

Checklist-based Support of Knowledge Workers in Robotic Process Automation Projects

Judith Wewerka
*Institute of Databases and
 Information Systems
 Ulm University
 Ulm, Germany
 0000-0002-4809-2480*

Manfred Reichert
*Institute of Databases and
 Information Systems
 Ulm University
 Ulm, Germany
 0000-0003-2536-4153*

Abstract—Robotic Process Automation (RPA) is the rule-based automation of business process tasks by software robots mimicking human interactions. RPA projects often fail or do not achieve the expected benefits due to a missing support of the humans developing the robots. In practice, such software robots are often developed and configured by knowledge workers without any IT background. These non-IT experts require appropriate development and configuration support. The aim of this paper is to provide a checklist-based support of knowledge workers in conducting RPA projects. After an initial design, which is based on interviews with knowledge workers, an exploratory RPA case study in industry, and surveys among bot users, the checklist is improved in three iterations with comprehensive user feedback. The finally obtained checklist is evaluated in six RPA projects in industrial practice showing satisfying results. In particular, no project has failed and the expected full time equivalents savings, reduction of errors, and improvement of process speed have been achieved. Altogether, the checklist enables smooth process support of knowledge workers in RPA projects.

Index Terms—Robotic Process Automation, RPA, RPA Project, Checklist

I. INTRODUCTION

In a continuously changing world, business process automation becomes increasingly important [15] and the resulting process-aware information systems need to become more efficient and cost-effective over time [8]. As a consequence, companies require an increasing degree of process automation. In this context, Robotic Process Automation (RPA) deals with the software-based automation of a business process or parts of it [7], [17]. In particular, software robots (bots for short) shall mimic human interaction with the software systems involved in the execution of business processes [18], [29].

In general, software engineering projects are known to be complex, long-running, and dependent on various factors. Moreover, they are problematic due to budget or schedule overrun or low-quality software [3]. Regarding RPA, which also creates a software artifact, i.e., bot, it is common that employees in the business units develop the bot. Usually, these employees are knowledge workers without any IT background [28], [30]. On one hand, the knowledge workers have good knowledge of the process tasks to be automated; on the other, they do not have profound IT knowledge [28], [30]. Special support is needed to help them develop bots properly.

Furthermore, there exists little research on developing RPA bots in knowledge-intensive domains (e.g., automotive engineering), which are dependent on the employees performing decision making tasks [11], [20]. As a consequence of the missing support of knowledge workers, RPA projects do not always achieve the expected benefits or fail completely [4], [23], [25]. RPA bot development itself is a knowledge-intensive process, which has not been supported systematically enough so far [19]. The aim of this paper is to provide a well designed and evaluated checklist for supporting knowledge workers in RPA projects. Note that checklists are considered as an appropriate means for coordinating knowledge-intensive processes in general [10].

The paper is organized as follows: Section II provides background information on RPA and software development stages. Section III deals with the initial development of the checklist for supporting knowledge workers in RPA projects (cf. Fig. 1). After having designed the checklist, it is refined in Section IV in three iteration steps according to Design Science Research [27]. The finally resulting checklist is evaluated in Section V, followed by a discussion of our results in Section VI. Section VII summarizes the paper and gives an outlook.

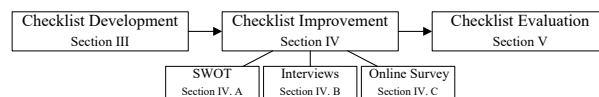


Fig. 1. Overview of checklist development, improvement, and evaluation.

II. BACKGROUNDS

This section provides background information on RPA and software development lifecycle stages.

Robotic Process Automation. The IEEE Standards Association defines RPA as follows [6]: “A preconfigured software instance that uses business rules and predefined activity choreography to complete the autonomous execution of a combination of processes, activities, transactions, and tasks in one or more unrelated software system to deliver a result or service with human exception management.” A detailed example can be found in [24]. RPA automates one or several

tasks of a business process. In this context, a business process consists of a set of process tasks performed in coordination in an organizational and technical environment. These process tasks describe pieces of work to be performed within a certain time period [12]. An instance of a business process is called *case* in the following.

RPA bot development lifecycle. Software development requires various stages for a successful completion: starting from planning and designing the software to its implementation. [16] introduces the development lifecycle to provide a structured way to conduct IT projects. In the context of bot development, we stick to the following four stages (cf. Fig. 2) to structure the checklist for RPA bot development. This structure supports knowledge workers with the transition from their daily tasks to RPA tasks, and provides stability and certainty to them. The *Evaluation* stage focuses on understanding and analyzing the as-is process, which is the candidate to be automated with RPA. The *Preparation* stage defines the to-be process that shall be executed by the bot. The *Implementation* stage includes the development of the RPA bot according to the design defined in the previous stage. The bot is tested to determine whether it behaves correctly. Finally, the *Maintenance* stage deploys, maintains, and operates the RPA bot. Moreover, it measures its performance.

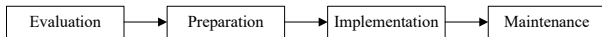


Fig. 2. Software development lifecycle stages according to [16].

III. INITIAL DEVELOPMENT OF THE CHECKLIST

This section develops an initial version of the checklist that shall support knowledge workers in developing RPA bots. The checklist is designed based on previous work (cf. Fig. 3). In particular, we carry out **interviews** with 12 knowledge workers who fill the role as RPA developers in engineering and we distribute **questionnaires** to them. The goal is to understand the effects currently achieved with RPA projects in the engineering domain [23]. We discover that the positive benefits promised in RPA literature are not always achieved in practice. In particular, shorter case duration and better quality are not confirmed by the empirical data gathered in the considered RPA projects.

Through an **exploratory case study** of three RPA projects in the automotive domain, we identify five main challenges in current RPA projects [25]. The challenges that should be tackled for any successful RPA development are as follows: (1) identifying the right process to automate, (2) understanding the factors influencing user acceptance, (3) explaining RPA to the users, (4) designing human bot interaction, and (5) providing software development guidelines for RPA development.

The current state-of-the-art is analyzed with a **systematic mapping study** [24]. Using the classification framework resulting from this mapping study, we determine challenges that have not been addressed in literature, i.e., Challenges (2), (4), and (5).

