Mobile Enterprise Application Development in Practice: an Analysis of Real-World User Stories

Matthias Jurisch, Stephan Böhm, Toby James-Schulz
Faculty of Design – Computer Science – Media
RheinMain University of Applied Sciences
Wiesbaden, Germany
Email: {matthias.jurisch, stephan.boehm}@hs-rm.de,
t.jamesschulz@outlook.com

***Abstract*—In development of mobile enterprise applications, saving time is an important factor. Short time to market and always changing technical environments require the ability to adapt to changes. However, these demands are in conflict with writing modular, well-thought-out software that allows easy reuse from results of past projects where requirements are often documented in user stories. These short descriptions of ways for a user to interact with the system are often the center of application development. In this paper, we present an analysis of user stories in practice and a categorization of terms used in these stories. We also evaluate how these categorizations relate to algorithmic similarities from information retrieval.**

Keywords–User Story; Mobile Enterprise Applications; Clustering

I. INTRODUCTION

In recent years, user centered design approaches have proven to be beneficial for software and mobile application(app) development. User centered design consists of a variety of tools to assess and evaluate user requirements during the development process, with the goal of delivering software up to par with the users usually high usability requirements. The assessment of user requirements at the beginning of the development process saves time and resources, as focusing on implementing features the user actually needs, leads to less changes having to be made in later stages of development. A way of gathering information about the requirements is through user stories [1], in which the requirements are described from a user's point of view. These stories then serve as a foundation for the further development of source code, screen designs and other software artifacts. Involving the user in the development process is not the only way enterprises can save resources during development, especially larger enterprises can save a lot of time through properly reusing existing software artifacts. Alas there should be a focus on standardized documentation to support the reuse of these software artifacts. This is a challenge large enterprises face, especially with multiple development projects running at the same time. Due to the fast pace of development, the developers often do not have time left for meaningful post-processing of software artifacts, leading to a lack of documentation and documentation standards.

Setting standards and building a library of well documented software artifacts in hindsight, forces a company to allocate resources for theoretically already finished development projects. This would lead to manpower being occupied with working off backlogs, instead of working on the latest projects. Saving time and resources through reuse is a desirable goal for organizations in the Mobile Enterprise Application (MEA) [2] market, but the quick time to market and the fast paced nature of the market environment leave no spare time to deal with backlogs, as new issues and tasks arise on a daily basis. This calls for automated methods to relate stories to each other, which can be built on well-known information retrieval methods. Automatically connecting user stories and recommending them to a developer working on a story would have several benefits: developers might not be aware that similar stories might exist and therefore will not search for similar stories on their own. Also, developers do not need to come up with search terms if they want to find similar stories. When looking at similar stories, developers can reuse existing source code or other artifacts for implementing the story they are currently working on. Computing the similarity of stories would also allow directly recommending other artifacts such as source code or screen designs. However, existing methods for relating user stories to each other often fall short on identifying synonyms and accurately representing domain-specific vocabulary.

To overcome this issue, in this work, we manually analyze a set of real-world user stories regarding the vocabulary used in the most relevant parts of the user story. The results of this analysis are then used to automatically categorize a larger set of user stories from real-world mobile enterprise application development projects. We also compare this automatic categorization to a standard similarity measure from the area of information retrieval.

The remainder of this work is structured as follows: Section II introduces the overall setting of mobile enterprise application development and gives some background on user stories. Section III identifies related work and a research gap that we address in this paper. Our approach is presented in Section IV. In Section V, we introduce our dataset. Term clustering results are shown in Section VI. Results of using these term clusters to categorize stories are discussed in Section VII. The relationship between these clusters and similarity measures from information retrieval is analyzed in Section VIII. Section IX discusses practical implications of this work. A conclusion is given in Section X.

II. BACKGROUND

Mobile Enterprise Applications (MEA) is not a term with an exact definition [2]. In this work, we use this term to

 ISBN: 978-1-61208-754-2

```
As a <user>
I want <feature>
So that <reason>
```

Figure 1. User Story Template

describe applications that are created or used in the context of the daily work of enterprises. These mobile enterprise applications are, just like regular mobile applications, often developed based on user stories. In modern software development, user stories are a common tool to manage and document user requirements. A user story should describe, what kind of interactions a user wants a software system to support and how this is beneficial for the user. The most common template for this is the role-feature-reason or Connextra-format [1]. This template is shown in Figure 1.

