Please quote as: Schmidt, S. L.; Li, M. M.; Weigel, S.; Peters, C. (2021): Knowledge is Power: Provide your IT-Support with Domain-Specific High-Quality Solution Material. International Conference on Design Science Research in Information Systems and Technology (DESRIST). Springer.

Knowledge is Power: Provide your IT-Support with Domain-Specific High-Quality Solution Material.

Simon L. Schmidt[1] [0000-0002-7529-1864], Mahei Manhei Li[1] [0000-0002-7171-9130], Sascha Weigel[1][0000-0002-8825-9409] and Christoph Peters[2,1] [0000-0001-8140-1516]

[1] University of Kassel, Kassel, Germany
{simon.schmidt, mahei.li, weigel, christoph.peters}@uni-kassel.de
[2] University of St.Gallen, St.Gallen, Switzerland
christoph.peters@unisg.ch

Abstract. As more and more business processes are based on IT services the high availability of these processes is dependent on the IT-Support. Thus, making the IT-Support a critical success factor of companies. This paper presents how this department can be supported by providing the staff with domain-specific and high-quality solution material to help employees faster when errors occur. The solution material is based on previously solved tickets because these contain precise domain-specific solutions narrowed down to e.g., specific versions and configurations of hard-/software used in the company. To retrieve the solution material ontologies are used that contain the domain-specific vocabulary needed. Because not all previously solved tickets contain high-quality solution material that helps the staff to fix issues the de-signed IT-Support system separates low- from high-quality solution material. This paper presents (a) theory- and practical-motivated design requirements that describe the need for automatically retrieved solution material, (b) develops two major design principles to retrieve domain-specific and high-quality solution material, and (c) evaluates the instantiations of them as a prototype with organic real-world data. The results show that previously solved tickets of a company can be pre-processed and retrieved to IT-Support staff based on their current queries.

Keywords: DSR, IT-Support, high-quality retrieval, AI, ontology

1 Introduction

More and more information technologies (IT) are getting deployed in organizations and so increase the complexity of infrastructures. While the day-to-day operations are dependent on the continuous availability of these complex IT services the IT-Support is under high pressure when fixing all issues as fast as possible. Whenever IT services stop operating or fail the user can often not continue to work. In other words: The company loses money. To fix an issue, general solution materials are often not sufficient as they do not apply to company-specific errors. Instead, domain-specific solution materials are needed. We define domain-specific solution materials as

descriptions of solutions that take specific software versions, special configurations for companies, company-specific hardware equipment, dependencies between soft-/ hardware components, and country-specific properties/laws into account. To decrease downtime and to provide the IT-Support staff, also called agents, with domain-specific high-quality solution material an IT-Support Support System (SUSY) is built in this paper and instantiated as an artifact. A typical workflow after an error occurred to get the IT services running again is as follows. First, the user opens a support request, also called a ticket, and describes the (un)precise error that occurred. The agent then seeks all information needed to identify the problem, categorizes the ticket, and may ask for additional relevant data and communicates with the user. Based on all information the agent tries to solve the problem either by their domain-expert-knowledge or by consulting documentation of the company. Latter often takes a lot of time due to manually searching information in various databases. During the process of solution-finding, all relevant data are being written down in the ticket, making tickets potentially high-quality solution materials that contain the domain knowledge of the organization with step-by-step solutions. This research paper will focus on previously solved tickets to retrieve solution material because (a) they contain precise domain knowledge and (b) are a good complement next to often outdated and resource-intensive knowledge bases. To provide agents with solution material the information retrieval (IR) literature can be used. Nevertheless, while general-purpose solutions also retrieve general solution materials that are not applicable for agents, this paper builds a domain-specific solution material retrieval system. Further, the retrieval system is combined with an algorithm that ensures that agents are only provided with high-quality solution material because thousands of tickets are written every day and most of them do not contain useful information, e.g., “solved the issue”. Therefore, solution materials with useful information need to be separated from the ones without. Hence, we define high-quality solution material in this paper as the solution material which was separated from low-quality solution material that does not contain useful information to help solve issues.