To overcome Challenge (2), we develop and empirically validate an **RPA user acceptance model** with 50 RPA users in engineering [21]. Our results indicate that the most important factors, which significantly influence perceived usefulness and perceived ease of use of RPA bots, are facilitating conditions, result demonstrability, innovation joy, and social influence. These findings are used to derive recommendations for the design and development of RPA bots increasing acceptance of employees using the bots during their daily work.

Challenge (4) is addressed by **evaluating the usability of RPA bots** in industry and providing user interface design guidelines to bot developers [22]. The results, which we obtain from 50 questionnaires filled by RPA users, indicate that error tolerance, perceptibility, directability of user's attention, suitability for the task, and availability of the RPA bot need to be improved. We derive seven guidelines for designing the user interface of RPA bots. Potential improvements include the quality of error messages, the efforts for error-handling, and the monitoring of the current status of the tasks assigned to the bot.

The findings we reported in [21], [22] refer to single items of the overall checklist, which are assigned to the development stages. Hence, a checklist to address Challenge (5) is derived. The checklist is aligned along the software development lifecycle (cf. Fig. 2). The initial version can be found in Fig. 4.

IV. ITERATIVE IMPROVEMENT OF THE CHECKLIST

The three checklist refinement steps are presented in the following sub-sections. The refinement is based on criteria we retrieve from literature: The content of the checklist is **applicable** and **relevant** in practice. The formulations are **clear and simple**, and they foster **understandability**. The checklist is **useful** and **easy to use** [2], [5], [9], [13], [26].

In order to decide, which checklist items are relevant, we define the following *inclusion criteria*:

- *Inclusion Criterion 1:* The checklist is kept general, i.e., it contains no project-, company-, or RPA provider-specific information.
- *Inclusion Criterion 2:* The checklist is primarily intended for knowledge workers with basic RPA knowledge who shall develop RPA bots.
- *Inclusion Criterion 3:* The checklist provides general checklist items and does not cover technical issues.

Overall, we want to thoroughly understand the pros and cons of the initially proposed checklist. In particular, missing checklist items shall be identified and a wide range of improvements should be gathered.

A. First Checklist Refinement Iteration - SWOT Analysis

We first perform an analysis of the strengths, weaknesses, opportunities, and threats (**SWOT**) of the initially proposed checklist [1]. Strengths/weaknesses are advantages/disadvantages of the checklist that arise from its use. Opportunities, in turn, are features of the checklist that can be used, whereas threats examine problems that the checklist

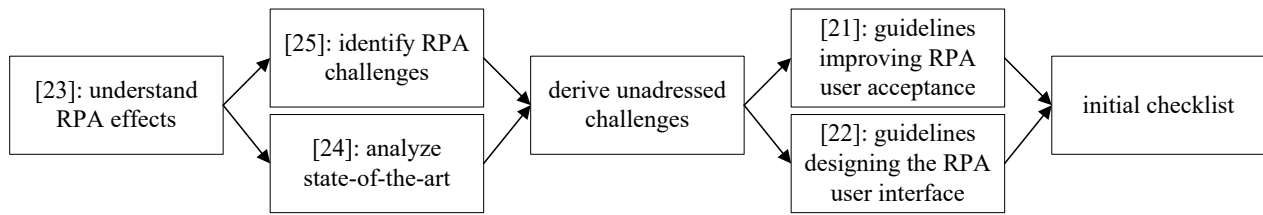


Fig. 3. Overview and context of previous research to derive initial checklist.

<p><u>Evaluation:</u></p> <ul style="list-style-type: none"> • Examine the proposed process: <ul style="list-style-type: none"> - The process will not change in the near future. - The process is rule-based. - The process is highly manual and repetitive. - The process has digital input and output. - The process does not have a lot of variants. - The process is well analyzed and documented. - The process saves 0.5 Full Time Equivalents (FTE). • Does your management support the implementation of the RPA bot? <p><u>Preparation:</u></p> <ul style="list-style-type: none"> • Have you talked to the end-users and have you understood their expectations? • Have you considered the following guidelines for a user-friendly design of the bot? <ul style="list-style-type: none"> - In case of an error, the bot provides an understandable error message that helps the end-users to fix the error. - In case of an error, the end-users can easily correct the mistake by themselves. 	<ul style="list-style-type: none"> - The end-users can always see the status of their task. How did you design this guideline? - The end-user's attention is attained if bot finishes or fails. How did you design this guideline? <p><u>Implementation:</u></p> <ul style="list-style-type: none"> • Does the bot work reliable? • Does the bot produce results within a reasonable response time? • Is the bot trustworthy? • Have you documented the implementation in detail? • Can the RPA bot be used without additional efforts? <p><u>Maintenance:</u></p> <ul style="list-style-type: none"> • Have you offered training sessions for the end-users? • Have you provided user manuals for them? • Have you established RPA opinion multipliers, i.e., colleagues advertising the RPA bot? • Have you demonstrated the advantages of RPA?
---	--

Fig. 4. Initial version of the checklist to support knowledge workers in RPA projects.

could cause [1]. Strengths and weaknesses are internal aspects for the person using the checklist, i.e., the RPA developer. Opportunities and threats are external aspects for the person using the resulting bot, i.e., the end-user [14]. Strengths and opportunities are helpful for achieving goals, whereas weaknesses and threats are harmful for achieving them. The results of the SWOT analysis (cf. Table I) can be used to further evolve the checklist.

Based on the SWOT analysis, we derive several checklist improvements. First, we emphasize the **strengths** of the checklist. To the best of our knowledge, there is no other artifact supporting knowledge workers during RPA projects (i.e., bot development). The checklist reduces the efforts needed to search for information on bot development and the developer can work without support and, therefore, is more autonomous. For knowledge workers the use of the checklist is intuitive, as they are familiar with working with checklists [10]. The checklist provides insights for RPA developers on user requirements as it is based on comprehensive research (cf. Fig. 3). These include user acceptance factors as well as recommendations for designing RPA user interfaces. The

checklist itself provides a good overview on necessary steps to develop RPA bots and is independent of any RPA tool provider. However, it can be easily adapted and expanded to project- or company-specific needs.