The *user* aspect of this template can relate to several aspects. Organizational roles as well as platforms (e.g., "Tablet User") can be used. The *feature* aspect represents the kind of interaction the user wishes a system to support and the *reason* represents the reason, why a user needs this kind of interaction. A typical example for a user story is: *As a user I want to mark and select favorites in order to receive information about my daily bus and train connections as fast as possible*. User stories are often accompanied by *acceptance criteria*, that define required properties of the implementation of a user story. Furthermore, there are several guidelines for creating user stories, one of these are the INVEST criteria [3]. These criteria state, that a user story should be *independent* from other user stories, *negotiable*, *valuable* with a benefit for the user that is clearly identifiable, *estimable* regarding its cost, *small*, and *testable* or verifiable.

In practice, the quality and granularity of user stories may vary. In our experience, the *reason* aspect of user stories is often left out. Besides, user stories are not always formulated in a way that they are easily understandable for an outsider. When trying to use user stories to improve software reuse, this is an important challenge.

III. Related Work

Using short descriptions, such as user stories to support software reuse is not an entirely new idea. Earlier works in this area are based on bug reports. Hipikat [4] proposes using bug reports, which are also short textual descriptions and contain some amount of information about a requirement or software change request, to build a system for recommending software artifacts that can be reused. This system is based on textual similarity of issue descriptions using information retrieval techniques. A related area is the area of issue triage, where software systems recommend developers for a given issue based on the history of a project. In this area, many approaches use methods from the area of information retrieval [5][6]. More recent approaches in this area use deep learning methods [7].

While these approaches have been applied to bug reports, only few approaches have applied these ideas to user stories: [8] proposes a recommendation system based on user stories and evaluates this system on a project history of a single project. Hence, no inter-project knowledge exchange is examined. In our previous work [9], we evaluated how well information-retrieval-based approaches can distinguish between two types of user stories and which aspects of the user story are important to it and collected first evidence on how these approaches perform on real-world data [10]. However, some important properties of user stories in the context of mobile development are unclear: First of all, quality and adherence to the structure of user stories used in practice in this area have not been assessed in the literature. Hence, it is not clear to what degree these user stories can be used for building a recommendation systems that suits practical needs and what kind of semantic similarities can be discovered in this kind of data. Also, using information-retrieval-based similarity measures often have the drawback that synonyms or semantically similar terms can not be identified. While the impacts of this could be offset by using term representations that encode semantic of terms like word embeddings, it is very common in this context to use proprietary or enterprise-specific terms that are not easily represented using these methods.

In this work, we tackle these issues by analysing a set of user stories from real-world mobile enterprise application development projects. Results of these analysis are then used as an input for clustering stories into different categories based on the terms used in specific parts of stories.

IV. Approach

In this work, we want to answer the following research questions:

1) How are user stories used in the mobile enterprise application development context? What parts of user stories are more/less important to developers?
2) Can terms in specific parts of the user story (e.g., in the user or feature part) be separated into clusters based on their semantics? If this is the case, what are the properties of these clusters?
3) How are these clusters related to similarities based on information retrieval techniques?

To address these questions, we first analyze a set of real-world user stories by hand. This manual annotation process is depicted in Figure 2. First, a data cleaning step is required (1). In this step we only select issues that contain text that can be identified as a user story. In the next step (2) we split these issues into acceptance criteria, story title and actual content. In the actual story content is split up into the user, feature and rationale content (3). The users are then clustered into different user types (4).

For the feature part, some more steps for a meaningful clustering are required. When looking at our data we found that many feature descriptions can be summarized by a verb and an object (e.g., "see search results", "enter address" or "edit favorites"). Hence, in one step we select these key verbs and objects (5). Both verbs (6) and objects(7) are then clustered in the last step.

These clusters of elements of user stories can then be used to separate the user stories themselves into three types of clusters (one for each user, one for each key feature verb and one for each key feature object). With the set of terms for each cluster, it is possible to search for these terms in the user stories and assign them to the respective cluster if they contain a term from it.

