Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-10 (2022) 1840–1845

A Case Study: Digitalization of Business Processes of SMEs with Low-Code Method

First Z. Cai* Second Y. Huang** Third S. Kessler*** Fourth J. Fottner****

* School of Engineering and Design, Technical University of Munich, Munich, Germany, (e-mail: zhen.cai@ tum.de)
** School of Management, Technical University of Munich, Munich, Germany, (e-mail: yun.huang@tum.de)
*** School of Engineering and Design, Technical University of Munich, Munich, Germany, (e-mail: stephan.kessler@tum.de)
**** School of Engineering and Design, Technical University of Munich, Munich, Germany, (e-mail: j.fottner@tum.de)

Abstract: Compared to large companies, small- and medium-sized enterprises (SMEs) in some critical industries, e.g. construction industry, have a lower level of digitization. One main reason is the complexity of digital transformation in the development and implementation phase. This phase involves cost-intensive and time-consuming stages due to legal constraints in Europe. From the perspective of technology, full-fledged IT Systems would not entirely meet the specific requirements of SMEs. This paper aims to provide an approach to support SMEs to digitalize their business processes with low costs and high flexibility by the Low-Code method. This approach is validated with a case study.

Keywords: small- and medium-sized enterprises (SMEs), digitalization, Low-Code method, BPMN, Low cost technologies

1. INTRODUCTION

As in 2011, Germany first presented the concept of Industry 4.0, which addressed the importance of digitalization in the industry branches (Tay et al., 2018). Through the digital transformation, companies could enhance the availability of information and visibility throughout their business processes, thus optimizing the profit and staying competitive (Liker and Choi, 2004; Tippins and Sohi, 2003). While many large companies have embraced the best practices, such as the digital twin in manufacturing, empirical evidence suggests this is not the case for small- and medium-sized enterprises (SMEs) in Europe (Bell, 2006). The SMEs in Europe are left behind in the process of digitalization.

SMEs are very important to the European economy, which are accounted for 99% of all European businesses and 67% of employment (European Commission, 2004-2010). In the meantime, SMEs have low turnover compared to the big players in the market because of their limited capabilities for investment in new technologies (Hampson et al., 2014). Meanwhile, the implementation procedure in Europe includes cost-intensive and time-consuming stages due to legal restrictions. Furthermore, most of the fully developed software would not entirely fit particular requirements of some SMEs.

The idea of Low-Code was first introduced in 2011. Low-Code Method is meant to provide an approach to creating an application through a graphical user interface instead of traditional hand-coded computer programming, which benefits non-professional software developers. Applying the Low-Code method for the digital transformation can reduce the costs of the IT implementation and increase the flexibility of the application (Phalake and Joshi, 2021). The Low-Code method has shown its advantages over traditional coding in terms of effectiveness and flexibility, which has been proven in many studies (Sanchis et al., 2020; Wang et al., 2021; Waszkowski, 2019).

In this paper, we want to propose an approach to digitize SMEs business processes with the aid of the Low-Code method. The paper is organized as follows: in chapter 2, we first analyze the theoretical background of digital transformation, and then we propose an approach of digitalization of processes with the Low-Code method and specified it into six steps. In chapter 3, a use case is introduced, and we apply our process with the Low-Code method on the use case and show the result within each step. In chapter 4, we evaluate the case study results and discuss the strengths and limitations of our approach. Chapter 5 concludes the paper and shows the future work.

2. THEORETICAL BACKGROUND AND METHODOLOGY

The digital transformation can be generally divided into three stages: evaluation and goal definition, strategy definition and implementation (Zaoui and Souissi, 2020), which reflects also the digitalization guidelines in Europe. From a management perspective, the guidelines of