Based on relevant literature requirements for the agents during the solution-finding process are identified and design principles (DPs) for the development of SUSY are derived. SUSY will help agents by providing them with domain-specific and high-quality solution material. Finally, this will support agents to find solution material fast and to have more time for documentation of new issues and communication to users. The goal of this paper is to develop DPs that will show researchers and practitioners how to build a SUSY e.g., for rebuilding and improving IT-Support systems. Hence, we formulated the following research question: *How can we use previously solved tickets to provide agents with domain-specific and high-quality solution material?* To achieve the goal and generate the output Figure 1 shows the DSR process accordingly to Peffers et al. [23] to design and evaluate the DPs following the problem-centered approach. First, an overview of related work will be given that – next to the introduction – motivates the study and shows the objective of the solution. Second, SUSY will be designed and developed based on design requirements (DRs) and DPs which will be instantiated as design features (DFs). Third, the DRs and DPs will be demonstrated as a prototype to multiple focus groups. Last, the DPs and the DFs will be evaluated before a discussion and the conclusion will close the paper.

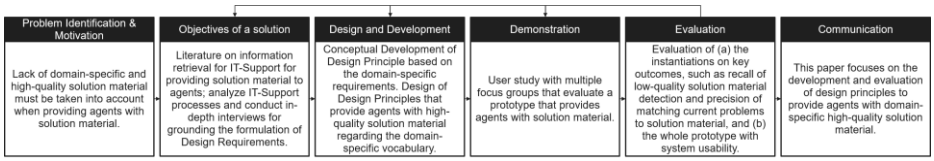


Fig. 1. DSR approach according to Peffers et al. [23]

2 Related Work and Theoretical Background

IT-Support departments are a critical success factor for continuous delivery of IT services for companies and the costs of missing availability can be enormous [12]. There are different data-driven approaches for IT-Support that draw from an interdisciplinary field, such as software engineering, data management, and statistics [12].

By analyzing future events, the failure of systems can be predicted through online failure prediction methods like failure tracking, symptom monitoring, undetected error auditing, and detected error reporting [32]. Many companies route tickets to the corresponding team based on a classification. This classification can be automatized, e.g. by classifying labels in issue tracking systems based on the textual description of an issue [1]. The time needed to fix an issue is an important key performance indicator. Following, predicting the time needed to resolve an incident [15] can help to prioritize work, to allocate resources, to calculate a budget, and to communicate with customers about the time when an issue is fixed [37].

In this paper, we will focus on providing agents with solution material with attention to previously solved tickets. Only a few studies [38, 39] focus on this topic and use ticket summaries to recommend a solution. First, the authors analyze the quality of the possible resolutions to rank them higher/lower according to the quality of the ticket. Second, to recommend a solution, they use a deep neural network ranking model that uses similarity of the summary of the ticket and all possible previously solved tickets together with the measured textual quality from step one. While these studies [38, 39] work with automatically generated incident logs, we focus on user-generated unstructured textual data in the form of tickets making it potentially more complex to compare issues as humans describe the same scenarios in different ways than machines.

The used real-world datasets are company IT-Support tickets and represent organic data [36], which can be leveraged for higher design knowledge relevance. State-of-the-art text classification can be done with Bidirectional Encoder Representations from Transformers (BERT) [6]. However, to analyze our organic data we turn to IR Literature for suitable techniques on how to retrieve solution materials [2]. IR encompasses techniques to find information based on unstructured data. They range from indexing, filtering, to searching techniques [2, 30]. Filtering techniques are based on natural language processing (NLP) and e.g., remove punctuations, or lemmatize words. Indexing techniques map documents and their content (words) to specific numbers that represent the document (e.g., a keyword score that denotes the importance of a word for a document). These indices summarize the information in a document and allow searching techniques to retrieve and compare information more rapidly e.g., with

linear search algorithms, or brute force search [30]. These IR techniques can be applied as general-purpose solutions to IT-Support to extract keywords from tickets to find relevant solution materials [18]. These keywords can be maintained (e.g., protégé editor [20]) and extended by word2vec approaches [27] or by manually adding keywords.

Because many previously solved tickets do not contain useful information, we turn to NLP techniques and machine learning (ML) algorithms to detect the quality of solution materials. As shown in [17, 21, 29] the quality of a text can be assessed by extracting text features with NLP (e.g., number of words, timeliness, relevancy, length of the text, the ratio of number of characters to the number of sentences, the ratio of stop words to the number of words, uniqueness measured by term frequency-inverse document frequency (tf-idf), readability measured by Coleman-Liau score), labeling the text quality (e.g., experts, crowdsource), and training an ML algorithm (e.g.: support vector machine, k-nearest neighbor, logistic regression) with the labels and text features.

Overall, to the best of our knowledge, there do not exist IT-Support systems that (a) use domain-specific vocabulary when retrieving solution material and (b) differentiate between high-/low-quality solution material. However, some similarities have been identified with peer support systems, which focus more on the social interaction among its systems, and especially among its peers, and less on the quality issues of each element [16]. To contribute to the existing literature on IT-Support we guide the development of our holistic system that takes domain-specific and high-quality solution material into account and present SUSY.