 ISBN: 978-1-61208-754-2

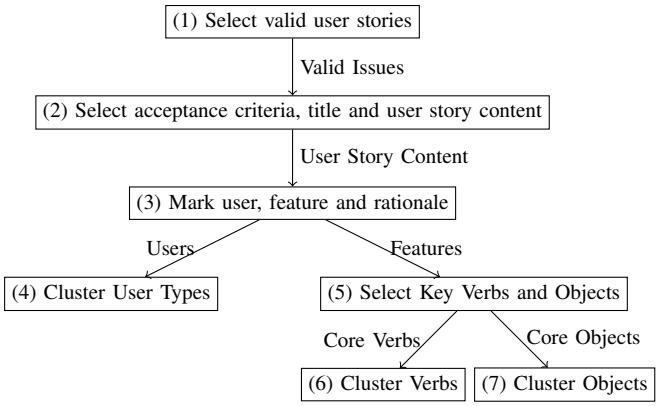

Figure 2. Manual clustering process

These clusters can then be examined regarding their distribution – if all user stories are basically in the same cluster, the clustering method is not very meaningful, while if the distribution is more even among clusters, these clusters might contain some more information. This evaluation is supposed to provide some insights on the second research question.

To answer our third research question, we will compare the similarities of stories inside a cluster with the similarities between stories of the overall dataset. We will use the similarity measures used in [10] to compute the similarities. Namely, we will use TF-IDF-based similarities. TF-IDF is a method for document representation based on term occurrence in documents that is very common in information retrieval [11]. Each document d (i.e., a user story) is represented by a vector $\boldsymbol{W}_d$, that contains an entry for each term used in the dataset. Each vector component $\boldsymbol{W}_{d,t}$ represents the importance of a term t for the document d.

This importance is computed by multiplying $tf_{d,t}$,the frequency of term t in document d, by a representation of how common the term t is in all documents. For measuring the commonality of the t ,the inverse document frequency $\log \frac{N}{df_t}$ is used. N represents the number of all documents and df_t is the number of occurrences for t term in all documents. This yields the following formula for a document's vector representation:

$$\boldsymbol{W}_{d,t} = tf_{d,t} * \log \frac{N}{df_t}$$

To compute the similarity of two documents d_1 and d_2, the cosine of the angle between their vector representation is used.

V. Dataset

The dataset of our evaluations consists of 1408 issues from a Jira system that are not necessarily labeled as *User Stories*. The Jira system is used for organizing app development in a department that mainly focuses on the development of mobile enterprise applications. The Jira System is used by around 100 Users. The user stories stem from more than 20 development projects, where a project is usually implemented by a small team using an agile development approach. An average project has around 50-100 user stories associated to it. While all projects stem from the same company, projects vary in size and type. Also, several different teams have worked on the projects examined in this paper.

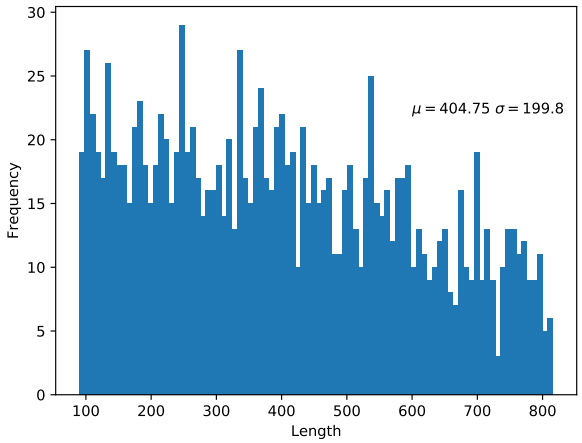

Figure 3. Histogram of User Story length

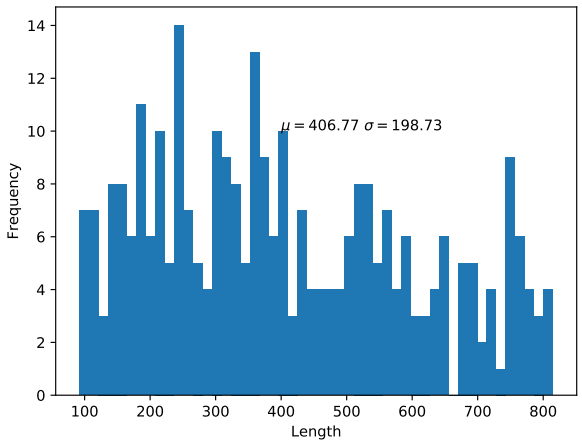

Figure 4. Histogram of User Story length of subset

As in [10], we only considered user stories that contain at least 80 characters of text, since the template alone already contains around 40 characters. The average length of these stories is 404.75 characters with a standard deviation of 199.8. A histogram of story length is displayed in Figure 3. This highlights an important aspect of user stories in practice: there is a large variety in the content and structure of user stories.