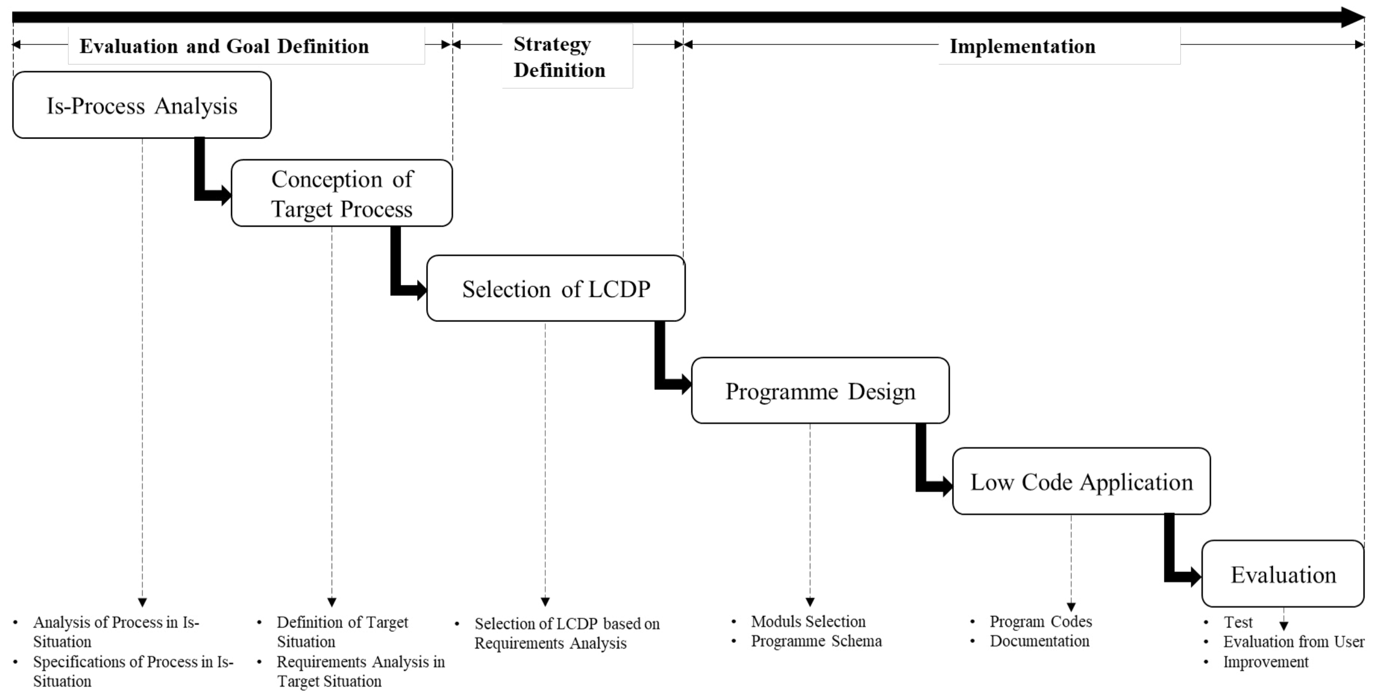

Evaluation and Goal Definition
Strategy Definition
Implementation
Is-Process Analysis
Conception of Target Process
Selection of LCDP
Programme Design
Low Code Application
Evaluation
• Analysis of Process in Is-Situation
• Specifications of Process in Is-Situation
• Definition of Target Situation
• Requirements Analysis in Target Situation
• Selection of LCDP based on Requirements Analysis
• Moduls Selection
• Programme Schema
• Program Codes
• Documentation
• Test
• Evaluation from User
• Improvement

Fig. 1. The Methodology of the Low-Code Process

managing IT system implementation are specified with approaches such as the Internet-based, process-oriented selection of standard software application method, shortened known as ePAVOS (Teich et al., 2008). From the technical perspective, approaches to develop IT system and/or software are also proposed, such as the Waterfall model, the V model are presented (Reinhard Höhn, 2008). The approaches from both perspectives contain classic three stages of digital transformation, only with different focuses and applied methods.

To apply the Low-Code method, we have combined the advantages of the approaches from both the management and technical perspectives. Our approach can be divided into three phases, as shown in Fig.1.

Phase 1 Evaluation and goal definition: In this phase, the current process (Is-Process), which should be digitalized, will be evaluated. Through methods like workshops, expert interviews and literature research, the workflow and its specifications (e.g. system environment, framework) of the Is-Process are defined. The next step is to determine the target process and its functionality, visually presented with a business process model such as BPMN modelling (Recker, 2012).

Phase 2 Strategy definition: In the strategy phase, to apply the Low-Code method, a suitable development platform (LCDP) should be selected based on results from the earlier phase. Different methods could be applied; for example, market research and a SWOT analysis could be performed (McGivern, 2013; Schawel and Billing, 2012).

Phase 3 Implementation: After choosing the LCDP, Low-Code modules are selected based on the functionality from analysis of the target process and a suitable program schema is built. After that, the Low-Code modules are combined based on the program schema with documentation. The last step is to test the functions of the final program with users and improve the program's performance based on the users' feedback.

3. CASE STUDY AND RESULTS

3.1 Phase 1 Evaluation and goal definition

A German SME focused on road construction collaborated in the work and provided the delivery note process as a use case to be digitalized and automated. The company's documentation process of delivery notes was a pain point, as this process was not digitalized and automated until then. The workflow involved many manual steps, leading to low productivity and high error quotes. The IT system's most critical information was also analyzed from expert interviews and workshops with the company. The existing infrastructure is Microsoft Windows, and the company has subscripted Microsoft office software. The requirements for digitization or automation of delivery note processing were summarized in a specification sheet and the target state of the process was derived. The functionality required was summarized in Table 1:

Table 1. Functionality of the required application

Functionality	Rating
Laptop and mobile application	required
Automatically extracting text document formats (.pdf, .jpg)	required
Automatically importing relevant data into the database	required
Summary and sorting of data in the same data group	required
Automatically recognizing handwritten characters	optional

Besides the functionality, the workflow of the delivery note process was summarized in BPMN form in Fig.2. By dealing with delivery notes, the most important data such as cost reference number, delivery date, delivery number and name of supplier were needed to be entered into the system manually. This procedure must be carried out for approx. 50 times per day, which was time-consuming. The delivery notes' critical data should be automatically extracted once received in the target process during digitization. The data should then automatically stored as key-value pairs in the database in .json file. The manual works should be reduced to the minimum and the whole workflow automated in the target situation (see Fig.3).

