The Journal of Systems & Software 202 (2023) 111697

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Checklists to support decision-making in regression testing☆

Nasir Mehmood Minhas [a,*], Jürgen Börstler [a], Kai Petersen [a,b]

[a] Department of Software Engineering, Blekinge Institute of Technology, Karlskrona, Sweden
[b] University of Applied Sciences Flensburg, Germany

ARTICLE INFO

Article history:
Received 18 September 2022
Received in revised form 19 February 2023
Accepted 1 April 2023
Available online 10 April 2023

Keywords:
Regression testing
Checklists
Test manager
Team readiness
Process improvement

ABSTRACT

Context: Practitioners working in large-scale software development face many challenges in regression testing activities. One of the reasons is the lack of a structured regression testing process. In this regard, checklists can help practitioners keep track of essential regression testing activities and add structure to the regression testing process to a certain extent.
Objective: This study aims to introduce regression testing checklists so test managers/teams can use them: (1) to assess whether test teams/members are ready to begin regression testing, and (2) to keep track of essential regression testing activities while planning and executing regression tests.
Method: We used interviews, workshops, and questionnaires to design, evolve, and evaluate regression testing checklists. In total, 25 practitioners from 12 companies participated in creating the checklist. Twenty-three of them participated in checklists evolution and evaluation.
Results: We identified activities practitioners consider significant while planning, performing, and analyzing regression testing. We designed regression testing checklists based on these activities to help practitioners make informed decisions during regression testing. With the help of practitioners, we evolved these checklists into two iterations. Finally, the practitioners provided feedback on the proposed checklists. All respondents think the proposed checklists are useful and customizable for their environments, and 80% think checklists cover aspects essential for regression testing.
Conclusion: The proposed regression testing checklists can be useful for test managers to assess their team/team members' readiness and decide when to start and stop regression testing. The checklists can be used to record the steps required while planning and executing regression testing. Further, these checklists can provide a basis for structuring the regression testing process in varying contexts.

1. Introduction

Practitioners working in large-scale software development face many challenges in their regression testing activities (Engström and Runeson, 2010). They have to choose between full regression testing (re-test all) and selective regression testing. Furthermore, in selective regression testing, test case selection is a complex decision-making activity (Engström and Runeson, 2010; Dalal et al., 2018). There are various factors that testers need to consider while selecting a subset of tests from a large test suite (Minhas et al., 2020). In selective regression testing, the goal is to maximize the coverage and fault detection ratio with a selected regression test set. While releasing a product, practitioners want to control the fault slippage and to be confident that they have tested enough, there are no critical faults in the release, and they have achieved the desired quality (Minhas et al., 2017, 2020; Jafrin et al., 2016; Nayak et al., 2016). Achieving these goals requires support from the organizational testing process. While analyzing various embedded systems and windows application-based projects, Kasoju et al. (2013) reported a lack of a structured testing process in organizations.

In our recent industrial studies (Minhas et al., 2017, 2020), we have identified the goals and challenges of regression testing in large-scale embedded software development companies. The identified challenges could be divided into two groups; process-related challenges and technical challenges. The primary cause of the identified challenges is the absence of a well-structured regression testing process. Instead of having a structured testing process for various testing activities, practitioners rely on expert judgment. (Engström and Runeson, 2010; Minhas et al., 2020).

However, evaluating or making a judgment without a structured mechanism symbolizes adhocism and may negatively impact the outcomes (Usman et al., 2018). There is a possibility that even experienced practitioners may overlook some essential aspects while making assessments and judgments (Usman et al., 2018). Similarly, it is hard for the new team members

☆ Editor: A. Bertolino.
* Corresponding author.
E-mail addresses: nasir.mehmood.minhas@bth.se (N.M. Minhas), jurgen.borstler@bth.se (J. Börstler), kai.petersen@bth.se (K. Petersen).

https://doi.org/10.1016/j.jss.2023.111697

to make good guesses with limited knowledge about the product, especially on what activities need to be considered before the start of regression testing and when they can decide to stop regression testing. To complement the expert judgments, checklist-based guidelines co-designed with expert practitioners could be a step toward potential solutions. In addition to existing strategies, regression testing checklists can help document and reuse best practices and address various regression testing challenges. Testing teams can adapt the proposed checklists to their own organizational needs. After adapting checklists to local needs, it will be easier for team members to understand organizational testing policies/activities quickly. They can benefit from checklists and become familiar with usual team practices. Without a checklist, it is highly likely that a new practitioner, for instance, can omit a necessary test or violate any team policy (Heroux and Willenbring, 2009). However, it is essential that checklists should be designed based on practitioners' needs and should be aligned with the teams' current practices. Checklists prepared without involving practitioners could be misused or ignored (Madaio et al., 2020).

This study aims to assist practitioners in improving regression testing by

1. identifying regression testing activities considered essential in practice and
2. introducing regression testing checklists based on essential activities.