Second, regarding **weaknesses**, the checklist might be too unspecific and general. As our goal is to provide rather general guidelines (cf. *Inclusion Criterion 1*), this weakness is not considered further. However, the checklist needs to be improved to overcome the weakness that knowledge workers need to understand the formulations of the checklist items. Hence, explanations need to be included, e.g., what *rule-based processes* or *process variants* are. The *implementation* stage might take longer when following all checklist items. However, this weakness can only be assessed if the checklist is used during RPA projects. In particular, the strengths should outweigh this specific weakness. Moreover, one can argue that no technical information is provided by the checklist. Technical issues, e.g., naming conventions, are often handled by companies. Therefore, we decide to not include them in the general checklist. Companies that want to apply the checklist, however, may customize it to their specific needs (*Inclusion*

TABLE I
RESULTS OF THE SWOT ANALYSIS.

	Helpful for achieving goals	Harmful for achieving goals
Internal	<ul style="list-style-type: none"> • No other artifact is known that supports knowledge workers in developing RPA bots. • The checklist reduces the efforts to search for information. Moreover, knowledge workers are more autonomous. • The checklist is easy to use as knowledge workers are used to work with checklists. • The checklist provides insights for the RPA developer on user requirements, e.g., acceptance factors and user interface design. • The checklist provides a good overview on necessary steps to develop RPA bots. • The checklist is independent of the RPA tool provider. • The checklist is easily adaptable and expandable. 	<ul style="list-style-type: none"> • The checklist is not specific enough due to general guidelines. • Knowledge workers need to understand the wording of the checklist items. • The checklist might cause more work to develop RPA bots. • No technical information is given in the checklist. • The RPA developer is not aware of the checklist.
External	<ul style="list-style-type: none"> • Checklist items address improvements for user experience. • Checklist items enable a user-friendly design of the RPA bot. • Checklist items include recommendations to increase RPA user acceptance. • Checklist items cover easy error-handling. • The checklist promotes the success of RPA projects. 	<ul style="list-style-type: none"> • End-users do not care whether the RPA developer uses the checklist or not.

Criterion 3 is not fulfilled). The weakness that the developer is not aware of the checklist, constitutes a serious concern. Here, the checklist needs to be communicated to the RPA developers.

Third, as potential **threat** we consider that the end-users do not care whether or not the bot has been developed based on the checklist. Note that this threat is not relevant, as it does not change anything. Still, the checklist supports the knowledge worker during bot development.

Finally, the **opportunities** for bot users are immense: checklist-based bot development improve user experience, increase user acceptance, and possess a user-friendly design with easy error-handling. Overall, the checklist promotes the success of RPA projects.

B. Second Checklist Refinement Iteration - Interviews with RPA developers

In the second step, we randomly choose six potential users of the checklist, i.e., knowledge workers who are RPA developers in the automotive industry. Each **interview** is conducted via *MS Teams* and lasts around 30 minutes. We present the checklist and ask whether checklist items are missing or existing checklist items shall be left out. Moreover, we ask for recommendations for improvements including the wording and visual representation to further sharpen the checklist. The RPA developers are selected based on various experience levels with RPA:

- Interviewee 1: 1.5 years of communicating, supervising, and programming RPA bots.
- Interviewee 2: 4 years of experience with RPA; established RPA in the company and works in the Center of Excellence for RPA.

- Interviewee 3: 9 months of scientifically working with RPA, coauthor of [22].
- Interviewee 4: 8 months of scientific experience with RPA, coauthor of [21].
- Interviewee 5: 1.5 years of experience with managing RPA projects.
- Interviewee 6: 1.5 years of communicating and explaining RPA to employees.

We systematically analyze the interview responses and divide them into the following categories:

Category 1: General information. Comments in this category emphasize the relevance of several checklist items and indicate which items should be kept in the checklist:

- *Examination of the process before RPA development, especially regarding its rule-basedness.*
- *Looking at Full Time Equivalents (FTE) savings.*
- *Standardization of the process to be automated.*
- *Enhancing reliability of the bot and the corresponding recommendations.*

Category 2: Recommendations for visual representation. Some interviewees provide ideas for a good presentation of the checklist to facilitate its usage. One idea is to consider the software development model already used in the company for the checklist. Another one is to use a uniform sentence structuring and appearance. The former violates *Inclusion Criterion 1*, the latter is implemented in the checklist.

Category 3: Improvements regarding wording. Many recommendations are made to improve the labeling of checklist items and, therefore, to facilitate the understanding of the checklist. These minor changes concern

- better explanations of concepts, e.g., *rule-based* and *FTE savings*,

- improved labeling, e.g., *session* instead of *training*,
- avoidance of ambiguities, e.g., *guideline*, and
- completion of sentences with details, e.g., what is meant by *non-changing IT systems*.

Category 4: Missing checklist items. Regarding the contents of the checklist, the interviewees identify several items that should be included in the checklist:

- At the beginning of the RPA project, clarify who is responsible for maintenance and operation of the RPA bot.
- Ensure management support right at the start, including the advertisement of the planned bot.
- Get early end-user feedback.
- Continuously monitor and improve the usage of the bot.
- In case of an error, guarantee that the end-user can not only correct the error, but also finish the task manually.
- Reuse code from existing bots.
- Differentiate between *Go Live* and *Maintenance* stage. The former deploys the RPA bot once, whereas the latter maintains and operates the RPA bot continuously.

Category 5: Checklist items to be left out. Only two items contained in the initial checklist are considered as candidates for being left out:

- *The process is well analyzed and documented.* Here, three interviewees independently agreed on excluding this checklist item. The respective process should be analyzed in detail as part of the RPA project, i.e., its detailed analysis is not a prerequisite.
- *How did you design this guideline?* According to the interviewee, it is not interesting how the design is determined, as the checklist should provide support and not force the developer to document design elements. We agree on this opinion and leave the question out.

After implementing the recommended changes in the checklist, it is sent to all six interviewees to confirm its correctness and completeness.

C. Third Checklist Refinement Iteration - Online Survey

During the third iteration, we design an **online survey** for RPA developers. The latter are invited via e-mail, Twitter, and different LinkedIn and Xing RPA groups. Over a period of two weeks, 89 filled questionnaires are gathered. The survey is available in English and German¹ and is structured as follows: In the first part, for each lifecycle stage (cf. Fig. 2), the respondents are asked to assess applicability, relevance, clarity and simplicity of formulations as well as understandability of the checklist items on a 5-point likert scale ranging from 1 (“strongly disagree”) to 5 (“strongly agree”). In addition, usefulness and ease of use of the checklist are assessed. In the second part, questions about the merits and drawbacks of the checklist as well as recommendations for improving checklist items or adding missing ones are provided, which can then be answered with free text. Additionally, we gather information on the age of respondents, their business area,

¹The survey is accessible via <https://survey.sogosurvey.com/t/tbQTIg>.

their experience with RPA (in years), and the number of completed RPA projects².