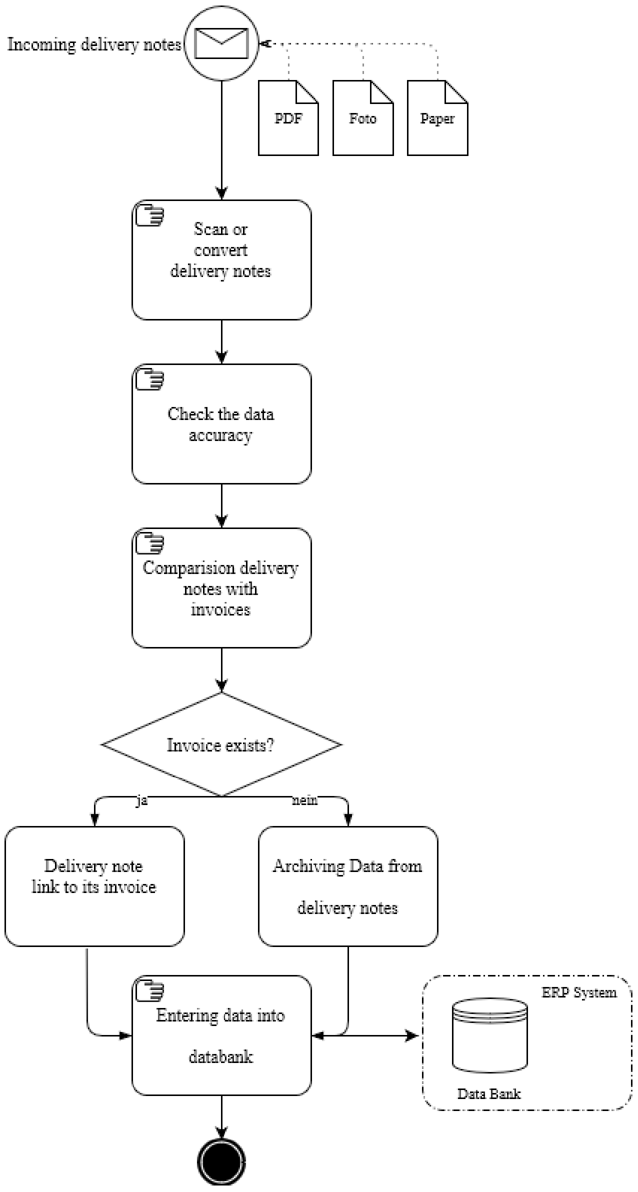

Incoming delivery notes
PDF
Foto
Paper
Scan or convert delivery notes
Check the data accuracy
Comparision delivery notes with invoices
Invoice exists?
ja
nein
Delivery note link to its invoice
Archiving Data from delivery notes
Entering data into databank
ERP System
Data Bank

Fig. 2. The BPMN Diagram of Delivery Notes in the current Situation

3.2 Phase 2 Strategy definition

Based on the first stage results, market research of the exiting Low-Code development platform (LCDP). The most famous LCDP providers examples are listed by Research and Markets (2021) and Waszkowski (2019):

- **Global**: Appian, Oracle APEX, Salesforce, Microsoft PowerApps, MatsSoft.
- **In Europe**: Outsystem (Portugal), Cloudflight (Germany), Mendix(Dutch).

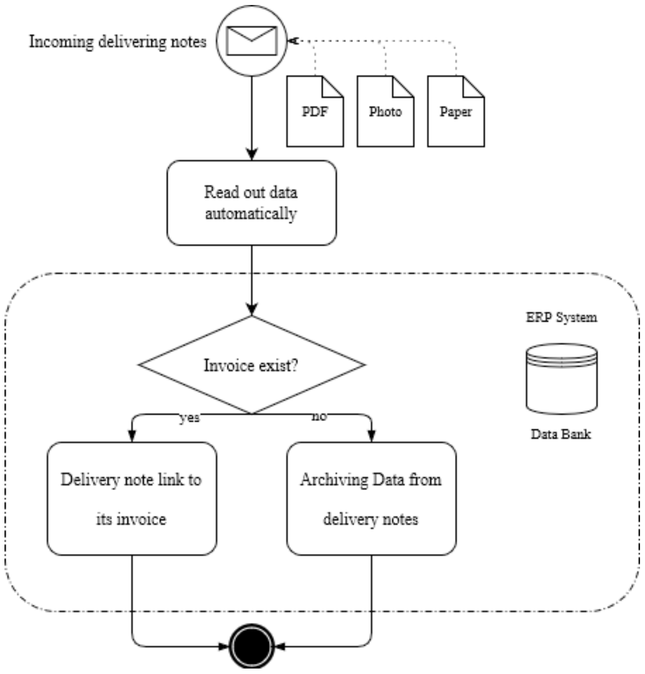

Incoming delivering notes
PDF
Photo
Paper
Read out data automatically
Invoice exist?
ERP System
Data Bank
yes
no
Delivery note link to its invoice
Archiving Data from delivery notes

Fig. 3. The BPMN Diagram of Delivery Notes in the target Situation

- **In Asia**: Tencent (China), Alibaba (China), Huawei AppCube (China), GeneXus (Japan).