Since manually labeling around 1500 stories would be very time consuming, we created a sub-sample of 300 randomly selected stories for the labeling process. Regarding length statistics, this sample is very similar to the full dataset. Mean length is 406.77 characters and standard deviation 198.7. A histogram of these lengths is shown in Figure 4. The average length of text that actually describes the user story (without acceptance criteria, title, etc.) is 99.30 characters. The mean length of the user is 12.81 characters, the mean feature Length is 60.73 characters and the mean rationale length is 47.47

 ISBN: 978-1-61208-754-2

characters. The overall average text length of issues that contain stories is 458.612 characters. All stories consist of a user and a feature, whereas only 58.82% contain a rationale.

VI. Term Clustering

For finding clusters in this dataset, we followed the process described in Section IV. After the first data cleaning step, where we only selected issues that actually contain a user story, 132 issues remained. In these stories, we searched for three types of clusters: (1) Clusters of *User Types*, (2) clusters of *core feature verbs* and (3) clusters of *core feature objects*.

For *users*, we found six clusters. The first of these group consists of several synonyms and translations for the word "user". Another group of users are user descriptions with a platform specification (e.g., "Tablet User"). A third and largest group contains user descriptions that are in some way related to a role in an organization (e.g., "developer"). Another group also relates to a role, while these users are associated with some kind of privileged roles (e.g., "admin, supervisor"). The fifth group is made up of external users such as customers. The sixth group includes users where a broader term, such as the whole department was used, e.g., quality control.

For *core feature verbs*, we found eight clusters. The first three of these clusters are comprised of terms for creating, updating and deleting data. The fourth cluster contains terms related to viewing and working with collections of data, such as sorting or filtering. The fifth cluster of terms is comprised of data management features related to exporting, sharing and importing data. The sixth cluster is related to system management features such as user management and notifications. Another cluster is dedicated specifically to search functionalities. The eighth group of terms is comprised of vocabulary for interacting with conversational interfaces, such as greeting and talking.

For *core feature objects*, we found seven clusters. The first cluster contains widget names. The second cluster groups terms for several types of data such as records or documents. Another cluster contains technical terms such as backend or platform. The fourth cluster represents feedback options for users of the applications. A fifth cluster is related to error handling. The sixth cluster contains terms for an app overview. The seventh cluster groups terms for notifications.

VII. Story Clustering

As described in Section IV, we used the clustered terms for sorting user stories into clusters, based on the existence of the clustered terms in the stories. For *user types*, this lead to cluster sizes as depicted in Table I. Clusters are listed in the same order as they are introduced in Section VI. Note that while the clusters for terms can not overlap, the clusters of user stories based on these terms can, since stories may contain terms from several clusters. The first conclusion from these clusters is, that the term cluster for *Department* seems to not be very important for most user stories, since we can find only 4 stories that contain terms from this cluster. As expected, the cluster containing synonyms for a *Generic User* leads to many user stories. More than a third of stories are part of this cluster.

Feature clusters based on verbs are shown in Table II. Clusters based on core feature verbs seem to be more balanced than clusters based on user terms. Cluster sizes range from 75-313 stories. Hence, these clusters are more likely able to divide the dataset into more meaningfully separated groups than clusters based on user terms.

TABLE I. CLUSTERS BASED ON USER TERM CLUSTERS

Cluster	*Number of Stories*
Generic User	570
Platform	128
Organizational Role	159
Supervisor Role	130
External Users	146
Department	4

TABLE II. CLUSTERS BASED ON CORE FEATURE VERBS

Cluster	*Number of Stories*
Create	202
Edit	160
Delete	75
Collections of Data	313
Importing/Exporting	250
System management	233
Searching	142
Conversational	296

TABLE III. CLUSTERS BASED ON CORE FEATURE OBJECTS

Cluster	*Number of Stories*
Widgets	363
Documents and Reporting	643
Technical Terms	679
Feedback	163
Error Handling	16
Views and Presentation	87
Notifications	66

Story clusters based on core feature objects are shown in Table III. There are two large clusters, namely *Record Types* and *Technical Terms* that contain many user stories and are hence probably to general to be used for separating the data. The cluster for *Error Handling* seems very small, while other clusters are in a similar range to clusters based on feature verbs.