We worked with 25 senior testing practitioners from 12 companies to identify essential regression testing activities and design regression testing checklists. Subsequently, of the same 25 practitioners, 23 representing ten companies participated in the evolution and evaluation of the proposed checklists. The proposed checklists are of two types (i) pre-regression testing checklists and (ii) post-regression testing checklists. Pre-regression testing checklists will help software test managers to assess the readiness of their team members and the team. Based on these assessments, they can decide whether to start or not regression testing activity. Further, post-regression testing checklists will help test teams track the post-regression testing activities. Based on the information collected from post-regression testing checklists, the test team can decide whether to stop or not regression testing activity. Moreover, test team members can use the proposed checklists to keep track of essential activities and take all necessary steps while performing regression testing. According to the classification of checklists provided in Hales et al. (2008), the proposed checklists could be categorized as a mnemonic tool. The primary purpose of such checklists is to provide an organizational framework for quickly recalling important information and current best practices.

The organization of the rest of the paper is as follows: Besides the background concepts, Section 2 briefly introduces checklists in software engineering practice and research. Along with the research questions, Section 3 provides a summary of methods opted in this study. Threats to the validity of this study and mitigation strategies to minimize the threats are discussed in Section 4. Section 5 presents the findings of this study, Section 6 provides the discussion on the process and outcomes of this study, and Section 7 concludes the study.

2. Background and related work

2.1. Regression testing

Regression testing is applied to a system under test after any change, including a defect fix or adding a new feature. The goals are to find defects and to obtain confidence about the quality of the systems under test. Regression testing can be performed in two ways (1) Re-test all – running all test cases in the regression suite and (2) Selective regression testing – running regression testing with a selected subset of test cases. The critical concern in this regard is the size of the test suite (Ammann and Offutt, 2016).

In large-scale software development, practitioners prefer to run regression tests with a selected scope. The primary challenge is determining the scope of regression testing (i.e., which tests to include in the regression test suite) (Engström and Runeson, 2010; Ammann and Offutt, 2016). Techniques used for SRT are test case selection, prioritization, and test suite minimization (Yoo and Harman, 2012; Lin, 2007).

2.2. Significance of checklists

Practitioners of various disciplines use checklists as a cognitive aid to ensure the correct completion of any task (Gawande, 2010).

> *"If the knowledge exists and is not applied correctly, it is difficult not to be infuriated."**(Gawande, 2010 The Checklist Manifesto)***

A checklist is a standardized tool that enlists the required process criteria for the practitioners performing a specific activity. It supports recording the presence or absence of the essential process tasks (Hales et al., 2008).

> *"Checklists seem able to defend anyone, even the experienced, against failure in many more tasks than we realized. They provide a cognitive net. They catch mental flaws inherent in us—flaws of memory, attention, and thoroughness."**(Gawande, 2010 The Checklist Manifesto)***

Two popular uses of checklists are, using checklists as mnemonic systems or as evaluation tools. The first is a reminder system to help practitioners avoid omitting essential tasks. It also ensures that practitioners follow the organizational framework and utilize best practices. Such checklists help minimize human error and improve overall performance. In contrast, the evaluative checklists can aid in the standardization of evaluation by providing assessment guidelines and ultimately improving the evaluation process's credibility (Hales et al., 2008).

Using a checklist to document any process is not a new concept. For example, in the aviation industry since the 1930s, it has been a standard operating procedure for pilots and other aviators to use checklists (Higgins and Boorman, 2016). Pilots use the checklists before, during, and after the flight (Hales and Pronovost, 2006). In medicine, checklists are used as a decision aid to identify a medical condition and decide on an appropriate course of treatment. In comparison, surgical checklists are recommended as a safety measure to reduce the margin of human error and any adverse effects during surgery (Chaparro et al., 2019).

Social and behavioral scientists are using self-reporting questionnaires as an assessment mechanism. Usually, such questionnaires include checklist items that enable the goal-based assessment of a phenomenon (Van de Schoot et al., 2012).

Software engineers use checklists in various tasks, including the audit of requirement/design specifications and code inspection (Usman et al., 2018). Checklists can help make a process repeatable, and practitioners can use various checklists in the software development life cycle. For instance, they can use release checklists to ensure that no essential steps are skipped. At the start, a checklist does not have to be exhaustive. If some items are missing, we can add the missing or new items later. Improvising a checklist is always helpful in adding future goals (Heroux and Willenbring, 2009). In the subsequent sections, we present related work regarding checklists in software engineering research and practice.

2.3. Use of checklists in software engineering

Perry (2007) provided generic checklists to aid software testing teams in different phases of their work. The author does not provide any checklist specific to regression testing. However, the checklists presented in this book can be taken as inspiration to introduce any testing-related checklists.

Based on their experience with a project, Heroux and Willenbring (2009) advocate using checklists to make processes repeatable. They suggest using checklists at various stages of a project. For instance, release checklists, developer checklists, and commit checklists. The authors stressed that checklists help practitioners remember essential but easily omitted steps. The checklists can help new team members to get familiar with the team practices. Heroux and Willenbring suggest that starting with simple checklists and improving them through iterations will help improve the related processes.