Each LCDP has its advantages and disadvantages. In this use case, the LCDP is filtered with the following criteria:

- **Organization and law constrictions**: the LCDP must be compatible with the European law for the digitalization, e.g. DSGVO and GoBD (Goldshteyn and Thelen, 2016; Arning, 2019).
- **System requirements**: the LCDP needs to work with Windows and mobile systems IOS and Android systems.
- **Functional requirements**: the LCDP has modules, which could fulfill the requirements in Tabel 1, e.g. recognize handwritten characters
- **Cost**: the operating costs should be low.

After applying those criteria to the LCDPs, we have chosen Microsoft PowerApps as the development platform.

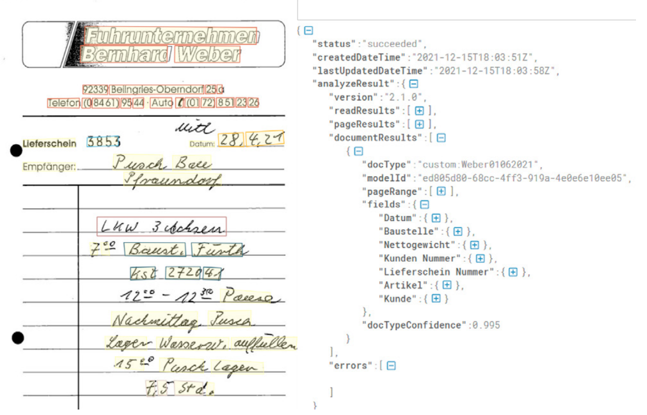
3.3 Phase 3 Implementation

Based on Table 1, the following LCDP modules are selected, as shown in Table 2:

Table 2. The LCDP Modules chosen based on the Functionality Analysis

Functionality	LCPD Module
Laptop and mobile application	Microsoft PowerApp
Automatically extracting text document formats (.pdf, .jpg)	Microsoft Azure Form Recognizer
Automatically importing relevant data into the database	Microsoft SharePoint Microsoft Azure blob container
Summary and sorting of data in the same data group	Microsoft SharePoint
Automatically recognizing handwritten characters	Microsoft Azure cognitive service

Azure Form Recognizer uses a pre-trained machine learning model to extract information from documents (Microsoft Azure, 2021a). However, from the requirement analysis, we know that only crucial data should be extracted from the documentation in this programme. So, we need first to train the Azure Machine Learning (ML) model so that the model would know which information are necessary for our programme. As we have labelled the seven samples of the delivery note, the ML model was available to do the analysis. The samples are very few compared to the standard training progress because the model is pre-trained from Microsoft. By using the Azure Form Recognizer, the workload is highly reduced. Besides, the labeling process is carried out through a graphic user face from Azure, which enables personal with no coding experiences to use it. The results in Fig.4 were obtained:



Fuhrunternehmen Bernhard Weber

92339 Beilngries-Oberndorf 25a
Telefon (08461) 9544 · Auto (0172) 8512326

mitt
Lieferschein 3853 Datum: 28. 4. 21
Empfänger: Pusch Bau
Pfraundorf

LKW 3 Achsen
7⁰⁰ Baust. Furth
Kst 272041
12⁰⁰ - 12³⁰ Pause
Nachmittag Pusch
Lager Wasserw. auffüllen
15⁰⁰ Pusch Lager
7,5 Std.

```
{
  "status":"succeeded",
  "createdDateTime":"2021-12-15T18:03:51Z",
  "lastUpdatedDateTime":"2021-12-15T18:03:58Z",
  "analyzeResult":{
    "version":"2.1.0",
    "readResults":[ ],
    "pageResults":[ ],
    "documentResults":[
      {
        "docType":"custom:Weber01062021",
        "modelId":"ed805d80-68cc-4ff3-919a-4e0e6e10ee05",
        "pageRange":[ ],
        "fields":{
          "Datum":{ },
          "Baustelle":{ },
          "Nettogewicht":{ },
          "Kunden Nummer":{ },
          "Lieferschein Nummer":{ },
          "Artikel":{ },
          "Kunde":{ }
        },
        "docTypeConfidence":0.995
      }
    ],
    "errors":[

    ]
}
```

Fig. 4. Handwritten Delivery Notes (left), Results from the custom trained Model in .Json Format (right)

On the left side was the delivery note with the handwriting characters. The relevant data were recognized and paired with the key. The key-data pairs were stored in .json files in the system, which Microsoft Power App could inquire about. As in the requirements, different delivery note formats should be considered. At this step, customized training models should be made for each format with the same label procedure. The models would be used in the Low-Code application. Based on the BPMN diagram of the target situation (Fig.3), a schema for the workflow was designed in Fig.5.