VIII. In-Cluster Similarities

To evaluate how these manually determined clusters relate to methods from information retrieval, we computed the similarity values of the 5 most similar stories that are in the same cluster and the similarities of all stories to the top 5 most similar stories regardless of cluster. We chose the top five most similar stories, since these are the stories that would be on top of a recommendation list. A histogram of similarities is shown in Figure 5. The mean for in-cluster similarities is 0.2859 and the standard deviation is 0.2056. The mean for similarities of all stories is 0.3032 and the standard deviation is 0.2008.

Contrary to our intuition, the distributions of similarities are fairly similar, with the mean for all similarities marginally higher than for similarities in the same cluster. This means that the most similar stories are very often, but not always in the same cluster. Hence, these clusters carry similar information to what is encoded with information retrieval techniques.

 ISBN: 978-1-61208-754-2

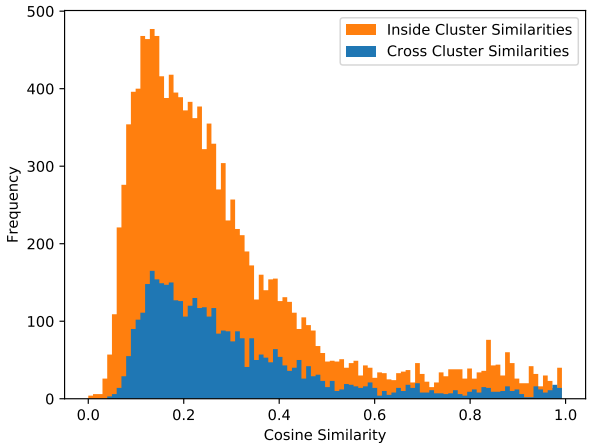

Figure 5. Histogram of top 5 User Story Similarity Inside Clusters compared to Story Similarity of all Stories

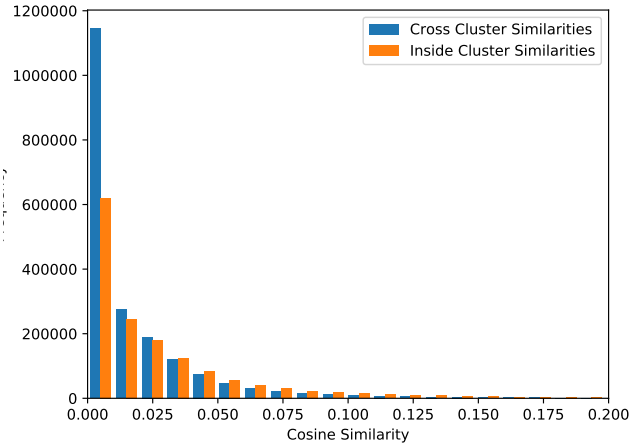

Figure 6. Histogram of User Story Similarity Inside Clusters compared to Story Similarity of all Stories

To further investigate this issue, we also computed the similarities of all stories in the same cluster and the similarities of all issues regardless of cluster. A histogram for both scores is given in Figure 6. The histogram shows only similarities between 0 and 0.2, since higher scores are relatively rare in comparison to this interval. Mean similarities inside clusters is 0.0302 with std. deviation of 0.0523. Mean similarity of all stories is 0.0172 wit std. deviation of 0.0342. This shows that clusters especially filter stories that are dissimilar according to TF-IDF.

IX. Practical Implications and Discussion

Regarding our research questions formulated in Section IV, we can draw the following implications. User Stories are an important part of daily work in mobile enterprise application development - we found that a significant amount of stories exist in a real world dataset. We also found that the rationale part of stories is frequently left out, which is not the case for other parts of the template. The rationale for leaving out the *reason* can be based on two factors: either the *reason* is woven into the *feature* part of the user story, or the author of the story sees the *reason* based in common sense. An example for this might be ”as a user I want a cancel button” or ”as a user I want to receive error notifications”. This leads to the longest and most detailed part of a story being the feature part. Another finding is that a generic user description (e.g., ”User of the application”) is the most common form of user type. User stories have the benefit of delivering a lot of information with little text. A practical implication arising from this is that further education on proper use of user stories is necessary to optimize information retrievability not only for humans but also for algorithms. The same result could probably be achieved by streamlining the templates of the Jira System, to make the resulting user stories more consistent in their form. Another implication is that it could be beneficial to create a repository with user stories describing basic features that every project needs. The repository should give the developer an easy overview of which features are missing e.g., can the user cancel input without losing data?, does the user receive easy to understand error messages?, etc. A repository like this could even be integrated into a templating system to automatically generate stories when a project is created.