Brykczynski (1999) surveyed 117 software inspection checklists from 24 sources. The author suggests that checklists help the reviewers in a software inspection process by providing recommendations to find the defects. Brykczynski further recommends that checklists should be updated regularly. These should not be longer than a page and should be based on relevant items based on questions. The author classified the existing checklists according to their application type, including requirements, design, code, testing, documentation, and process.

Brito and Dias-Neto (2013) conducted empirical studies to evaluate their checklist-based technique (TestCheck). TestCheck is used to inspect the software testing artifacts. TestCheck consists of three separate checklists for assessing the test plan, test case, and test procedure. The authors aimed to evaluate and improve their technique through a series of evaluations in this study. They believe that their approach needs to undergo the evaluation process further, and there is a need to provide the tool support for the smooth transition of the technique to the industry.

Usman et al. (2018) proposed checklists to improve the software effort estimation. The authors revealed that expert judgment is the most common practice for effort estimation and could lead to wrong estimates without any process support. Usman et al. proposed a process to develop and evolve the estimation checklists. They started with understanding the current estimation process and identifying the relevant checklist factors from the existing literature. The authors developed and improved the checklist in different iterations based on the findings. They validated the proposed checklist in two steps (i.e., statically (a trial use) and dynamically (a real use). The authors claimed that checklists could increase practitioners' confidence in their estimates.

Madaio et al. (2020) designed a checklist to understand organizational challenges and opportunities around fairness in AI. The authors conducted semi-structured interviews and co-design workshops with 48 practitioners from 12 companies to understand the needs and concerns of practitioners and develop the AI fairness checklists. Madaio et al. concluded that checklists could help formalize the ad-hoc process and empower individual advocates. They were hopeful that their proposal could support the practitioners in addressing AI ethics issues. Although Madaio et al. (2020) did not evaluate their checklist, however, in the future, they are planning to conduct pilot studies with different teams.

Petersen et al. (2021) proposed a context checklist for industrial software engineering research and practice. They have listed three primary purposes of the proposed checklist, (i) to help record the experience in projects in an industrial setting, (ii) to help decide between the use of past decisions vs experience and knowledge, and (iii) to support researchers in deciding the contextual information to report in primary studies and information to extract in secondary studies. The authors evaluated the proposed checklists using interviews and questionnaires with the practitioners and researchers. Based on the feedback, Petersen et al. revised the checklists to overcome the deficiencies the practitioners and researchers identified.

Molléri et al. (2020) presented a checklist to support the survey research in software engineering. Using 12 methodological studies, the authors identified stages and recommended practices of the survey process. They used thematic analysis and vote-counting methods to aggregate knowledge from the existing selected studies. The authors evaluated the checklist by applying it to the existing surveys and analyzing the results. Later the authors collected the experts' feedback on the proposed checklist and improved the checklist.

Host and Runeson (2007) proposed two separate checklists to support the software engineering researchers and reviewers in conducting and reviewing case studies. The authors conducted a literature survey to identify the existing checklists in the first stage. They merged the checklists they found in the literature into a single checklist and classified the items according to different case study phases. Later, the authors reduced the size of the checklist by grouping similar items. After validating the checklist with the Ph.D. students, the authors updated it to accommodate the validation feedback.

Kitchenham et al. (2010) merged two checklists into a single checklist to evaluate the quality of software engineering experiments. The authors constructed the checklist using the findings of two studies (Kitchenham et al., 2009; Dybå and Dingsøyr, 2008). The authors performed a two-step validation of their proposed checklist by applying the checklist criteria to the selected papers from human-centric software engineering experiments.

2.4. Summary

From the review of the related literature, we learned that the use of checklists in software engineering research and practice is evident. The checklists in software engineering research are used to assess the quality of adopted empirical research and provide guidelines to conduct empirical studies systematically. Software engineering professionals use checklists to assess activities, including effort estimation, code reviews, and testing. To the best of our knowledge, no checklists are available explicitly designed for regression testing. However, to start with a checklist, we took inspiration from the testing checklists presented in Perry (2007), mainly for structuring the checklist items for the corresponding regression testing activities.

3. Methodology

The primary objective of this study is to support practitioners in structuring the regression testing process by introducing regression testing checklists. We followed an iterative approach to design, evolve, and evaluate the regression testing checklists. We intended to find the answer to the following **research questions** to fulfill the study's objective.

- RQ1 What activities do practitioners consider while planning, performing, and assessing regression testing?
- RQ2 What checklists and checklist items can be helpful for practitioners while planning, performing, and assessing regression testing?
- RQ3 What is the perspective of practitioners about the proposed checklists?

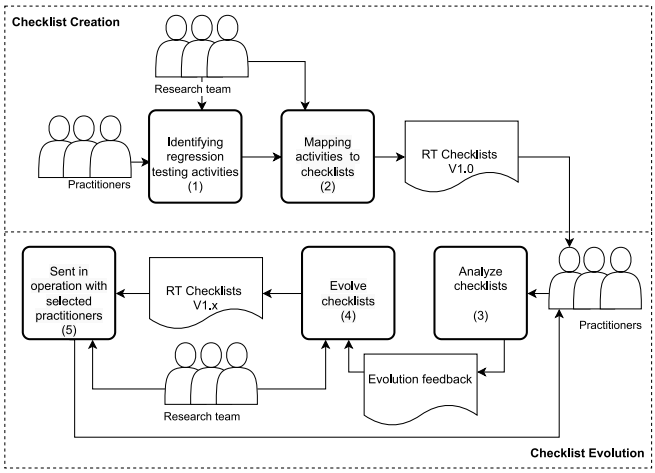

Fig. 1. Overview of the approach used to design and evolve the regression testing (RT) checklists.