The Low-Code application "Mein Lieferschein" was created based on the program schema, whose name means "my delivery note" in English. Because the application was developed in German language, the user interfaces' wording was all in German. The Application interface is shown in Fig.6. On the left side, the user could upload pictures or .pdf documents. At the head, the format from different suppliers could be selected. A corresponding ML-Model in the backend is selected for the data analysis later by selecting the supplier. After the user uploads the document, the application automatically analyzes the document. The data would be extracted in the corresponding column. The user could check the data if they are written correctly. If an error occurs in the data, the user could manually correct it. Also, the user could attach documents like notes, pictures individually in the column "attachments". After checking the data, the user could upload the critical data into the

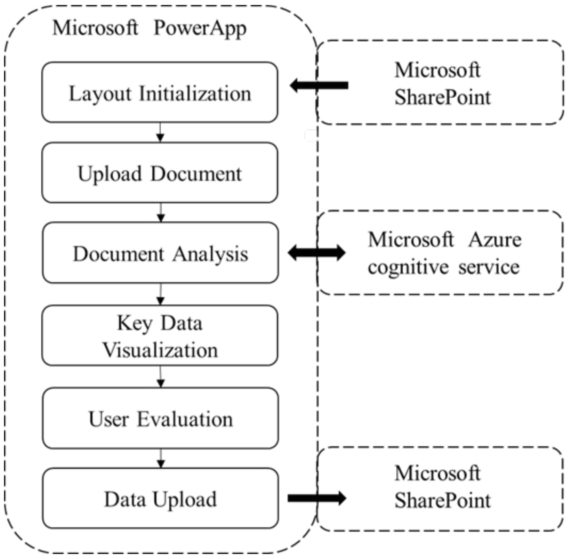


Fig. 5. Programme Schema of the Low-Code Application "Mein Lieferschein"

database in Microsoft SharePoint, which is also shown in Fig.7.

4. DISSCUSSION

This application was then evaluated by the SME involved. The difficulties of the process, namely the recognition of different delivery note formats and handwriting on the delivery bills, were successfully resolved. In addition, the delivery note process was automated, and the manual workload of the process was significantly reduced. The application could be easily adapted to changes in the process and did not incur any costs. In order to use this Low-Code process, it is necessary to subscribe to Microsoft services, which is already available in the company.

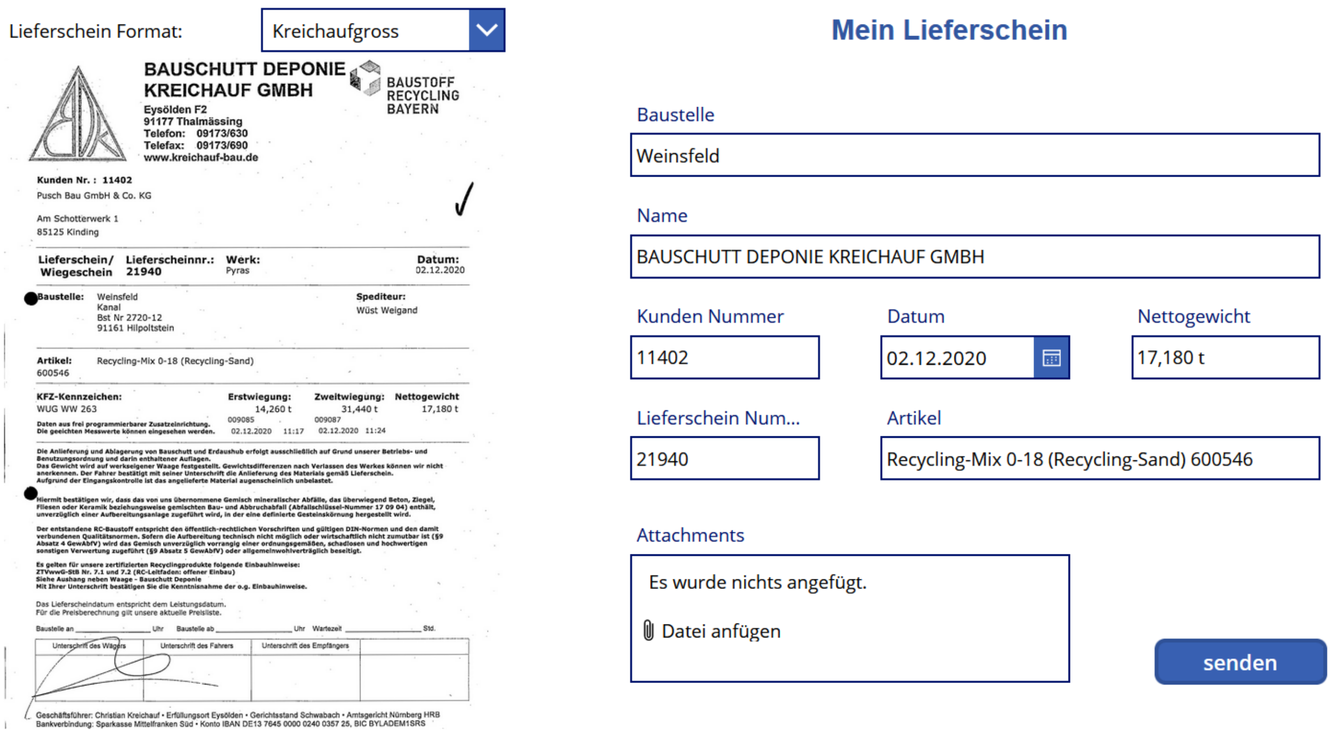
The evaluation of the SME shows that the digitization of the delivery note process through the Low-Code process is successful.