3 Designing SUSY

3.1 Developing Design Requirements

To ensure that the real-world problems of all IT-Support stakeholders will be considered we (a) conducted literature research about IT-Support and its related problems and challenges, especially in day-to-day routines, and (b) conducted 21 in-depth interviews with stakeholders of the IT-Support e.g., agents, knowledge managers, global managers, etc. To generalize our outcome, we conducted the in-depth interviews with stakeholders from three different IT-Support departments around the globe. Through the formulated DRs the first part of the objectives of the DSR approach according to Peffers et al. [23] is addressed.

While agents are under high pressure and the complexity of infrastructures rises, they often struggle to find appropriate solution material. Automatically retrieved knowledge (**DR1**) would help them find solutions faster because in most cases they still search manually in different databases. This would especially help low-skilled agents. These are often found in external IT-Support departments that have been outsourced [4, 13, 22]. The manual search often takes a long time with up to 30 minutes until information about a specific error has been retrieved, opening a huge potential for automatically supplying agents with solution material. Nevertheless, different companies offer different IT services that vary in complexity and configuration. The retrieved solution material needs to take this domain-specific vocabulary of the

company into account when providing agents with solution material of the exact error (**DR2**). This domain-specific solution material needs to be found especially when the IT-Support is outsourced [4, 10].

Not all tickets are useful as solution material to solve tickets. A ticket might address the same problem but might not be suitable as a guide for solution-seeking agents resulting in wasted time reading the low-quality solution materials that may only contain "solved the problem on phone. Ticket closed". Hence, these low-quality solution materials need to be found and separated (**DR3**) from the high-quality solution materials. As the basis of the problem space, all derived DRs were evaluated in a formative artificial way to match the real-world problems of the IT-Support stakeholders. Accordingly, we asked the stakeholders in the in-depth interviews as well as in subsequent meetings and received great approval after iteratively adapting the DRs several times according to the high-quality feedback gained. After receiving no additional new feedback to further develop the DRs the interview series was stopped after 21 in-depth interviews. The DRs can be found in Figure 2 together with the derived DPs and instantiated DFs.

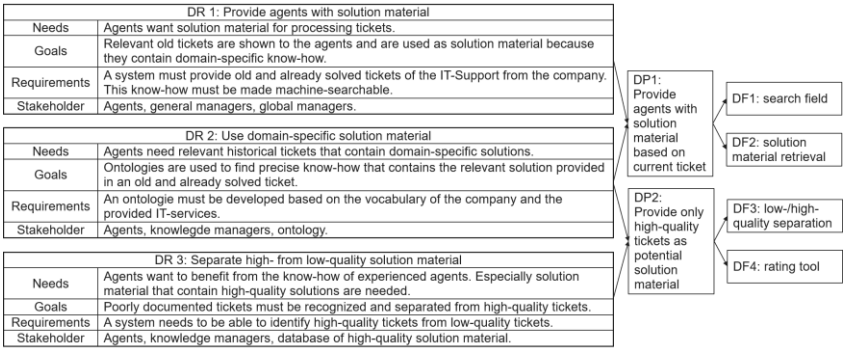


Fig. 2. Derived DRs with the developed DPs and instantiated DFs

3.2 Design Principles for Solution Material Retrieval

Based on the design requirements, we developed two major DPs to (a) provide agents with domain-specific and (b) high-quality solution material based on previously solved tickets. With these DPs we want to address scholars as well as practitioners to build and improve a SUSY for the IT-Support. The DPs follow the anatomy of Gregor et al. [9] to ensure better reusability and describe the aim, users, implementer, context, mechanism, and rationale. Further, a recommendation that summarizes the whole DP, a visual representation to better understand the core of the DP, and an exemplary technology for implementation to increase the reusability, was added. Ultimately, two DPs were formulated.