3.1. Research approach

Fig. 1 presents the overview of the approach we followed to design and evolve the regression testing checklists. We involved the practitioners in our process, from checklists' activities identification to checklists' verification. We conducted individual and group interviews with senior testing practitioners to identify the activities that should be considered during the overall regression testing process. Based on the input from practitioners, we mapped the regression testing activities to the elements required for decision-making before and after the regression testing process (see Fig. 3). This mapping provided us with a basis for the regression testing checklists. We handed over the initial draft of checklists to the practitioners and asked them to assess each item to see if they were relevant. Based on their feedback, we made improvements to the checklists. Finally, the industry practitioners evaluated the proposed checklists. They assessed checklists concerning comprehensiveness, usefulness, and relevance to their context.

3.2. Selection of participants

We followed a snowball sampling approach (Kitchenham and Pfleeger, 2002b) to select the participants for our study. Since we were interested in recruiting senior testing practitioners, we imposed a constraint that participating practitioners must have five years or more of experience in software testing.

We contacted seven senior testing practitioners from three large Swedish companies, two of whom had already participated in our previous studies on regression testing (Minhas et al., 2017, 2020; bin Ali et al., 2019). We received responses from five practitioners. We conducted introductory workshops with them to present the study idea and finalize the operational aspects of the study. Also, we requested them to provide us with the contacts of senior testing practitioners from their companies or contact network. All five practitioners consented to participate in the study and introduced us to a few more practitioners who fulfilled the criteria of the required experience. Along with contacting the practitioners from the companies we had already worked with, we also sent various requests through LinkedIn. We conducted introductory workshops with willing practitioners, one for each company, regardless of the number of participants. We continued with this approach and stopped after getting the consent of 25 senior testing practitioners from 12 companies.

3.3. Data collection

We used multiple data collection methods to introduce our research goals to the prospective participants, collect data on regression testing practice, evolve checklists, and verify the checklists.

3.3.1. Introductory workshops

Our study required to involve senior testing practitioners in all phases, and we needed a long-term commitment from them. We conducted 12 online workshops to introduce our research idea of working on regression testing checklists to prospective participants. The example of the content used in these workshops is presented as Appendix B in the supplementary data. Apart from the short introduction, the workshops were mainly Q&A-based discussions. These workshops aimed to ensure senior testing practitioners' informed and consented participation.

3.3.2. Interviews (checklists creation)

In the second phase, we conducted seven individual and five group interviews to collect data on regression testing practice. The average duration for individual interviews was 60 min, and for group interviews, it was 75 min. For designing and conducting the interviews, we followed the guidelines of Runeson and Höst (2009). We used semi-structured interviews, as these provide flexibility in improvising and ordering the interview questions by following the discussions (Runeson and Höst, 2009). The interview questions were open-ended, and we improvised them, given the participants' context. During these interviews, we asked questions to understand the regression testing practice in the companies, for example, what regression testing activities are considered essential and what stopping criteria the testing practitioners use. The detailed interview questionnaire is available as Appendix C in the supplementary data. To ensure the quality of the interview instrument, we underwent expert reviews. A senior practitioner (reviewer 1) with ten years of research and development experience in software testing and an academic researcher (reviewer 2) with sixteen years experience in software engineering research evaluated the interview instrument. Reviewer 1 agreed with the instrument and did not suggest adding or updating anything in the instrument. However, reviewer 2, who also has experience developing checklists, suggested some changes, including adding a question about the product domain and changing the phrasing of Question 10. The interview instrument in Appendix C represents the version after incorporating reviewers' feedback.

3.3.3. Workshops (checklists evolution)

Workshops provide a practical approach to designing, evaluating, or co-creating any artifact of interest (Thoring et al., 2020). We conducted online workshops for *checklists evolution*. We designed an evolution tool to evolve the checklists generated based on interview results. Along with the initial draft of checklists, we shared the evolution tool with the practitioners who participated in the study's second phase (i.e., interviews). The first author participated as an observer in five workshops conducted with the participants of companies C1, C2, C6, C7, and C8. The participants discussed and analyzed every single point before making any judgment. The participants from C3, C4, C9, C11, and C12 preferred to run evolution workshops on their own, as they believed that doing this would help them avoid any influence of the researchers, hence avoiding any biases in their judgments. Two practitioners from C5 and C10 could not participate in the evolution phase. The participants reflected on each item on the checklist, and for every item, they rated the checklist items based on the question given below.

