided according to the behaviour of individual users. Basically, the suggestions are big data driven. Whilst, non-personalized recommendation systems [11] are simpler than the personalized ones because they rely merely on the ranking of items. Non-personalized recommendation systems use, collect, or generate the same data as the personalized, but they do not use these data and, therefore, do not exploit the values of big data. In this case study, we only focus on personalized recommendation systems because they feature the characteristics of big data.

Recommendation systems provide several functions and services to the customers. These services aim at maximizing benefits of companies through [16]:

- Increasing the number of items sold
- Selling more diverse items
- Increasing user satisfaction
- Increasing user fidelity
- Better understanding of what the user wants

These functions rely on data collected from users to study their behaviour. The data include for example [12]:

- When the user pauses, rewinds, or fast forwards
- What day the user watches content (e.g., people usually watch TV shows during the week days and movies during the weekends)
- The date on which the user watches a certain content
- The time at which the user watches a certain content
- The address where the user watches a certain content (zip code)
- The device on which the user watches contents
- When the user pauses on and leaves a content (and whether they ever come back)
- The ratings given by the user for different contents
- The user searches
- The user's browsing and scrolling behaviour.

These collected data could be used to generate values for the companies to enhance their recommendation system. However, the data and values can only be generated and exploited if the system has been analyzed and engineered with respect to requirement of big data systems. We will apply both the traditional requirement engineering model and the proposed big data requirement engineering model on a recommendation system for movies and compare their results.

Algorithmically, the recommendation system uses two types of recommendation algorithms which are: non-personalized and personalized as previously mentioned. Non-personalized algorithms are based on top rating items, top selling items, and Content Filtering [17]. Personalized algorithms, on the other hand, are based on Collaborative Filtering [18], Slope One [19], and Content Filtering [17]. Matrix Factorization (MF) [20] and Restricted Boltzmann Machines (RBM) [21] are examples of personalized recommendation algorithms used in Netflix [22]. These algorithms are performed on big datasets. The goodness of fit of the results could be evaluated by root-mean-square error (RMSE). RMSE is "a frequently used measure of the differences between values (sample and population values) predicted by a model or an estimator and the values actually observed"[1].

Data filtering is important in recommendation systems because it affects the results of the algorithms. There are two data filtering techniques: content filtering and collaborative filtering [23]. Content filtering is based on the items of metadata such as movie's actors, movie's director, and the categories of movies. Collaborative filtering is based on assumptions, predictions, and testing many of users to infer

[1] https://en.wikipedia.org/wiki/Root-mean-square_deviation

an individual user [23]. In order to reduce RMSE, both Collaborative Filtering and Content Filtering are used in the recommendation system.

B. Evaluation

The case study described in previous section is used here to test the effectiveness of the proposed model for big data requirement engineering. As mentioned in Section III, applying the traditional requirement engineering may not be able to discover values from data in the recommendation system at hand.

Kotonya and Sommerville [7] suggested a conceptual linear requirement engineering process model. The processes are requirement elicitation, requirement analysis and negotiation, requirement documentation, and requirement validation. In the elicitation process, the functional requirements of the recommendation system will be defined. The main functional requirement in the system considered in the case study is viewing recommendations for the user. Then, the requirements are analyzed and negotiated with the customer. After the negotiation is completed, requirements are documented using diagrams and forms. One of these diagram is the use case diagram. Once the requirements have been documented, they will be validated and proper changes

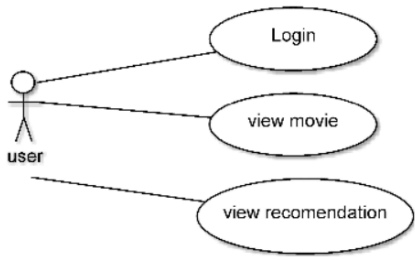

Figure 3: a use case diagram for the recommendation system using the traditional requirement engineering process

will be made. The result of this process is a part of the requirement specification of the system (see Figure 3).

In contrast, by using big data requirement engineering model developed in this study (Section IV), the data scientist will be involved in very early stage in the requirement engineering processes. In the first process, which is the requirement elicitation, the requirement engineer and the data scientist elicit the requirements of the recommendation system. After that, the data scientist will start the data acquisition process. In this process, the data scientist will decide which data to be collected for the recommendation system. In our case study, the data scientist will collect the aforementioned user behavior as shown in [12].

The data on the users' behavior will be analyzed to find the values in the system. Netflix has provided a dataset of 100,840,507 ratings from 480,189 users who rate 17,770 movies [15]. The dataset contains four parameters which are: DATASET<user, movie, date of grade, grade> [15]. The user and movie fields are integer containing the ID's of the users and movies. The date of grade field is the date the user rated the movie. The grade field is integer containing a number between 1 (lowest rating) and 5 (highest rating).