DP1 (see Table 1) is informed by the IR literature (e.g.: [2, 19, 20, 27, 30]). It describes how to provide agents with solution material automatically based on their current ticket. In this paper, NLP is used in combination with an ontology. Different NLP techniques are used: (1) Tokenization, (2) removal of numbers, (3) removal of punctuation, (4) lower casing, (5) lemmatization, and (6) stop word removal. We

created our ontology using a semi-automated approach. Since the ticket data we use for our project already contains tags that relate tickets to categories (e.g., “password”), we used these tags to initially populate our ontology. Furthermore, we applied keyword extraction methods (tf-idf) and used word embeddings to extend the ontology. To be able to find new terms, we trained a word2vec model on the ticket data. We used this model to find similar terms (or nearest neighbors) to the words already present in the ontology (using cosine similarity) as proposed by [25]. The identified candidate terms were shown to domain experts to evaluate the quality of the results. If deemed relevant the new terms were added to the ontology. These new terms included previously missing concepts and synonyms (e.g., “PW” as an acronym for password) or translations of already known terms. The newly added terms were then appended to the similarity queries on the word2vec model, allowing us to find even more candidate words through this iterative approach. The procedure was discontinued after no new candidate terms were found through the described similarity query. We clustered all ontology terms based on their word vectors derived from word2vec using k-means to segregate the data into meaningful categories. We determined the optimal number of clusters using the within-cluster sums of squares and average silhouette methods. To ensure quality, these clusters were once again evaluated by domain experts.

DP2 (see Table 2) is informed by the literature of text mining and ML (e.g.: [7, 17, 21, 26, 29]) and describes how to separate low-/high-quality solution material to provide agents with high-quality solution material only. In a first step, a set of previously solved tickets needs to be labeled manually according to their solution material quality, e.g., 1 = high-quality, 0 = low-quality. Next, the first feature readability is extracted based on the Coleman-Liau score. NLP then pre-processes tickets (Tokenization, removal of numbers, removal of punctuation, lower casing, and lemmatization) and automatically extracts more text features, that are, number of sentences, number of verbs, number of nouns, number of stop words, and individuality measured by the sum of the tf-idf indices of a ticket. In the next step, these features have to be compared to select the most appropriate ones for the classification task, e.g. with a Chi-square test. With the final text features and the labels, ML algorithms are trained and then detect low-/high-quality solution material in new sets of tickets. In the last step, the low-quality solution materials are deleted from the set of solution materials, resulting in providing agents with high-quality solution material only. Through continuous feedback from agents during the operation of SUSY, the low-/high-quality solution material detection can further learn resulting in potentially better results the more agents use SUSY.

Table 1. DP1: Provide agents with problem-solution material based on current ticket

Aim, users, and implementer
To provide agents (users) with solution material to solve new issues, similar tickets have to be found from a system (aim) developed by a programmer (implementer)
Context, mechanism
A set of previously solved tickets needs to be compared to a new ticket (context) to retrieve the solution material. This is done by comparing the similarity between new and previously solved tickets considering the domain-specific vocabulary. To do this, an ontology can be built e.g., with a semi-automatic approach that creates the ontology based on frequently occurring terms (e.g., tf-idf algorithm). The ontology can be extended by word embeddings

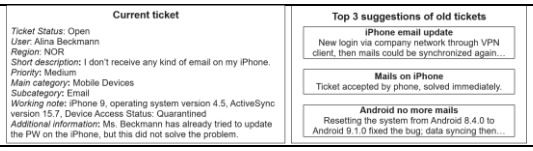
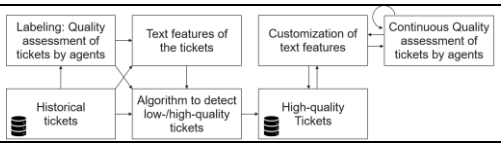
(word2vec), adding similar neighboring terms, or by manually adding words to the ontology (e.g., by agents or knowledge manager). To further improve an ontology its terms can be structured and grouped using clustering algorithms. When comparing tickets, the tickets can be reduced to their most important domain-specific terms. Viewing tickets as a set of those terms, similarity can be measured through metrics like Jaccard similarity.
Rationale
Overall, new and previously solved tickets are compared because already solved similar tickets can provide the needed solution material for the current ticket and manual search for solution materials is time-consuming. Next to the ontology, NLP has to be used for various pre-processing steps [14, 31]. These include tokenization, lemmatization, stop word removal [34], part-of-speech tagging [28], dotation removal, number handling, lowercasing, and stemming [5].
Recommendation
Find similar previously solved tickets to a new ticket, extract the solution material, and provide it to agents.
Visual representation

Technology for implementation
For NLP we recommend using Python with its' powerful libraries e.g., Stanza for 66 languages [26]. Servers from Microsoft Azure can be used to deploy the code online e.g., the free Azure App Service "F1" with 1 GB RAM, shared kernels, 1 GB storage, and Linux as the operating system. If the whole code-base is saved on GitHub one can build a "pipeline" e.g., between GitHub and the Microsoft Azure App Service to automatically deploy code online when pushing it to GitHub. To create and maintain an ontology one can use the Protégé editor [20]. It allows for the easy creation of ontologies using the web ontology language (OWL) [19]. Furthermore, new classes and instances can be imported using spreadsheets. More advanced text mining techniques like the mentioned word2vec can be applied using the Python library Gensim [27], but it should be kept in mind that they need a vast amount of textual data to produce reasonable results. Clustering and feature extraction (e.g. tf-idf) can be done with scikit-learn [7].