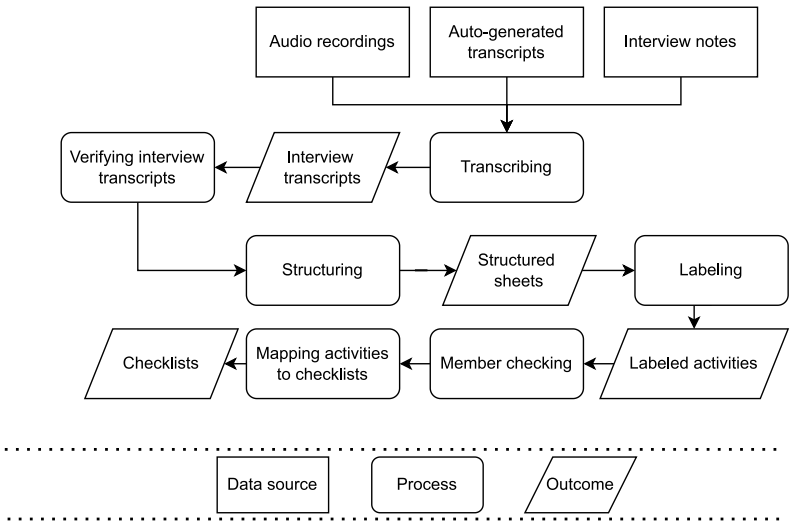

Fig. 2. Data analysis steps.

- Is the checklist item relevant? Yes, no, do not know

If practitioners considered the checklist item relevant, they chose "yes". If they did not find it relevant, the option chosen would be "no". If they were indecisive, they reported it as "Do not know". We asked the participants to suggest additional items if they thought we had missed any. In case of suggesting a new item, the practitioners needed to specify the checklist for which the item was proposed, and they could also provide additional comments to motivate their suggestion. The tool for evolving checklists is described as Appendix D in the supplementary data.

3.3.4. Survey (checklists evaluation)

After the checklists were finalized, we requested the practitioners to provide feedback based on the checklists' trial run and team discussion. We used an online survey for *checklists evaluation*. To design the survey questionnaire, we followed the guidelines by (Kitchenham and Pfleeger, 2002a). We created an evaluation questionnaire using Google forms and sent it to all the participants. The evaluation questionnaire was created on the following parameters:

- comprehensiveness,
- usefulness,
- customizability, and
- adoptability

We also added a question to ask if the participants' companies are willing to use the checklists. The complete evaluation questionnaire is provided as Appendix E in the supplementary data.

3.4. Data analysis

Detailed analysis was required for the qualitative data, whereas summaries and graphs were required for the data collected during the checklist evolution workshops and survey (i.e., evaluation of checklists). We followed thematic analysis to analyze the qualitative data and took inspiration from the methods described in Lacey and Luff (2001), Cruzes and Dyba (2011). Fig. 2 presents the detail of the steps carried out for the data analysis, whereas the steps are outlined in the following subsections.

3.4.1. Transcribing

All interviews were conducted using online meeting tools and recorded with the consent of all participants. During the interviews, we enabled the automated transcription facility provided by the online meeting tool, and one note-taker took notes. We finalized the transcripts using auto-generated transcripts, interview notes, and audio recordings. We ensured to transcribe the participants' words verbatim to avoid bias and misinterpretation. On average, we invested four hours in transcribing an interview. Later, with the help of an independent volunteer, the first author verified the transcripts by comparing them with sources to ensure we did not miss any vital information.

3.4.2. Structuring

Lacey (Lacey and Luff, 2001) suggests organizing the data into easily retrievable sections after finalizing the transcripts. We converted the transcripts into a structured excel sheet (see Table 1). We extracted the information under column headings which were derived from interview questions. However, finding desired information against the relevant questions was not straightforward. We often needed to scan the answers to multiple questions to find the relevant information. This step helped us familiarize ourselves with the data. We used color coding to differentiate between emerging themes and to add more clarity.

3.4.3. Labeling

We have reported participants' original statements regarding the state of regression testing practice and essential activities as Appendix A (Tables A.13 & A.14) in the supplementary data. After having the interview results in a structured form with appropriate color codes, the next step was to have appropriate labels for similar themes. We used labels for the activities considered essential in the companies before and after regression testing. We followed the philosophy of axial coding while assigning the labels to activities (Böhm, 2004) and assigned appropriate labels by grouping similar statements. For example, we grouped the following three statements: i. selecting a smaller but effective subset of test cases, ii. selecting the right test cases, and iii. good knowledge of test cases helps in selecting the right test cases. We labeled the mentioned statements as *"selection of right test cases"*. In assigning a label to a group of similar statements, we ensured that we used a label that reflected the perspective of the practitioners involved. We continued this exercise until we grouped all similar statements and labeled them appropriately.

Table 1
Structure of transcription sheet — RT (regression testing).

Heading	Description
CID	ID for the Participants' companies.
State of RT	How regression testing is performed in the case companies?
Significance of RT	How significant is RT for the case companies?
Frequency	Frequency of releases.
Significant activities	RT activities that are significant for the participating companies.
Essential aspects before RT	The aspects that practitioners suggest to consider before the start of RT.
Essential aspects after RT	The aspects that practitioners suggest to consider after RT.
RT goals	The goals that practitioners set for RT.
RT CL	If the participants already using any checklists for RT?
CL Usefulness	The perception of practitioners about the usefulness of prospective RT checklists.