We first present general information, i.e., **age**, business area, and RPA experience of the 89 respondents. The majority is younger than 30 years (53%, N=47); 31% (N=28) of the respondents have an age between 30 and 40 years, 10% (N=9) are between 41 and 50 years old, and 6% (N=5) are older than 50 years (cf. Fig. 5, left). The right side of Fig. 5 shows the **business areas** of the respondents. Note that the latter may belong to several areas and, therefore, the total number exceeds 89. The majority works in IT (44%, N=50). 19% (N=22) work in Automotive, and 20% (N=23) in Finance. Healthcare (5%, N=6), Research (7%, N=8), and other business areas (5%, N=6) are only represented with a small number of respondents.

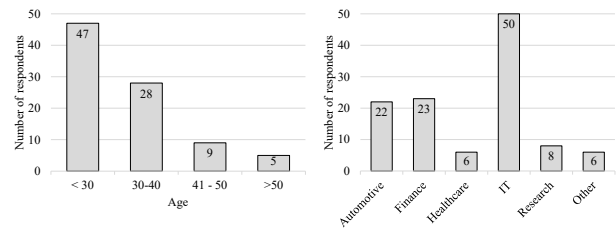


Fig. 5. Age and business area of the respondents.

The **RPA experience** of the respondents is measured with two questions. First, we ask them in how many RPA projects they have been involved: 38% (N=34) have been involved in more than ten RPA projects, 24% (N=21) in six to ten projects, 13% (N=12) in three to five projects, 19% (N=17) in one or two projects, and 6% (N=5) in no RPA project so far (cf. Fig. 6, left). Regarding the years of RPA experience, 15% (N=13) have worked less than one year with RPA, 34% (N=30) one or two years, 40% (N=36) three to five years, and 11% (N=10) more than five years (cf. Fig. 6, right). Hence, the respondents are rather young and experienced with RPA, not only in terms of years working with RPA but also in terms of the number of RPA projects they have been involved in.

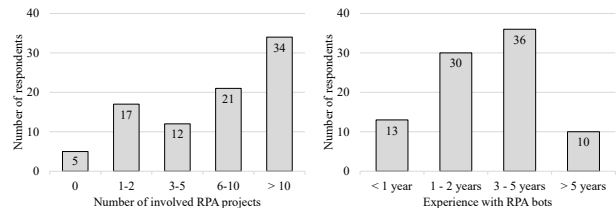


Fig. 6. RPA Experience measured in number of involved RPA projects and years.

The answers to the first part are summarized in Table II. Overall, all values exceed 4.15, which is a highly satisfying result. In detail, **applicability** ranges from 4.21 in *implementation* up to 4.42 in *evaluation*, the mean value is 4.30. **Rele-**

²The raw data are available at https://www.researchgate.net/publication/351714177_Raw_data_of_online_survey.

TABLE II
RESULTS OF THE ONLINE SURVEY (SCALE FROM 1="STRONGLY DISAGREE" TO 5="STRONGLY AGREE").

	Evaluation	Preparation	Implementation	Go Live	Maintenance	Mean
applicable	4.42	4.27	4.21	4.34	4.27	4.30
relevant	4.20	4.34	4.28	4.19	4.29	4.26
clearly and simply	4.41	4.34	4.41	4.55	4.39	4.42
understandable	4.56	4.38	4.42	4.58	4.40	4.47
useful						4.24
easy to use						4.32
Mean	4.40	4.34	4.33	4.41	4.34	4.34

vance lies between 4.19 for *go live* and 4.34 for *preparation* with a mean value of 4.26. **Clear and simple formulations** are rated highest with 4.55 for *go live* and lowest with 4.34 for *preparation*. The mean value is 4.42. **Understandability** of the checklist is between 4.38 in *preparation* and 4.58 in *go live*. The mean value is 4.47. **Usefulness** is assessed with 4.24 and **ease of use** with 4.32. Overall, understandability is the criterion with the best assessment (4.47) and usefulness is rated lowest (4.24). Regarding the different lifecycle stages, *evaluation* is assessed in mean with 4.40, *preparation* and *maintenance* both with 4.34, *implementation* with 4.33, and *go live* with 4.41. Hence, *implementation* is the stage with the lowest rating and *evaluation* is the one with the highest one. To conclude, the overall mean of all criteria is 4.34, which is on a scale ranging from one to five, highly satisfactory.

Finally, we evaluate the second part, i.e., the free text fields on the perceived merits and drawbacks, recommendations for improvement, and missing checklist items. Regarding perceived **merits** of the checklist (cf. Fig. 7), the respondents emphasize that the checklist provides a good overview (N=11), is especially useful for initial RPA projects (N=8), and is comprehensive (N=7). They state that the checklist establishes a sound standard procedure for performing RPA projects (N=7) and that all relevant project aspects are covered (N=6). Additionally, the checklist prevents developers from omission errors (N=5) and is simple (N=5), short and sweet (N=4), and to the point (N=3). Finally, the checklist is comprehensive for users without any IT background (N=3).

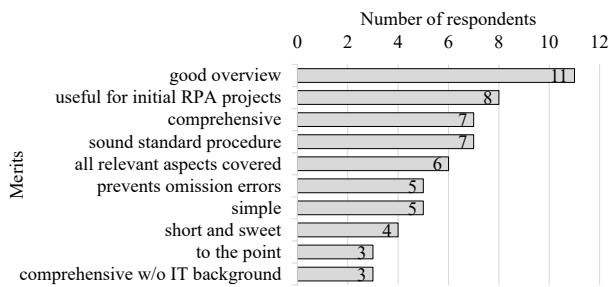


Fig. 7. Merits of the checklist.

The perceived **drawbacks** of the checklist (cf. Fig. 8) include that it covers not enough details (N=11), it should provide an explicit support of each lifecycle stage (N=6), its items are not complete (N=3), and it is not project-specific (N=3). Several respondents remark that no explanation of RPA

is given (N=3) and there is no information for experienced RPA developers (N=2). However, four answers indicate that there are no significant drawbacks at all.