Table 2. DP2: Provide only high-quality tickets as potential solution material

Aim, users, and implementer
To provide the domain-specific solution material retrieval system (user, see DP1) with high-quality solution materials (aim) a programmer (implementer) must develop an algorithm that detects the low-quality solution materials and separates them from the high-quality solution materials (aim).
Context, mechanism
A set of previously solved tickets (context) needs to be processed with NLP to create text features that are used within ML algorithms that assess the quality of solution materials (mechanism).
Rationale
The high- and low-quality solution materials need to be separated because only the latter contain well-documented step-by-step solution materials in a way that is understandable for agents. Only the consideration of several text features leads to a selection of the right tickets for the database. According to Otterbacher [21], for measuring the quality of product reviews, the number of sentences and number of words are strongly correlated with the relevance of

the review, which is the most important of five criteria for assessing quality. Similar results of the length of a post about its quality were found by Rhyn & Blohm [29] for crowdsourcing texts; it was their strongest predictor. Liu et al. [17] examined the quality of product reviews based on the characteristics of informativeness, subjectivity, and readability.
Recommendation
Use an algorithm (e.g., logistic regression, k nearest neighbor) for identifying low-quality tickets that considers the most relevant features (e.g., uniqueness, number of verbs, readability, and number of stop words) in terms of ticket quality.
Visual representation

Technology for implementation
For NLP we recommend Python and Stanza [26]. Several different ML algorithms, e.g., from scikit-learn [7], should be tested with different combinations of the extracted text features according to the domain-specific tickets. To measure uniqueness, one can take the sum of the tf-idf indices of a ticket. For readability, the Python package readability 0.3.1 can be used for readability grades such as Coleman-Liau score as well as for further sentence information such as characters per word, word per sentence, etc. To identify the most likely feature candidates that are irrelevant for classification the Chi-squared method from scikit-learn can be used [7]. To label the data, multiple agents should be asked to assess the quality of the tickets to train the algorithms. To measure inter-rater reliability Cohen's kappa can be used.

3.3 SUSY and its Design Features

The DPs were applied to design SUSY and its core DFs. Figure 3 illustrates the built DFs derived from the DPs. SUSY is deployed on two Microsoft Azure Web Apps. While the backend is deployed on a Linux-based Web App and written in Python, the frontend is deployed on a Windows-based Web App and written in PHP/ Node.JS. The frontend communicates with the backend by an application programming interface.

Whenever agents need solution material, they can simply go to the website of the frontend and either paste the query of the user or search for specific terms (**DF1**: search field). The matching system of SUSY then uses its domain-specific knowledge base and understands the company-specific content of the request. To match the request with a potential solution material the ontology uses the vocabulary and tags the request with the relevant terms. Then, the ontology compares the tags of the request with the potential solution materials and displays the top five matches. We call these matches highly relevant because they are semantically similar and contain the domain-specific information an agent needs to resolve an issue (**DF2**: solution material retrieval). A total of five solution materials are presented to the agents and are of high quality only (**DF3**: low-/high-quality separation). This is provided by the data logic of the backend that pseudonymizes all tickets accordingly to the general data protection regulation and analyzes all potential solution material whether they contain high-quality solution material. Only the pseudonymized tickets of high quality are stored as solution material for the company/agents. To see the perceived quality of solution material agents are given the opportunity to rate solution materials (**DF4**: rating tool). This helps twofold:

Firstly, the agents can see which provided solution materials contain perceived high-quality solutions briefly. Secondly, the algorithm that evaluates the quality of the solution material (**DF3**) automatically receives feedback and can learn through the human-in-the-loop design. Thus, the agents receive potentially more high-quality tickets the more they use the system.

Fig. 3. The user interface of SUSY and corresponding design features

4 Evaluation of the DPs and Artifacts

The DPs were evaluated according to the FEDS framework from Venable et al. [35] following the evaluation strategy technical risk and efficacy. The design principles were evaluated (a) via a technical experiment by instantiating the DP1 and DP2 as specific design features (**DF2** and **DF3**, see Figure 4) and evaluating the performance of the algorithms using real-world data as a summative-naturalistic evaluation, and (b) via an initial prototype evaluation that demonstrates the useability of the instantiated DPs in a summative-naturalistic evaluation (all DFs) with future users of SUSY [24, 35]. According to [11] the evaluation is analytical and experimental to highlight the practical relevance, usefulness, and performance of the artifact in a real-world scenario.