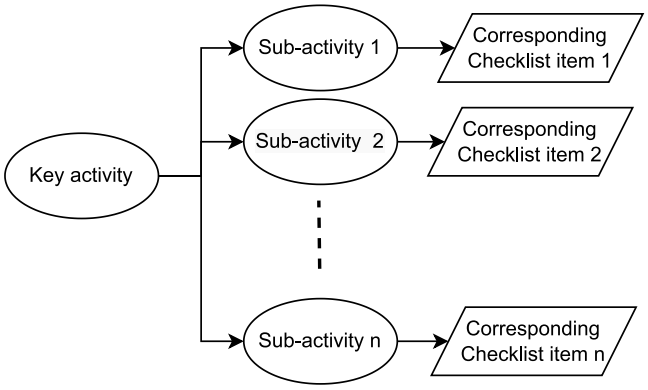

Fig. 3. Mapping regression testing activities to checklist items.

3.4.4. Member checking

Along with the participants' original statements regarding the state of regression testing practice and essential activities, we sent the labeled activities to the participants. We asked them to verify if we had interpreted their perspectives correctly. However, we did not receive any corrections from them. This was because we did not deviate from participants' statements while defining the labels for the activities.

3.4.5. Mapping

After identifying regression testing activities considered essential in the companies, the next step was to create checklist items based on these activities. We divided activities into sub-activities (where possible) and then transformed each activity into a checklist item. While creating the checklist items (questions), we took inspiration from the structure of software testing checklists presented in Perry (2007). Fig. 3 presents how we mapped the regression testing activities to the subactivities and then transformed them into the checklist items.

For example, the regression testing activity labeled in the previous step **"selection of right test cases"** was divided into two sub-activities (1) Identifying test cases related to changes and (2) Identifying test cases related to impacted modules. Later these activities were mapped to the following checklist items.

- Have the test cases associated with changed parts been identified?
- Have the test cases associated with the impacted module been identified?

4. Validity threats

This study's results are based on industry practitioners' experiences and perceptions. There are various aspects related to this study that can pose threats to its validity. In the following, we have discussed potential validity threats and the strategies we opted to mitigate these threats. In our discussion of threats to validity, we follow the guidelines by Runeson and Höst (2009) and Wohlin et al. (2012).

Construct validity. This aspect of validity could be associated with the choice of treatment for the study and its expected outcomes. In our study, we can link it to creating the data collection instruments. We used well-known guidelines and followed the established procedures to minimize this threat to validity. While designing the data collection instruments, we followed the guidelines by Runeson and Höst (2009) for the interview questionnaire, and Kitchenham and Pfleeger (2002a) for the design of the survey questionnaire. To avoid inconsistency and bias, we involved a senior researcher and a practitioner to review the instruments. Based on their feedback, we augmented our data collection instruments. To evaluate the proposed checklists, we used an online questionnaire. We kept this questionnaire simple to facilitate our participants. A simple questionnaire could be a threat to construct validity as it can lead to false positive feedback. However, in our study, this threat was minimum because we involved senior practitioners with experience ranging from five to twenty-three years. Most of them have an interest in the improvement of regression testing practice. Therefore, respondents were expected to provide an objective assessment of the checklists.

Internal validity. Internal validity focuses on whether we can be sure that the treatment caused the outcome. In our study, the primary aspect that could impact the outcome was identifying regression testing activities. We involved participants from different organizations to ensure that identified activities were correct and complete. Hence source triangulation is ensured. Similarly, to ensure the selection of relevant participants, we selected them using the snowball sampling technique based on their experience and interest in regression testing. Further, the study participants reviewed and validated all the results generated in this study to ensure that the interpretation of results was free of any misinterpretation or biases. To ensure that checklist items are correct and fulfill the needs of practitioners, we involved the participants in evolving the checklists after creating the initial version of the checklists.

External validity. The external validity threats refer to the concept of generalization of the results. Although, we did not claim the generalizability of our checklists. However, the similarity in views of practitioners from 12 different companies working on

Table 2
List of practitioners who participated in checklist design and evolution.

Participant ID	Role	Experience in years
P1	Manager SQA	15
P2	Test Lead	10
P3	Test Lead	11
P4	Senior Test Lead	12
P5	Test Lead	10
P6	Senior Manager QA	17
P7	Test Architect	8
P8	Test Lead	23
P9	Tech Lead	16
P10	Senior Test Engineer	20
P11	Senior SQA Engineer	11
P12	SQA Engineer	6
P13	Senior SQA Engineer	9
P14	Test Lead	11
P15	Senior SQA Engineer	7
P16	Test Manager	9
P17	Test Engineer	8
P18	Test Engineer	6
P19	Test Manager	15
P20	Test Engineer	5
P21	Test Engineer	6
P22	Test Lead	12
P23	QA Lead	22
P24	Head SW QA	20
P25	Test Engineer	15

diversified domains indicates the possible generalizability of the proposed checklists. We have provided contextual information of the participants' companies. We have also provided the detail of data collection instruments (See Appendices C, D, & E in the supplementary data). This may help generalize the context and replicate the study in the future.