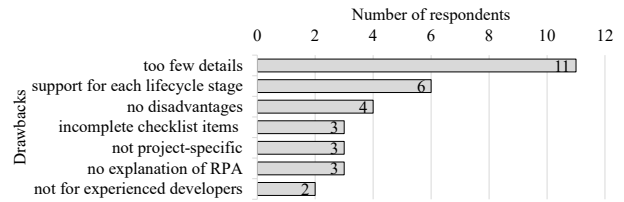


Fig. 8. Drawbacks of the checklist.

Concerning the **recommendations** given for improving the checklist (cf. Fig. 9), some respondents ask for more details (N=7), but without giving examples. Seven respondents suggest concrete improvements for specific lifecycle stages, e.g., to evaluate whether the IT system is suited for automation (*evaluation*) or to identify and save reusable code components (*implementation*). Three different improvements regarding the checklist design are given, namely to use check boxes for every item in the checklist, to represent the checklist as a table, and to implement its logic using a workflow tool. We decide to follow the idea of the check boxes. Further representation ideas can be either realized by the RPA developer or the company adopting the checklist.

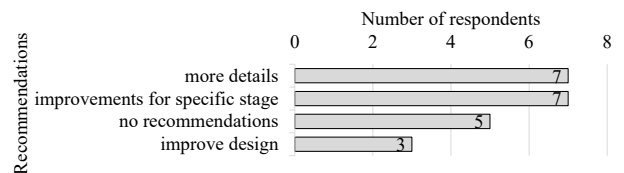


Fig. 9. Recommendations to improve the checklist.

We categorize the **missing checklist items** mentioned by the respondents as follows (cf. Fig. 10): several respondents state that no items are missing (N=7). Six respondents ask for more information on data protection and testing or validation of the implemented bot. RPA knowledge (N=4) and stakeholders (N=4) shall be included. Other missing items include the business benefits (N=3) and information regarding the coordination with the works council (N=2). According to two respondents, administrative tasks, e.g., requesting access rights for the bot, should be included.

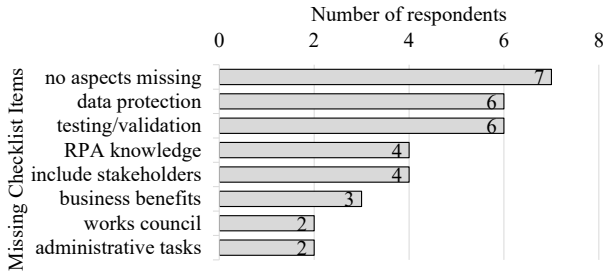


Fig. 10. Missing checklist items.

Based on the survey results, we add the following checklist items:

- *Evaluation*: we add an item to calculate the expected FTE savings. Furthermore, we add an item right at the beginning, i.e., the involvement of subject matter experts to understand the task correctly. Additionally, to complete the *evaluation* stage, the general feasibility of RPA should be evaluated based on the checklist items in this stage and an appropriate item is added.
- *Implementation*: we add an item to identify and save reusable components. The testing of the RPA bot has been neglected so far. Taken the various comments into account, we add an item, i.e., *Test the RPA bot* and some sub-items to consider, e.g., to create a testing environment, to select release criteria, and to perform quality assurance tests. Another item, which has not been addressed, is security. Therefore, we add the checklist item *Implement the RPA bot securely*. Secure implementation includes that the RPA bot only has the roles and access rights needed to complete its tasks and that try-catch statements are used for error-handling.
- *Maintenance*: two items are addressed, i.e., to check the bot daily and to maintain it regularly. We try to ameliorate the phrases in maintenance to account for these demands.

The checklist improved with the results of the three iterations can be found in Fig. 15 in the Appendix³.

V. EVALUATION OF THE CHECKLIST

The derived checklist is evaluated in six RPA projects, i.e., P01-P06. We provide the checklist to the knowledge workers developing the RPA bots and accompany as well as observe the projects to measure FTE savings, error reduction, and mean case duration to assess whether or not the checklist helps to better achieve RPA benefits. Note that shorter case duration and better quality have not been confirmed by the empirical data gathered in previous RPA projects [23]. Table III provides a short overview on the six projects. Note that more detailed information cannot be given due to privacy reason.

Fig. 11 shows the **FTE savings** for the six RPA projects (P01-P06). Three projects save more than 1FTE (P01, P04, and

³An English and a German version are available at https://www.researchgate.net/publication/351714181_Final_Checklist_English_and_German.

TABLE III
OVERVIEW OF THE SIX RPA PROJECTS.

Project	Bot description
P01	Bot fills a template with data from various systems.
P02	Bot copies and pastes data from Excel files into an Enterprise Resource Planing (ERP) system.
P03	Bot reads an Excel file and creates corresponding entries in the ERP system.
P04	Every hour, the bot downloads recent data from systems for visualization.
P05	Bot opens different calculations and saves data in one central location.
P06	Bot checks an information system and sends an e-mail as soon as new content becomes available.

P05), with P01 achieving FTE savings of 2.86FTE. The other three projects have medium FTE savings between 0.54FTE and 0.70FTE.

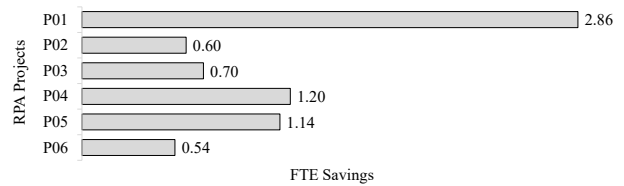


Fig. 11. FTE Savings per RPA project.

Better result quality is measured by the percentage of cases with error before and after RPA introduction. Fig. 12 visualizes this measure: three projects (P02, P03, and P06) reduce the number of cases with error by 46% to 71%. P01 implements a business process that has no cases with error before RPA introduction and the RPA bot works perfectly not causing any errors. For P04, it is not possible to get the required data. P05 implements a business process that has not been performed manually. Only with RPA, it becomes possible to save the calculation data as required. Therefore, before RPA introduction no data is available.