DF2 (solution material retrieval), which is mainly based on **DP1**, matches a query to potential solution material. To prove the value of **DF2** a database of 963 unique tickets of high-quality solution material was used. We searched solution material for 40 different random queries that are based on previously solved tickets to make the evaluation as realistic as possible. Because it is from high interest that agents are provided with very precisely matched solution materials we decided for precision as the key metric. **DF2** presents the agents with a maximum of five solution materials. These retrieved solution materials have been evaluated whether they would help to solve the query or not. The ontology was refined by iteratively adding new terms to its vocabulary using the method described in **DP1**. After several iterations, our final artifact reached a precision of 0,87 for matching high-quality solution material to domain-specific queries. In comparison, a tf-idf search algorithm has been evaluated to the same queries and data set and only reached a precision of 0,29 which shows the applicability of the ontology-based retrieval.

DF3 (low-/high-quality separation), which is mainly based on **DP2**, detects low-quality solution material based on extracted text features and deletes the low-quality

solutions. First, a total of 800 tickets were labeled from two different persons with a Cohen's kappa of 0,71, suggesting a substantial agreement between the annotators. Next, the dataset was split into a training (n = 480) and an evaluation (n = 320) dataset, and the algorithms k nearest neighbor, logistic regression, decision tree, support vector machine, and two-class-Bayes were evaluated. Because it is important to detect as many low-quality solution materials as possible, we decided for the recall of low-quality solution materials as the key metric. The final model (logistic regression with the features number of verbs, uniqueness, readability, and number of stop words) reached a recall of 0,91 for detecting the class of low-quality solution material as indicated in Table 3. This table also shows the accuracy, F1, recall, and precision based on the weighted average as well as the precision for the class of low-quality solution materials.

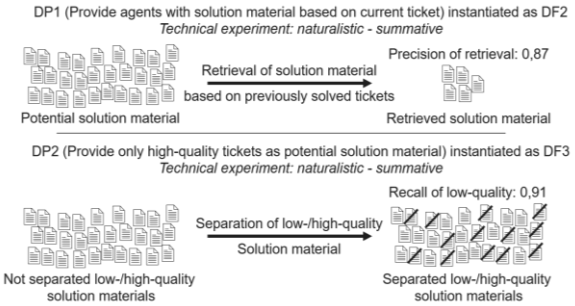


Fig. 4. Evaluation of the instantiations of DP1 and DP2

Table 3. DF3 Evaluation – class of low-quality (lq) and weighted average (wa)

Algorithm	Rec_{lq}	$Prec_{lq}$	Acc	Rec_{wa}	$Prec_{wa}$	$F1_{wa}$	$F1_{wa[Train]}$
k-nn	0,81	0,70	0,68	0,68	0,67	0,67	0,82
Log. Reg.	0,91	0,72	0,74	0,74	0,73	0,73	0,63
Dec. Tree	0,81	0,74	0,71	0,71	0,70	0,70	0,71
SVM	0,80	0,79	0,75	0,73	0,73	0,73	0,70
2Class Bayes	0,88	0,68	0,68	0,68	0,67	0,62	0,59

To further evaluate the DPs through instantiations SUSY was evaluated as a summative-artificial evaluation through five focus groups with stakeholders of the IT-Support of five to six people each. To analyze the strengths and weaknesses of SUSY qualitative techniques were used. Based on these results of the "start, stop, continue" exercise the DPs, as well as the DFs, were further developed. Last, the System Usability Scale [3] to evaluate the overall usability of SUSY was used. The dataset of the participants was limited to the participants of the focus groups because they are the alpha tester. The data provides preliminary evidence to suggest that the alpha version of SUSY already has good usability, with a score of 76,4 (n = 18) [3]. Overall, the DPs have been evaluated as instantiations in summative-naturalistic ways using real-world data. While SUSY is still in the alpha phase the prototype proves preliminary evidence for good useability, and DF2 and DF3 already show sufficient results regarding previous studies: **DF2** reached a precision of 0,87 based on 40 queries. For comparison, in [33] ten queries were evaluated with an ontology-based retrieval system for IT-Support that reached a precision of 0,86 compared to a keyword-based approach that

reached a precision of 0,13. **DF3** reached a recall of 0,91 to detect low-quality solution materials while in a similar study [29] crowd-support texts were analyzed and the detection of low-quality texts reached a recall of 0,75.