Reliability. This aspect concerns the extent to which the data and analysis depend on the specific researchers. The results are reliable if they are free of biases, and independent researchers can reproduce them using similar methods. We took various measures to minimize the threats to the reliability of the study. For example, concerning data collection, with the prior consent of the participants, we recorded all interviews. We generated the structured transcription sheets using recordings, interview transcripts, and notes taken during the interviews. We have explained all aspects of data collection, analysis, and reporting in Section 3.

5. Results

Twenty-five practitioners from 12 companies participated during the initial phases of the study, and 23 practitioners from 10 companies participated in the evolution and evaluation phases. Our goal was to suggest checklists that could be useful and fit in the industry context. We designed the checklists in an iterative process and involved the industry participants in all phases (i.e., checklist design, evolution, and evaluation). Our findings represent the perspective of senior testing practitioners. The participants' experience ranges from five to twenty-three years, and their average experience is twelve years. The participants' organizational roles are test engineer, senior test engineer, test lead, QA manager, senior SQA manager, test architect, and head QA unit (see Table 2). Further, most participants work in large-scale environments. The size of a company can be classified based on the number of professionals working in it and the volume of projects (Minhas and Iqbal, 2011). A large company will have more than 250 practitioners working (Lindgren and Münch, 2016). In our sample, only two companies, C6 and C8, have less than 250 practitioners working. The participants' companies use agile methodologies, including scrum, CI/CD, and DevOps. The domains represented in our study are financial, banking, healthcare, transport, surveillance and security, telecommunication, AI solutions, and security systems. For more detail on contextual information of the companies represented by the participants, please see Table 3.

5.1. State of regression testing practice in companies

The detail of how regression testing (RT) is performed in the participants' companies is provided as Appendix A (Table A.13) in the supplementary data. Most participating practitioners consider regression testing to be an indispensable activity. In one of the companies, practitioners only perform regression testing for significant changes and use exploratory testing for minor changes. The practitioners at most companies use exploratory testing after regression testing to ensure that all risk areas are functioning correctly. The practitioners at one company use smoke testing before regression testing to check if the build is stable. The frequency of regression testing varies between companies, and it mainly depends on the domain and criticality of the product/module under test.

Testing practitioners set regression testing goals, most of them (10 of 12) do it informally, and only a few have a defined methodology for setting and estimating goals. Participants from the companies stated that expert judgment is a primary driver for most decisions. For example, when to stop regression testing is based on expert judgment, and test managers and team leads make this decision based on their experience and knowledge. Information considered to stop regression testing includes execution of the planned regression suite, pass/fail ratio, comparison of pass rate to the defined threshold (e.g., if pass rate is 90% or above), and severity of identified bugs.

The companies are transitioning from manual and partially automated to fully automated regression testing, and some companies have introduced CI/CD and DevOps pipelines. The scope of regression testing is defined based on the changes and their impact. Regression testing is performed with a selected set of test cases whenever a change occurs (adding a new feature or fixing a bug). Near the release, the practitioners prefer to run the complete regression suites, which are automated in most cases. A different kind of test case selection is used in one of the companies, as a senior manager revealed that their regression suites are enormous, and running all tests is costly. To cope with the cost, they are experimenting with running all tests for the most commonly used features instead of running all tests in the regression suite — the participant suggested a slogan for it "running all tests that matter".

Concerning the regression test plan, the majority consider it as a part of the test plan, and they do not have a separate regression test plan. The regression plan is part of the sprint planning meeting in three companies, and during every sprint, they make essential decisions about regression testing. For example, what are the new fixes (tickets), what do they have to test, and how much should they test?

5.2. Regression testing activities ***(RQ1)***

The results of RQ1 provided us with the basis for creating the regression testing checklists. Here, the intent was to investigate the activities involved in planning, performing, and evaluating regression testing. We conducted 12 interviews with 25 practitioners and asked them to highlight the activities they believe are essential to regression testing. Table 4 outlines the activities considered essential for regression testing in the participating companies. Practitioners consider selecting the right test cases (e.g., more coverage with fewer test cases) as a key to their

Table 3
Contextual information of the companies represented by the participants (size classification: *"small < 50, large > 250"* (Lindgren and Münch, 2016)).

CID	Employees	Size	Leading participant	Product domain	Approach	Test team size	Participants
C1	3000	Large	Manager IT/SQA	Financial	Agile	18	P1, P2, P3
C2	500	Large	Senior QA Lead	Financial	Agile, Scrum	18	P4, P5, P6
C3	8000	Large	Test lead	Transport	Agile, DevOps	20	P7
C4	4000	Large	Tech lead	Surveillance/Security	Agile	17	P8
C5	10000	Large	Senior Tester	Charging System	Agile, DevOPs	50	P10
C6	150	Medium	Senior SQA Engineer	Healthcare	Agile, Scrum	10	P11, P12, P13, P14
C7	5000	Large	Senior SQA Engineer	Telecom	Agile, CI-CD Pipelines	10	P15, P16, P17, P18
C8	200	Medium	Test Manager	Telecom	Agile, DevOps	10	P19, P20, P21
C9	3000	Large	Test Lead	AI Solutions	Agile	75	P22
C10	20000	Large	Test Manager	Security Systems	Agile	10	P23
C11	13000	Large	Head SQA	Banking	Agile	25	P24
C12	1000	Large	Test Engineer	Hardware & Software	Agile/Scrum	12	P25

Table 4
Activities considered essential for regression testing (RT) in the companies.