In this process, filtering techniques and algorithm types are also specified and documented. The data scientist will define the values of these collected data using the data scientist model which is the recommendations. See Figure 4. This model presents the algorithms and the filtering techniques that the data scientist has chosen. In our case, the data scientist model will be defined as follow:

- The values (actionable intelligence, knowledge) are: view recommendations
- Data sources are: user behaviors, dataset<user, movie, date of grade, grade>
- Algorithms and techniques are: Matrix Factorization (MF), Restricted Boltzmann Machines (RBM), Collaborative Filtering, Content Filtering
- A "Rely" relationship is defined between the data source and the algorithms and techniques.

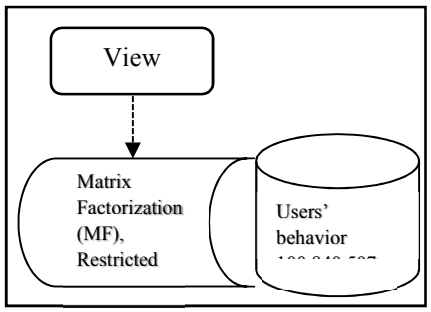

Figure 4: data scientist model for recommendation system

Meanwhile, the functional requirements will be analyzed and defined by the requirement engineer in the requirement analysis phase. In addition to that, the requirement engineer will also define the use case diagram for the software. In our case, the use case diagram defined by the requirement engineer is shown in Figure 3.

After that, both the requirement engineer and the data scientist will consolidate their diagrams in the consolidation process. This process will generate the actionable intelligence use case diagram. To consolidate the two diagrams, the software engineer and the data scientist will match the actionable intelligence from the data scientist model with one or more use cases. In our case, "View Recommendation" actionable intelligence from the data scientist model matches the "View Recommendation" use case, so they will be consolidated as shown in Figure 5.

This consolidation and generation of the actionable use case will be done according to some rules. These rules are explained in detail as follows:

- At least one use case in the data scientist model should be associate to at least one use case from the requirement engineer's diagram.
- The actionable intelligence from the data scientist and the use case from the use case diagram to be consolidated should have some common activities. For example, in our case study, finding the lowest cost of two activities in the software which are shipment and delivering. These two activities are in two use cases in the use case diagram which are shipment packages and delivering packages.
- The relationship: <<Rely>> relationship from the use case to the data scientist's model. The <<Rely>> relationship is used here because the use case relies on the data and the algorithms have been used in the data scientist model.
- The actors and any use case related to the use case to be consolidated should remain the same after consolidation process.

The actionable use case (Figure 4) shows clearly how to use big data in the software. In addition, the diagram

facilitates documenting these actionable intelligences to help the developer during the software development. Also, any maintenance of the software or changing of the data or algorithm can be done easily by modifying the actionable use case.

As compared to the traditional use case diagram, we can

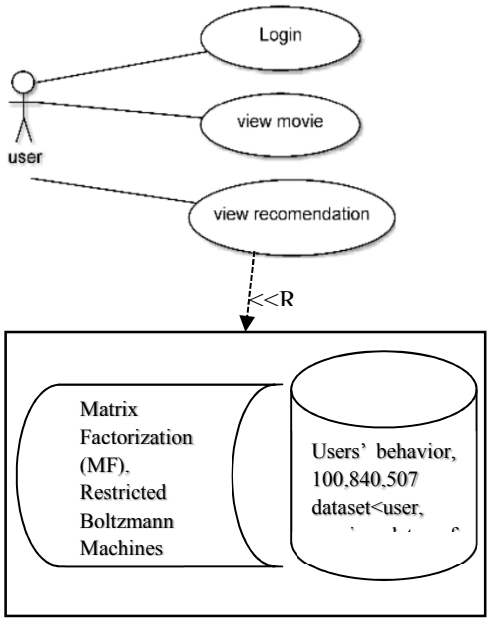

Figure 5: actionable use case diagram using big data requirement process

notice that the traditional use case is unable to present or document the values in the big data software. In contrast, the actionable use case was actually able to present and document the values discovered by the data scientist. Finding these values requires an experienced data scientist who can collect, analyze, and disseminate the data. These activities allow the data scientist to perform business domain specific and statistically sophisticated analytics. These tasks are typically difficult to solve by a software engineer.

VI. CONCLUSION

Requirement engineering for big data systems is significantly different from requirement engineering for conventional systems. This difference requires a new model and underlying processes for big data requirement engineering. In this research, we proposed a model for big data requirement engineering that facilitates the identification of the potential values and needs of big data analytics in the system being developed. The model involves software engineers and data scientists in an interactive and collaborative manner to ensure that big data is well accounted for during the software engineering process. The proposed model has also been evaluated by applying it on a case study i.e.: a recommendation system. The use of this model in big data software engineering therefore help exploit the big data generated or used by the system and turn them into useful information to enhance the users' experience, optimize the systems' utility, and consequently maximize profit.