5 Discussion, Implications, and Future Work

This paper presents a way to provide the IT-Support with domain-specific (DP1) and high-quality (DP2) solution material based on previously solved tickets and a domain-specific ontology. Based on issues from the IT-Support we generated DRs applicable for agents during the solution-finding process. These DRs describe the call for solution material (DR1), considering the domain-specific vocabulary (DR2), and the detection of low-quality solution material to separate them from the high-quality data (DR3). Based on these DRs DPs were derived and DFs were instantiated as a prototype while all outputs of the DSR paper were constantly evaluated during all stages. There is a known body of knowledge to retrieve solution material for the IT Support e.g., from IR, ML, and data mining communities. By combining these known solutions the identified novel problems were solved making SUSY an exception that shows a new solution to a known problem [8].

The DPs are formulated based on an anatomy that gives deep insides into the theoretical and practical decisions done. Hence, we not only share design knowledge in the form of DPs but also give relevant tips for the implementation and present examples for the selection of technologies. The used data is based on an organic [36] real-world dataset of tickets that companies have made available to us. This means that the data is not designed and was not generated for research purposes. The instantiated DP1 provides preliminary evidence that a domain-specific ontology is well suited to retrieve precise solution material (precision = 0,87). The ontology was created with a semi-automatic approach based on frequently occurring terms and was extended by manually adding important keywords. In future research, we want to extend the ontology with word-embeddings. DP2 provides that text mining techniques in combination with ML algorithms predict low-quality solution material with a high recall (recall = 0,91). In specific, logistic regression with four text features was used. In future studies, we want to add non-text features, e.g., the total time to fix and the number of involved agents to further improve the algorithm.

The derived DPs allow future research in that direction and so are not without limitations. The DPs and DFs were only evaluated with alpha testers and SUSY will soon be rolled out to more testers to gain deeper insights. In upcoming works, we also want to show that the derived DPs increase the productivity of agents and want to derive more DPs that focus on highlighting the unstructured text in different parts (e.g.: solution, problem, noise) to analyze the unstructured text faster.

In conclusion, we focused on (1) deriving theoretically and practically grounded DRs, (2) developing DPs and their artifacts, (3) evaluating each instantiation, and (4) evaluating the summative-naturalistic design of SUSY. Addressing these four steps, a set of DPs to support the development and instantiation of future support systems for the IT-Support were presented. Through the DPs and DFs, we hope to inspire

researchers to rebuild/further develop SUSY to solve the day-to-day problems of the IT-Support when searching for solution material as the IT-Support is one of the most important departments of a company and keeps all IT services running.
This research is funded by the German Federal Ministry of Education and Research (BMBF) and supervised by PTKA (Project HISS - 02K18D060).

References

1. Alonso-Abad JM, López-Nozal C, Maudes-Raedo JM et al. (2019) Label prediction on issue tracking systems using text mining. Progress in Artificial Intelligence 8:325–342
2. Baeza-Yates R, Ribeiro-Neto B (1999) Modern information retrieval. ACM press, New York
3. Brooke J (1996) SUS: A 'Quick and Dirty' Usability Scale. In: Jordan PW, Thomas B, Weerdmeester BA et al. (eds) Usability Evaluation in Industry. Taylor & Francis, London, pp 189–194
4. Chagnon C, Trapp A, Djamasbi S (2017) Creating a Decision Support System for Service Classification and Assignment through Optimization. AMCIS 2017 Proceedings
5. Denny M, Spirling A (2018) Text preprocessing for unsupervised learning: Why it matters, when it misleads, and what to do about it. Political Analysis 26:168–189
6. Devlin J, Chang M-W, Lee K et al. (2018) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805
7. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort et al. (2011) Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12:2825–2830
8. Gregor S, Hevner AR (2013) Positioning and Presenting Design Science Research for Maximum Impact. Mis Quarterly 37:337–355
9. Gregor S, Kruse LC, Seidel S (2020) The Anatomy of a Design Principle. Journal of the Association for Information Systems Forthcoming
10. Grupe FH (1997) Outsourcing The Help Desk Function. Information Systems Management 14:15–22. doi: 10.1080/10580539708907040
11. Hevner AR, March ST, Park J et al. (2004) Design Science in Information Systems Research. Mis Quarterly 28:75–105. doi: 10.2307/25148625
12. Kubiak P, Rass S (2018) An Overview of Data-Driven Techniques for IT-Service-Management. IEEE Access 6:63664–63688. doi: 10.1109/ACCESS.2018.2875975
13. Lacity MC, Khan SA, Willcocks LP (2009) A review of the IT outsourcing literature: Insights for practice. The Journal of Strategic Information Systems 18:130–146
14. Lamkanfi A, Demeyer S, Soetens QD et al. (2011) Comparing Mining Algorithms for Predicting the Severity of a Reported Bug. 15th European Conference on Software Maintenance and Reengineering:249–258
15. Lee Y, Lee S, Lee C-G et al. (2020) Continual Prediction of Bug-Fix Time Using Deep Learning-Based Activity Stream Embedding. IEEE Access 8:10503–10515
16. Li MM, Peters C, Leimeister JM (2017) Designing a Peer-Based Support System to Support Shakedown. International Conference on Information Systems
17. Liu J, Cao Y, Lin C-Y et al. (2007) Low-Quality Product Review Detection in Opinion Summarization. Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning:334–342
18. Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge University Press, Cambridge