However, there are still some limitations of this application. In Table 3, the accuracy of recognition of handwriting from Microsoft Azure is shown. Because the Azure Model is trained with the samples within the English language region (Microsoft Azure, 2021b), the special characters in the German language, like "ü", could not be identified correctly. In addition, the number "1" in the German is written like "7" in handwriting because of the difference in writing habits. For the above reasons, the correctness of handwriting recognition was not ideal.

Table 3. Accuracy of Handwriting Recognition

Label	Kunde (Client)	Lieferschein Nummer (Delivery note number)	Datum (Date)	Baustelle (Construction Site)
Accuracy	71.9%	99.5%	99.5%	67.8%

The advantages of the application are also evident. The **flexibility** of the application is guaranteed as the application could be used in cross-platform and devices. In addition, the functionality required from the user is mainly met. Delivery notes in different formats from different



Lieferschein Format: Kreichaufgross

BAUSCHUTT DEPONIE KREICHAUF GMBH
Eysölden F2
91177 Thalmässing
Telefon: 09173/630
Telefax: 09173/690
www.kreichauf-bau.de

BAUSTOFF RECYCLING BAYERN

Kunden Nr. : 11402
Pusch Bau GmbH & Co. KG

Am Schotterwerk 1
85125 Kinding

Lieferschein/ Wiegeschein	Lieferscheinnr.: 21940	Werk: Pyras	Datum: 02.12.2020

Baustelle: Weinsfeld
Kanal
Bst Nr 2720-12
91161 Hilpoltstein

Spediteur: Wüst Weigand

Artikel: 600546 Recycling-Mix 0-18 (Recycling-Sand)

KFZ-Kennzeichen:	Erstwiegung:	Zweitwiegung:	Nettogewicht
WUG WW 263	14,260 t	31,440 t	17,180 t
Daten aus frei programmierbarer Zusatzeinrichtung. Die geeichten Messwerte können eingesehen werden.	009085 02.12.2020 11:17	009087 02.12.2020 11:24	

Die Anlieferung und Ablagerung von Bauschutt und Erdaushub erfolgt ausschließlich auf Grund unserer Betriebs- und Benutzungsordnung und darin enthaltener Auflagen.
Das Gewicht wird auf werkseigener Waage festgestellt. Gewichtsdifferenzen nach Verlassen des Werkes können wir nicht anerkennen. Der Fahrer bestätigt mit seiner Unterschrift die Anlieferung des Materials gemäß Lieferschein.
Aufgrund der Eingangskontrolle ist das angelieferte Material augenscheinlich unbelastet.

Hiermit bestätigen wir, dass das von uns übernommene Gemisch mineralischer Abfälle, das überwiegend Beton, Ziegel, Fliesen oder Keramik beziehungsweise gemischten Bau- und Abbruchabfall (Abfallschlüssel-Nummer 17 09 04) enthält, unverzüglich einer Aufbereitungsanlage zugeführt wird, in der eine definierte Gesteinskörnung hergestellt wird.

Der entstandene RC-Baustoff entspricht den öffentlich-rechtlichen Vorschriften und gültigen DIN-Normen und den damit verbundenen Qualitätsnormen. Sofern die Aufbereitung technisch nicht möglich oder wirtschaftlich nicht zumutbar ist (§9 Absatz 4 GewAbfV) wird das Gemisch unverzüglich vorrangig einer ordnungsgemäßen, schadlosen und hochwertigen sonstigen Verwertung zugeführt (§9 Absatz 5 GewAbfV) oder allgemeinwohlverträglich beseitigt.

Es gelten für unsere zertifizierten Recyclingprodukte folgende Einbauhinweise:
ZTVwwG-StB Nr. 7.1 und 7.2 (RC-Leitfaden: offener Einbau)
Siehe Aushang neben Waage - Bauschutt Deponie
Mit Ihrer Unterschrift bestätigen Sie die Kenntnisnahme der o.g. Einbauhinweise.

Das Lieferscheindatum entspricht dem Leistungsdatum.
Für die Preisberechnung gilt unsere aktuelle Preisliste.