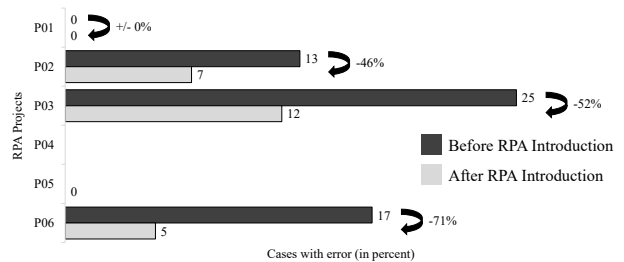


Fig. 12. Percentage of cases with error before and after RPA introduction and percentage change.

Finally, we assess whether the RPA introduction increases process speed. Two measures are evaluated in this context, i.e., the number of cases per day and the mean case duration. For three processes (P02, P03, P06), the **number of daily cases** is the same before or after introducing RPA (cf. Fig. 13). Again,

P05 has not been executed manually, therefore, the number of cases per day is not available before RPA introduction. In two projects, RPA allows increasing the number of daily cases: P01 manages 40 cases per day before and 100 cases per day after RPA introduction (+150%); P04 increases the number of daily cases from one to 17 (+1600%).

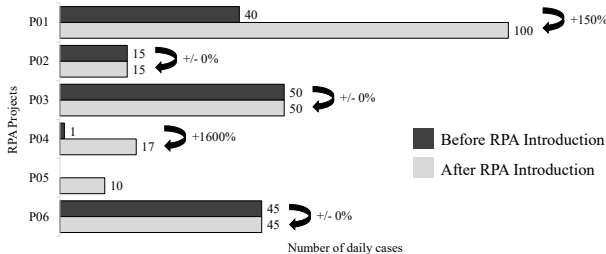


Fig. 13. Number of daily cases before and after RPA introduction and percentage change.

The **mean case duration** does not change for P03 as well as P06. It is not assessable for P05 (cf. Fig. 14). For the other projects, a clear reduction in mean case duration can be achieved after introducing RPA. P01 decreases the mean case duration from 12 minutes to 1.34 minutes (-89%), P02 from 16 minutes to 6 minutes (-63%), and P04 from 45 minutes to 30 minutes (-34%).

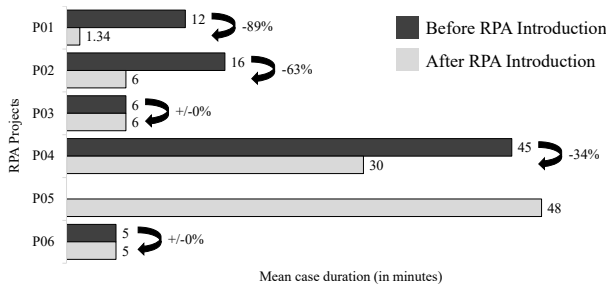


Fig. 14. Mean case duration in minutes before and after RPA introduction and percentage change.

VI. DISCUSSION

Regarding the **strengths** and **merits** of the checklist, the results of the SWOT analysis and the online survey overlap: the checklist is a good and short overview of necessary steps to develop RPA bots. Further, it is easy to use and can be easily adapted and extended. Additionally, it helps preventing omission errors in RPA projects. In the SWOT analysis, we emphasize that the checklist is useful as it is independent of any RPA tool provider. Moreover, no other artifact is known that supports knowledge workers in conducting RPA projects, and insights into users requirements are given. None of these merits is picked up in the online survey, but several respondents point out that the checklist is very useful in areas where RPA projects are conducted for the first time.

Concerning **weaknesses** and **drawbacks**, only one aspect is present in both the SWOT analysis and the online survey, i.e., the checklist is not specific, but rather general. The weakness that the user has to understand the wording (SWOT analysis) is not confirmed by the online survey. Indeed, it is mentioned that the checklist is comprehensive. This might be due to the numerous wording improvements suggested during the interviews. As general drawbacks no technical information, no information on RPA, and no details on dedicated lifecycle stages are provided. We are aware of these drawbacks. However, we consciously opt for general guidelines, which can be filled with details by RPA developers or companies adopting the checklist (cf. *Inclusion Criterion 1*).

Missing checklist items addressed in the interviews and the online survey are different, i.e., no common items can be identified. Therefore, the missing items identified during the interview, e.g., ensure management support, monitor the bot, and reuse of code, are sufficiently addressed. Most of the comments made by the online survey are too general. However, we are able to identify several checklist items that may be improved, which mainly focus on the *evaluation* and *implementation* stages.

The checklist fulfills all criteria (cf. Section IV) in a highly satisfying manner: The latter are evaluated with means between 4.24 and 4.47 on a scale from one to five. The assertions of the SWOT analysis are confirmed, i.e., the checklist can be **easily used** (4.32) and it is **useful** (4.24). One weakness revealed by the SWOT analysis concerns understandability. In the online survey, **understandability** is evaluated with a mean of 4.47, i.e., this weakness is overcome. Regarding the values of the lifecycle stages, the assessments range between 4.33 and 4.40. Even though *evaluation* is assessed as best stage, five missing items concern this first stage. Apart from *evaluation*, missing checklist items are detected in the *implementation* stage, which has the lowest assessment (4.33) of all stages. Overall, even though the respondents are rather young (average age around 33 years) and experienced with RPA (average experience around three years and seven developed RPA bots), the results indicate that the checklist is applicable, relevant, clearly and simply formulated, understandable, useful, and easy to use. All lifecycle stages are sufficiently covered.

The practical application shows the relevance of the checklist. The high **FTE savings** suggest several improvements in performing RPA projects with the checklist compared to RPA projects without: First, process tasks that are better suited for automation are chosen. Only projects automating tasks suitable for RPA can achieve high FTE savings. Second, RPA bots developed with the checklist are used more often than bots developed without the checklist. No matter how good the process is suited for automation, FTE savings depend on the end-users. If the latter refuse bot usage, no FTE savings can be achieved at all. Therefore, RPA user acceptance improves for the RPA projects using the checklist.

Regarding the **cases with error** before and after RPA development, three process tasks improve, one remains the same, and for two no data is available. Note that cases with

error using the RPA bot are mostly due to employees not providing the correct input [25]. Hence, the checklist helps RPA developers in developing less error-prone RPA bots.