19. McGuinness DL, Van Harmelen F (2004) OWL Web Ontology Language Overview. W3C Recommend 10
20. Musen MA (2015) The Protégé Project: A Look Back and a Look Forward. AI Matters 1:4–12. doi: 10.1145/2757001.2757003
21. Otterbacher J (2009) 'Helpfulness' in online communities. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp 955–964
22. Patil S, Patil YS (2014) A Review on Outsourcing with a Special Reference to Telecom Operations. Procedia - Social and Behavioral Sciences 133:400–416
23. Peffers K, Tuunanen T, Rothenberger MA et al. (2007) A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 24:45–77
24. Peffers K, Rothenberger M, Tuunanen T et al. (2012) Design Science Research Evaluation. International Conference on Design Science Research in Information Systems:398–410
25. Pembeci İ (2016) Using Word Embeddings for Ontology Enrichment. International Journal of Intelligent Systems and Applications in Engineering 4:49–56. doi: 10.18201/ijisae.58806
26. Qi P, Zhang Y, Zhang Y et al. (2020) Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. arXiv preprint arXiv:2003.07082
27. Radim R, Sojka P Software Framework for Topic Modelling with Large Corpora. Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks 2010:45–50
28. Rajman M, Besançon R (1998) Text Mining: Natural Language techniques and Text Mining applications. In: Spaccapietra S, Maryanski F (eds) Data Mining and Reverse Engineering. Springer, Boston, MA, pp 50–64
29. Rhyn M, Blohm I (2017) A Machine Learning Approach for Classifying Textual Data in Crowdsourcing. Wirtschaftsinformatik 2017 Proceedings
30. Roshdi A, Roohparvar A (2015) Review: Information retrieval techniques and applications. International Journal of Computer Networks and Communications Security 3:373–377
31. Runeson P, Alexandersson M, Nyholm O (2007) Detection of duplicate defect reports using natural language processing. In: International Conference on Software Engineering, pp 499–510
32. Salfner F, Lenk M, Malek M (2010) A survey of online failure prediction methods. ACM Computer Surveys 42:1–42. doi: 10.1145/1670679.1670680
33. Shanavas N, Asokan S (2015) Ontology-based Document Mining System for IT Support Service. Procedia Computer Science 46:329–336. doi: 10.1016/j.procs.2015.02.028
34. Srividhya V, Anitha R (2010) Evaluating preprocessing techniques in text categorization. International journal of computer science and application 47:49–51
35. Venable J, Pries-Heje J, Baskerville R (2016) FEDS: a Framework for Evaluation in Design Science Research. European Journal of Information Systems 25:77–89. doi: 10.1057/ejis.2014.36
36. Xu H, Zhang N, Le Zhou (2020) Validity Concerns in Research Using Organic Data. Journal of Management 46:1257–1274. doi: 10.1177/0149206319862027
37. Yedida R, Yang X, Menzies T (2021) When SIMPLE is better than complex: A case study on deep learning for predicting Bugzilla issue close time. arXiv preprint arXiv:2101.06319v1
38. Zhou W, Tang L, Zeng C et al. (2016) Resolution Recommendation for Event Tickets in Service Management. IEEE Trans Netw Serv Manage 13:954–967. doi: 10.1109/TNSM.2016.2587807
39. Zhou W, Xue W, Baral R et al. (2017) STAR: A system for ticket analysis and resolution. Proceedings of the 23rd ACM:2181–2190. doi: 10.1145/3097983.3098190