Baustelle an ____ Uhr Baustelle ab ____ Uhr Wartezeit ____ Std.

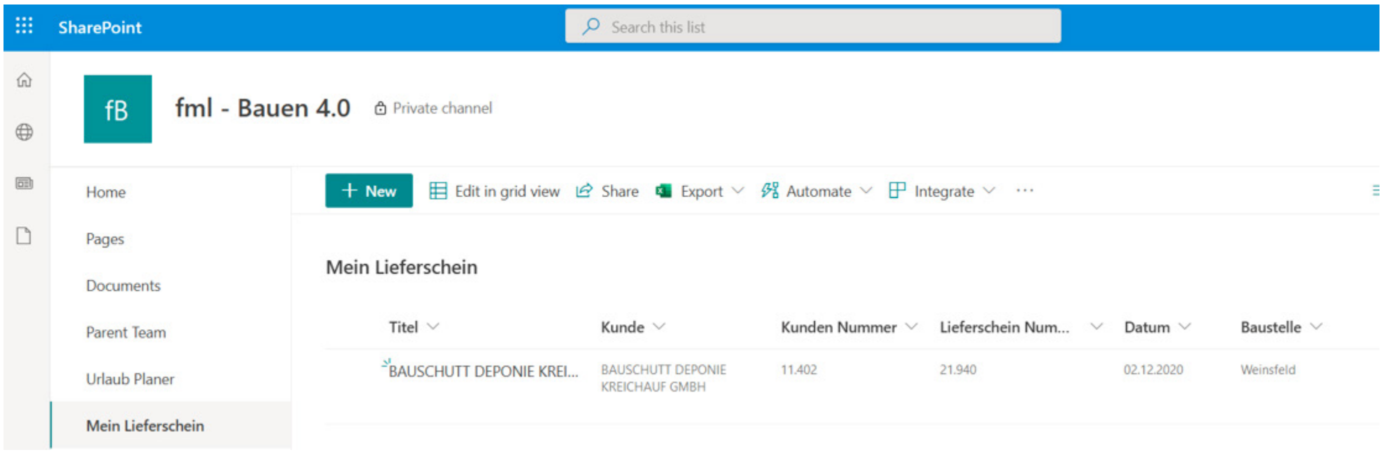
Unterschrift des Wägers	Unterschrift des Fahrers	Unterschrift des Empfängers	
[illegible]			

Geschäftsführer: Christian Kreichauf • Erfüllungsort Eysölden • Gerichtsstand Schwabach • Amtsgericht Nürnberg HRB
Bankverbindung: Sparkasse Mittelfranken Süd • Konto IBAN DE13 7645 0000 0240 0357 25, BIC BYLADEM1SRS

Mein Lieferschein

Baustelle: Weinsfeld
Name: BAUSCHUTT DEPONIE KREICHAUF GMBH
Kunden Nummer: 11402
Datum: 02.12.2020
Nettogewicht: 17,180 t
Lieferschein Num...: 21940
Artikel: Recycling-Mix 0-18 (Recycling-Sand) 600546
Attachments: Es wurde nichts angefügt.

Fig. 6. User Interface of the Low-Code Application “Mein Lieferschein”



Mein Lieferschein

Titel	Kunde	Kunden Nummer	Lieferschein Num...	Datum	Baustelle
BAUSCHUTT DEPONIE KREI...	BAUSCHUTT DEPONIE KREICHAUF GMBH	11.402	21.940	02.12.2020	Weinsfeld

Fig. 7. Database of the Low-Code Application “Mein Lieferschein”

suppliers can be recognized by using different ML-models. With some limitations, the recognition of handwriting is also fulfilled. Furthermore, the **modularity** of this application is shown. Users could add or erase functions of this application easily. For example, if the user wants to add an extra approval process for the delivery note, it can be easily extended in the application. Finally, this application is **cost-effective**, because there was no extra cost caused in this use case.

5. CONCLUSION AND FUTURE WORK

This study has shown that the Low-Code method offers a new approach to simplify the digitization of business processes and support SMEs. After validation by the use case “digital delivery note”, it can be stated that the advantages of the Low-Code method are predominantly in the following three aspects:

- **Flexibility**: The application of the Low-Code method can be used across different systems.
- **Modularity**: The applications or the model are always user and process specific definable.
- **Cost-effective**: The cost of the Low-Code platform with more negligible consumption, such as SME application, is low.

For future research, it is suggested to compare services and workflow from other Low-Code Platforms. The limitations mentioned in the discussion could be overcome by using the services from the European Low-Code platform developers, since they would develop the services under the consideration of the culture specialties, like the special characters in their languages. Another point of future work is to connect the Low-Code application with the document management system (DMS) from the SMEs. In this research work, because of the legal restriction of data protection, we used a separate database in Microsoft SharePoint. Different LCPD provides connections to conventional DMS, such as SAP and Saperion. If we could use the data in the DMS database, more applications and functions could be realized with the Low-Code method.