The speed of process execution is evaluated with two measures. First, we look at the **number of daily cases**: at least two projects using the checklist increase the number of daily cases. Therefore, RPA possibilities are more manageable for the RPA developers. Further, P05 is a newly created process task, which would not exist without RPA. Second, the **mean case duration** is evaluated. For projects using the checklist, the percentage change regarding mean case duration ranges between -89% and +/0% with an average of -31%. Therefore, the checklist helps to speed up business process tasks with RPA bots. The developers adhere to the guideline to provide answers to the end-users within a reasonable response time.

In conclusion, in the investigated RPA projects, the ones using the checklist result in significantly better effects than the ones without a checklist [23].

VII. SUMMARY AND OUTLOOK

In this paper, we present checklist-based support for knowledge workers in developing RPA projects. The checklist shall help knowledge workers to develop first RPA bots. Its initial design is based on the identification of RPA challenges [25] as well as guidelines to improve RPA user acceptance [21] and to design RPA user interfaces [22]. The initial design of the checklist is then iteratively improved over three cycles. Each improvement cycle aims to understand the pros and cons of the checklist and to identify missing checklist items. The final checklist is evaluated in six real-world RPA projects. By using the checklist, none of the projects fails and the expected benefits such as FTE savings, error reduction, and faster process execution are achieved. In future research, we will focus on further improvements of the checklist including guidelines for updating (e.g., versioning). Further, we will evaluate the checklist with project participants as well as in RPA projects from various domains. However, given the satisfying results, we are certain that the current checklist already adds value to RPA projects and should be accessible to all.

REFERENCES

- [1] L. G. Fine, "The SWOT analysis: Using your Strength to overcome Weaknesses, Using Opportunities to overcome Threats," *CreateSpace Indep Publ Platf*, 2009.
- [2] D. H.-L. Goh, A. Chua, D. A. Khoo, E. B.-H. Khoo, E. B.-T. Mak, and M. W.-M. Ng, "A checklist for evaluating open source digital library software," *Inform Rev*, vol. 30, no. 4, pp. 360–379, 2006.
- [3] G. R. Grambow, "Context-aware Process Management for the Software Engineering Domain," Ph.D. dissertation, Universität Ulm, 2016.
- [4] L.-V. Herm, C. Janiesch, A. Helm, F. Imgrund, K. Fuchs, A. Hofmann, and A. Winkelmann, "A Consolidated Framework for Implementing Robotic Process Automation Projects," in *18th Int Conf Bus Proc Manag*, 2020, pp. 471–488.
- [5] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Q*, vol. 28, no. 1, pp. 75–105, 2004.
- [6] "IEEE Guide to Terms and Concepts in Intelligent Process Automation," IEEE Std 2755-2017, 2017.
- [7] M. Lacity, L. Willcocks, and A. Craig, "Service Automation: Cognitive Virtual Agents at SEB Bank," *Outs Unit Work Res Pap S*, vol. 17, no. 1, pp. 1–29, 2017.
- [8] M. Lohrmann and M. Reichert, "Effective application of process improvement patterns to business processes," *Softw & Syst Model*, vol. 15, no. 2, pp. 353–375, 2016.
- [9] W. Martz, "Validating an evaluation checklist using a mixed method design," *Evaluat progr planning*, vol. 33, no. 3, pp. 215–222, 2010.
- [10] N. Mundbrod, F. Beuter, and M. Reichert, "Supporting knowledge-intensive processes through integrated task lifecycle support," in *19th Int Ent Dist Obj Comp Conf*, 2015, pp. 19–28.
- [11] N. Mundbrod, J. Kolb, and M. Reichert, "Towards a system support of collaborative knowledge work," in *10th Int W Bus Proc Manag*, 2012, pp. 31–42.
- [12] N. Mundbrod and M. Reichert, "Flexible Task Management Support for Knowledge-Intensive Processes," in *21st Int Ent Dist Obj Comp Conf*, 2017, pp. 95–102.
- [13] N. Prat, I. Comyn-Wattiau, and J. Akoka, "A taxonomy of evaluation methods for information systems artifacts," *J Manag Inform Syst*, vol. 32, no. 3, pp. 229–267, 2015.
- [14] D. Rehak and M. Grasseova, "The ways of assessing the security of organization information systems through SWOT analysis," in *Crisis Management: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2014, pp. 162–184.
- [15] M. Reichert and B. Weber, *Enabling flexibility in process-aware information systems: challenges, methods, technologies*. Springer, 2012.
- [16] W. W. Royce, "Managing the development of large software systems: concepts and techniques," in *9th Int Conf Softw Eng*, 1987, pp. 328–338.
- [17] M. Schmitz, C. Stummer, and G. Michael, "Smart Automation as Enabler of Digitalization? A Review of RPA/AI Potential and Barriers to Its Realization," in *Future Telco, Management for Professionals*, P. Krüssel, Ed. Springer, 2019, pp. 349–358.
- [18] A. Stople, H. Steinsund, J. Iden, and B. Bygstad, "Lightweight IT and the IT function: experiences from robotic process automation in a Norwegian bank," in *Norsk konferanse for organisasjoners bruk av IT*, 2017.
- [19] R. Syed, S. Suriadi, M. Adams, W. Bandara, S. J. J. Leemans, C. Ouyang, A. H. M. ter Hofstede, I. van de Weerd, M. T. Wynn, and H. A. Reijers, "Robotic Process Automation: Contemporary Themes and Challenges," *Comput Indus*, vol. 115, 2020.
- [20] R. Vaculín, R. Hull, T. Heath, C. Cochran, A. Nigam, and P. Sukaviriya, "Declarative business artifact centric modeling of decision and knowledge intensive business processes," in *15th Int Ent Dist Obj Comp Conf*, 2011, pp. 151–160.
- [21] J. Wewerka, S. Dax, and M. Reichert, "A User Acceptance Model for Robotic Process Automation," in *24th Int Ent Dist Obj Comp Conf*, 2020, pp. 97–106.
- [22] J. Wewerka, C. Micus, and M. Reichert, "Seven Guidelines for Designing the User Interface in Robotic Process Automation," *submitted for publication*, 2021.
- [23] J. Wewerka and M. Reichert, "Towards Quantifying the Effects of Robotic Process Automation?" in *24th Int Ent Dist Obj Comp W*, 2020, pp. 11–19.
- [24] —, "Robotic Process Automation a Systematic Mapping Study and Classification Framework," *Ent Inform Syst (accepted for publication)*, 2021.
- [25] —, "Robotic Process Automation in the Automotive Industry - Lessons Learned from an Exploratory Case Study," in *15th Int Conf Res Chall Inform Sci*, 2021, pp. 1–17.
- [26] R. J. Wieringa, "A unified checklist for observational and experimental research in software engineering (version 1)," in *16th Int Conf Evaluat & Assess Soft Eng*, 2012, pp. 161–165.
- [27] —, *Design science methodology for information systems and software engineering*. Springer, 2014.
- [28] L. Willcocks and M. Lacity, "Robotic Process Automation: The Next Transformation Lever for Shared Services," *Outs Unit Work Res Pap S*, vol. 15, no. 7, pp. 1–35, 2015.
- [29] —, "Robotic Process Automation at Telefónica O2," *MIS Q Exec*, vol. 15, no. 1, pp. 21–35, 2016.
- [30] L. Willcocks, M. Lacity, and A. Craig, "The IT Function and Robotic Process Automation," *Outs Unit Work Res Pap S*, vol. 15, no. 5, pp. 1–39, 2015.