Clustering terms into different types of categories for users, feature verbs and feature objects also led to some insights. The clusters for users show, that a large portion of user roles in Mobile Enterprise Application Development are related to organizational aspects. Another aspect are platforms, on which the application is run, but the most common roles are organizational or generic. From building clusters for feature terms, we found that classical paradigms such as the CRUD-pattern [12] can be used to categorize terms used in user stories. Also, many stories consider some type of handling lists or collections of data and system management as well as transferring data. A different area is related to conversational interfaces. Feature objects come from several different categories: They can be related to display items such as widgets or can relate to technical terms for application components.

When it comes to improving recommendations through these clusters, we found that clusters based on our manual categorization and similarities computed using information retrieval methods overlap in many cases – TF-IDF-based similarities in the same cluster are nearly identical to similarities of stories in general when only considering the top 5 most similar stories. This indicates that using these clusters is not likely to be a successful way of improving similarity measures.

X. Conclusion and Outlook

In this paper, we presented an analysis of user stories used in a real-word mobile enterprise application development context. Our main finding is, that user stories in this domain can be categorized based on specific parts of the common template. These categorizations contain similar information to what can be achieved through automatic information retrieval methods. We also found, that the rationale part of user stories is frequently (in roughly 40% of stories in our dataset) left out while other aspects like the feature description seem to carry more meaning for story authors.

As future work several directions are possible. A comparative evaluation of stories from consumer application develop-

 ISBN: 978-1-61208-754-2

ment and mobile enterprise application development could help to highlight differences and challenges that are important in both sectors. Another possible aspect of future work might be creating a set of synonyms or even an ontology of enterprise-specific terms to improve similarity measures.

ACKNOWLEDGMENT

This work was funded by the German Federal Ministry of Education and Research, grant no. 03FH032PX5; the PROFRAME project at RheinMain University of Applied Sciences. All responsibility for the content of this paper lies with the authors.

REFERENCES

[1] M. Cohn, User Stories Applied: For Agile Software Development. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2004.

[2] A. Giessmann, K. Stanoevska-Slabeva, and B. de Visser, "Mobile enterprise applications–current state and future directions," in System Science (HICSS), 2012 45th Hawaii International Conference on, Jan 2012, pp. 1363–1372.

[3] B. Wake, "INVEST in good stories, and SMART tasks," blog post, [retrieved: 2019.09.10], 2003. [Online]. Available: http://xp123.com/articles/invest-in-good-stories-and-smart-tasks

[4] D. Cubranic, G. C. Murphy, J. Singer, and K. S. Booth, "Hipikat: A project memory for software development," IEEE Trans. Softw. Eng., vol. 31, no. 6, Jun. 2005, pp. 446–465.

[5] J. Anvik and G. C. Murphy, "Reducing the Effort of Bug Report Triage: Recommenders for Development-Oriented Decisions," ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 20, no. 3, 2011, pp. 10:1–10:35.

[6] P. Runeson, M. Alexandersson, and O. Nyholm, "Detection of duplicate defect reports using natural language processing," Proceedings - International Conference on Software Engineering, 2007, pp. 499–508.

[7] S. Mani, A. Sankaran, and R. Aralikatte, "Deeptriage: Exploring the effectiveness of deep learning for bug triaging," arXiv preprint arXiv:1801.01275, 2018.

[8] H. Pirzadeh, A. D. S. Oliveira, and S. Shanian, "ReUse : A Recommendation System for Implementing User Stories," in International Conference on Software Engineering Advances, 2016, pp. 149–153.

[9] M. Jurisch, M. Lusky, B. Igler, and S. Böhm, "Evaluating a recommendation system for user stories in mobile enterprise application development," International Journal On Advances in Intelligent Systems, vol. 10, no. 1 and 2, 2017, pp. 40–47.

[10] M. Lusky, M. Jurisch, S. Böhm, and K. Kahlcke, "Evaluating a User Story Based Recommendation System for Supporting Development Processes in Large Enterprises," in CENTRIC 2018, The Eleventh International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services, 2018, pp. 14–18.

[11] D. Jurafsky and J. H. Martin, Speech and Language Processing (2Nd Edition). Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2009.

[12] J. Martin, Managing the Data Base Environment, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1983.

 ISBN: 978-1-61208-754-2