Although the model proposed in this study has been evaluated qualitatively through a case study, an empirical evaluation is also important. Evaluation of the model in an empirical manner requires its application in a real life big data software projects to collect data about: (1) model performance metrics (effectiveness of the model) and (2) big data generated from the system and its added values for the business. Then, these data can be tested and the model can be evaluated empirically. This empirical evaluation is one of our future works.

In addition to the empirical evaluation, we anticipate that other UML diagrams (sequence diagram, activity diagram, state diagram, etc.) can be enhanced and upgraded besides the use case diagram to support big data software. For example: the enhanced sequence diagram, the activity diagram, the state diagram, etc. shall correspond to the injected actionable intelligence and data scientist's role. The enhancement would help the software engineer to represent and document the values in a collaborative manner with the data scientist. These future works may make remarkable development in big data software engineering.

AKNOWLEDGMENTS

This work is financially supported by Universiti Putra Malaysia and Fundamental Research Grant Scheme (FGRS), Ministry of Higher Education, Malaysia.

We also like to express our gratitude wholeheartedly to anyone who have involved directly or indirectly in this research, especially to our research assistants, Ms. Zairin Mhd Lajis and Hazuraini Mohd Suffian Soo for their passion and dedication this research.

Besides, this work would not be possible without the help and support from fellow academic and supporting staff here at the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. Thanks.

REFERENCES

[1] Gorton, I. Bener, A. B. and Mockus, A. (2016). Software Engineering for Big Data Systems. IEEE SOFTWARE. pp: 32-35.

[2] Bagriyanik, S., & Karahoca, A. (2016). Big data in software engineering: A systematic literature review. Global Journal of Information Technology, 6(1). IEEE

[3] Eridaputra, H. Hendradjaya, B. and Sunindyo, W. D. (2014). Modeling the Requirements for Big Data Application Using Goal Oriented Approach. School of Electrical Engineering and Informatics, Indonesia

[4] Chen H. M., Kazman R., Haziyev S. (2016) Agile Big Data Analytics Development: An Architecture-Centric Approach. Hawaii International Conference on System Sciences.

[5] Sutcliffe A. (2002). User-Centred Requirements Engineering. Springer.

[6] Lau, L. Yang-Turner, F. and Karacapilidis, N. (2014). Requirements for Big Data Analytics Supporting Decision Making: A Sense making Perspective. Springer Science & Business Media, 49 -70.

[7] Sommerville I. (2011) SOFTWARE ENGINEERING - 9th ed.

[8] Liu, Z. Yang, P. and Zhang, L. (2013). A Sketch of Big Data Technologies. Seventh International Conference on Internet Computing for Engineering and Science, 26 -29.

[9] Systems and software engineering —System life cycle processes

[10] Beohm B., Egyed A. (1998) Software Requirements Negotiation: Some Lessons Learned. IEEE.

[11] F. Ricci, L. Rokach and B. Shapira, Introduction to Recommender Systems Handbook, Springer Science+Business Media, LLC 2011.

[12] https://blog.kissmetrics.com/how-netflix-uses-analytics/
[13] Abran, A., Moore, J. W., Bourque, P., Dupuis, R., & Tripp, L. L. (2004). Guide to the software engineering body of knowledge: 2004 version SWEBOK. IEEE Computer Society.
[14] Fayyad, U. M. (1996). Data mining and knowledge discovery: Making sense out of data. IEEE Expert: Intelligent Systems and Their Applications, 11(5), 20-25.
[15] BI, W., & WANG, W. (2011). Walking in Netflix: A Case Study of Collaborative Filtering for Social Media Recommendation System.
[16] Cabacas, R., Wang, Y., & Ra, I. H. (2014). Qualitative Assessment of Social Network-Based Recommender Systems Based on Essential Properties (pp. 1-11). Springer.
[17] Zeng, C., Xing, C. X., & Zhou, L. Z. (2003). A personalized search algorithm by using content-based filtering. Journal of Software, 14(5), 999-1004.
[18] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295). ACM.
[19] Lemire, D., & Maclachlan, A. (2005, April). Slope one predictors for online rating-based collaborative filtering. In Proceedings of the 2005 SIAM International Conference on Data Mining (pp. 471-475). Society for Industrial and Applied Mathematics.
[20] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8).
[21] Salakhutdinov, R., Mnih, A., & Hinton, G. (2007, June). Restricted Boltzmann machines for collaborative filtering. In Proceedings of the 24th international conference on Machine learning (pp. 791-798). ACM.
[22] Amatriain, X., & Basilico, J. (2015). Recommender systems in industry: A netflix case study. In Recommender Systems Handbook (pp. 385-419). Springer US.
[23] Schweder, S. G. (2008). Recommender System and the Netflix Prize (Master's thesis).
[24] W. Isaacson (2011). Steve Jobs. Simon and Schuster.

.