APPENDIX

<p>Evaluation:</p> <ol style="list-style-type: none"> Evaluate the task proposed to be automated with RPA. <ul style="list-style-type: none"> <input type="checkbox"/> The task is rule-based, i.e., the decision making is based on consistent rules without any exceptions. Therefore, the task does not have a lot of variants. <input type="checkbox"/> The task is highly repetitive and always requires the same actions. <input type="checkbox"/> The input and output of the task are digital, i.e., structured and electronically readable. <input type="checkbox"/> The task and used IT systems will not change in the near future. The roadmaps of the IT systems do not indicate new releases or changes. The only exception is a bot, which only runs once, e.g., to copy data from an old to a new system and its savings are reasonable high. <input type="checkbox"/> Involve subject matter experts to understand the task correctly. Evaluate the organizational environment. <ul style="list-style-type: none"> <input type="checkbox"/> The responsible management supports the automation. <input type="checkbox"/> The management is advertising for the planned RPA bot. <input type="checkbox"/> The responsibilities for maintenance and operation of the RPA bot are clarified. <input type="checkbox"/> Necessary working capacities for maintenance and operation are available. Evaluate the expected savings. <ul style="list-style-type: none"> <input type="checkbox"/> How many minutes can the automation of the task save per execution? <input type="checkbox"/> How many employees perform this task? <input type="checkbox"/> How often is the task executed per week? <input type="checkbox"/> Check whether the development of the RPA bot is worthwhile by converting the savings into FTE: (number of executions per week * number of employees * min. per execution) / weekly working hours. Evaluate the feasibility of the RPA project considering the aspects from above. 	<ul style="list-style-type: none"> <input type="checkbox"/> The end-users can always see the status of their task. <input type="checkbox"/> The end-user's attention is attained if the RPA bot finishes or fails. <p>Implementation:</p> <ol style="list-style-type: none"> Implement the RPA bot reliably. <ul style="list-style-type: none"> <input type="checkbox"/> If the same input is provided, the RPA bot always produces the same output. <input type="checkbox"/> If an error occurs, a message is provided. <input type="checkbox"/> The RPA bot does not lead to system errors in the IT systems it operates. Implement the RPA bot trustworthy. <ul style="list-style-type: none"> <input type="checkbox"/> The RPA bot works as designed. <input type="checkbox"/> The RPA bot is privacy-protecting and tamper-proof, especially regarding employee data. <input type="checkbox"/> The RPA bot is documented in detail in the code. Implement the RPA bot securely. <ul style="list-style-type: none"> <input type="checkbox"/> The RPA bot should only ever have the roles and access rights necessary to perform its tasks. <input type="checkbox"/> Use try-catch statements and other error handling extensively. Implement the RPA bot with the end-user. <ul style="list-style-type: none"> <input type="checkbox"/> Provide a minimal viable product (MVP) for the end-user. <input type="checkbox"/> Include the end-user in testing the RPA bot. Implement the RPA bot reusable. <ul style="list-style-type: none"> <input type="checkbox"/> Save reusable components of your RPA bot such that other developers can use them. Test the RPA bot. <ul style="list-style-type: none"> <input type="checkbox"/> Create a testing environment. <input type="checkbox"/> Select release criteria carefully with the end-users (user acceptance test). <input type="checkbox"/> Perform quality assurance tests. <p>Go Live:</p> <ol style="list-style-type: none"> Communicate the Go Live of your RPA bot. <ul style="list-style-type: none"> <input type="checkbox"/> Establish RPA opinion multipliers, i.e., colleagues advertising the RPA bot. <input type="checkbox"/> Include the management to communicate the advantages of the RPA bot. Facilitate the Go Live for the end-user. <ul style="list-style-type: none"> <input type="checkbox"/> Offer sessions for the end-users where they learn the usage of the RPA bot. <input type="checkbox"/> Provide user manuals for the end-users, which are easily understandable for non-IT users.
<p>Preparation:</p> <ol style="list-style-type: none"> Prepare the end-users. <ul style="list-style-type: none"> <input type="checkbox"/> You consider all end-users and think of end-users outside of your business unit. <input type="checkbox"/> You present the possibilities of RPA to the end-users and understand the end-users' expectations in the RPA bot. <input type="checkbox"/> You standardize the task such that it can be used by every end-user. Prepare the RPA bot. <ul style="list-style-type: none"> <input type="checkbox"/> If possible, reuse code from existing RPA bots. <input type="checkbox"/> In case of an error, the RPA bot provides an understandable error message that helps the end-user to fix the error. <input type="checkbox"/> In case of an error, the end-user can easily correct the mistake or finish the task manually. 	<p>Maintenance:</p> <ol style="list-style-type: none"> Maintain the RPA bot. <ul style="list-style-type: none"> <input type="checkbox"/> Monitor daily the usage of the RPA bot. <input type="checkbox"/> Improve the RPA bot continuously and regularly. <input type="checkbox"/> Change the RPA bot to account for changes in IT systems to ensure that the RPA bot always works. Maintain the end-user involved in the RPA project. <ul style="list-style-type: none"> <input type="checkbox"/> The end-user can monitor the bot via a dashboard. <input type="checkbox"/> The end-user can suggest improvements of the RPA bot.

Fig. 15. Final checklist to support knowledge workers in RPA projects.