ACKNOWLEDGEMENTS

This research was funded by the German Federal Ministry of Education and Research within the research project "Building 4.0". The authors gratefully acknowledge the cooperation of Pusch Bau GmbH & Co. KG toward completing the current research project.

REFERENCES

Arning, M. (2019). *DSGVO - BDSG: Kommentar*. Kommunikation & Recht. Fachmedien Recht und Wirtschaft dfv Mediengruppe, Frankfurt am Main, 3., völlig neu bearbeitete und wesentlich erweiterte auflage edition.

Bell, S. (2006). *Lean enterprise systems: Using IT for continuous improvement*. Wiley series in systems engineering and management. Wiley. doi:10.1002/0471756466.

European Commission (2004-2010). Sme performance review. URL https://ec.europa.eu/growth/smes/sme-strategy/performance-reviewensbafactsheets.

Goldshteyn, M. and Thelen, S. (2016). *Praxishandbuch digitale Betriebsprüfung: Anforderungen der neuen GoBD an Buchführung, Datenspeicherung und Datenzugriff*. Schäffer-Poeschel Verlag, Stuttgart.

Hampson, K., Kraatz, J.A., and Sanchez, A.X. (2014). The global construction industry and r&d. 4–23. doi: 10.4324/9781315774916.

Liker, J.K. and Choi, T.Y. (2004). Building deep supplier relationships. URL https://mycourses.aalto.fi/pluginfile.php/675796/course /section/117679/.

McGivern, Y. (2013). *The practice of market research: An introduction*. Always learning. Pearson, Harlow and Munich, 4. ed. edition.

Microsoft Azure (2021a). Form recognizer custom and composed models. URL https://docs.microsoft.com/en-us/azure/applied-ai-services/form-recognizer/concept-custom.

Microsoft Azure (2021b). Form recognizer receipt model. URL https://docs.microsoft.com/en-us/azure/applied-ai-services/form-recognizer/concept-receipt.

Phalake, V.S. and Joshi, S.D. (2021). Low code development platform for digital transformation. In M.S. Kaiser, J. Xie, and V.S. Rathore (eds.), *Information and Communication Technology for Competitive Strategies (ICTCS 2020)*, volume 190 of *Lecture Notes in Networks and Systems*, 689–697. Springer Singapore, Singapore. doi:10.1007/978-981-16-0882-7_61.

Recker, J. (2012). Bpmn research: What we know and what we don't know. In W. van der Aalst, J. Mylopoulos, M. Rosemann, M.J. Shaw, C. Szyperski, J. Mendling, and M. Weidlich (eds.), *Business Process Model and Notation*, volume 125 of *Lecture Notes in Business Information Processing*, 1–7. Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/978-3-642-33155-8_1.

Reinhard Höhn, S.H. (2008). *Das V-Modell XT*. Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/978-3-540-30250-6.

Research and Markets (2021). Global low code development platform market research report: Forecast (2021-2026) - cloud deployment to witness the largest market share. URL https://www.researchandmarkets.com/reports/5317184 /global-low-code-development-platform-market-2021.

Sanchis, R., García-Perales, , Fraile, F., and Poler, R. (2020). Low-code as enabler of digital transformation in manufacturing industry. *Applied Sciences*, 10(1). doi:10.3390/app10010012. URL `https://www.mdpi.com/2076-3417/10/1/12`.

Schawel, C. and Billing, F. (2012). Swot-analyse. In C. Schawel and F. Billing (eds.), *Top 100 Management Tools*, 249–251. Gabler Verlag, Wiesbaden. doi: 10.1007/978-3-8349-4105-3_82.

Tay, S., Te Chuan, L., Aziati, A., and Ahmad, A.N.A. (2018). An overview of industry 4.0: Definition, components, and government initiatives. *Journal of Advanced Research in Dynamical and Control Systems*, 10, 14.

Teich, I., Kolbenschlag, W., and Reiners, W. (eds.) (2008). *Der richtige Weg zur Softwareauswahl: Lastenheft, Pflichtenheft, Compliance, Erfolgskontrolle*. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg. doi: 10.1007/978-3-540-71262-6.

Tippins, M.J. and Sohi, R.S. (2003). It competency and firm performance: is organizational learning a missing link? *Strategic Management Journal*, 24(8), 745–761. doi:10.1002/smj.337.

Wang, Y., Feng, Y., Zhang, M., and Sun, P. (2021). The necessity of low-code engineering for industrial software development: A case study and reflections. In *2021 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 415–420. doi:10.1109/ISSREW53611.2021.00112.

Waszkowski, R. (2019). Low-code platform for automating business processes in manufacturing. *IFAC-PapersOnLine*, 52(10), 376–381. doi: 10.1016/j.ifacol.2019.10.060.

Zaoui, F. and Souissi, N. (2020). Roadmap for digital transformation: A literature review. *Procedia Computer Science*, 175, 621–628. doi:10.1016/j.procs.2020.07.090.