A#	Activities before RT	Companies
1.	Acquiring domain knowledge	C1, C2, C5, C6, C7, C8, C9, C11, C12
2.	Ensuring communication of changes	C1, C3, C5, C6, C12
3.	Knowing new features/Changes	C3, C4, C5, C6, C7, C8, C9,C10, C12
4.	Ensure that changes are freezed	C3, C6, C10, C11, C12
5.	Ensure changes have been tested	C2, C4, C7, C9, C10, C12
7.	Identifying impact of changes	C1, C2, C4, C5, C6, C7, C9, C10, C12
8.	RT scope is decided	C1, C2, C4, C8, C12
9.	Selection of right test cases	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12
10.	Having required test environment	C10, C11
11.	Organizing test data	C2, C6, C7, C8, C10, C11, C12
12.	Defining regression testing goals	C1, C2, C3, C4, C12
13.	Making regression test plan	C7, C8, C10
14.	Assigning responsibilities	C2,C8
15.	Test suite maintenance	C1, C6, C7, C9, C11
A#	Activities after RT	Companies
1.	Having the planned regression test suite executed	C1, C3, C5, C6, C8, C9, C10, C12
2.	Creating test reports	C2, C6, C7, C10, C11, C12
3.	Analyzing test results	C1, C2, C3, C5, C6, C7, C8, C9, C10, C11, C12
4.	Assessing goals achievement	C1, C3, C4, C5, C6, C7, C9, C11
5.	Assessing ratio of pass vs fail	C1, C4, C5, C6, C7, C9, C11, C12
6.	Ensure pass percentage is above threshold	C2, C5, C6, C7, C9, C11, C12
7.	Critical bugs have been identified and resolved	C1, C2, C3, C4, C8, C10, C11, C12

success. This could only be possible if they have good domain knowledge, understand the system specifications, and know the changes and their impact. To understand the impact of changes, the practitioners need to know the dependencies among the modules/subsystems. All changes must have been tested sufficiently, and all tickets/changes must be checked in (code freeze) before the start of regression testing. Availability of the required test environment and data is also essential to start regression testing. In the end, the practitioners ensure to run the planned regression tests completely. They generate test reports, analyze the results, and decide subsequent actions based on the test results. They look at the pass vs. failure test cases and decide to release the product if the pass percentage is above the defined threshold. They can decide to release the product with the medium severity bugs, and fixing bugs will be part of the next release. However, in case of severe faults, they must stop the release. In many cases, the teams decide on the goals before the start of regression testing, and they assess the achievement of their goals after finishing the regression testing activity. We found similar activities in some related studies to some of the regression testing activities identified in our study. Table 5 presents the regression testing activities identified in our study and those also listed in related studies. A complete summary of regression testing activities considered essential in the participants' companies is presented as Appendix A (Table A.14) in the supplementary data.

5.3. Regression testing checklists ***(RQ2)***

During the interviews, we asked questions about regression testing activities and regression testing checklists. These questions helped us understand the practitioners' perspective on the regression testing checklist and identify the activities required for regression testing. We mapped regression testing activities to individual checklists based on these findings. Most participants were convinced about the usefulness of regression testing checklists, provided these checklists cover essential aspects only. They pointed out that checklists can help add structure to regression testing practice, and practitioners will not skip any essential step while performing regression testing.

Table 5
Regression testing activities identified from selected studies.

Activities identified in literature	Ref	Identified in our study
Understanding requirements specifications	Harrold and Orso (2008), Engström and Runeson (2010), Minhas et al. (2020), bin Ali et al. (2019)	Acquiring domain knowledge
Understanding Changes	Harrold and Orso (2008), Yoo and Harman (2012), Engström and Runeson (2010), Minhas et al. (2020), bin Ali et al. (2019)	Knowing new feature/changes
Identification of affected areas (Dependencies)	Engström and Runeson (2010), bin Ali et al. (2019), Minhas et al. (2020)	Identifying impact of changes
Modules to be tested	Engström and Runeson (2010)	Deciding RT scope
Selection of right test cases	Harrold and Orso (2008), Engström and Runeson (2010), Minhas et al. (2020)	Selection of right test cases
Collaborate with developers	Minhas et al. (2020)	Communicating changes
Preparing test reports	Engström and Runeson (2010), Minhas et al. (2020), bin Ali et al. (2019)	Creating test reports
Analyzing test results	Engström and Runeson (2010), bin Ali et al. (2019), Minhas et al. (2020)	Analyzing test results

> *Checklists can help in adding formalism to practice. Although we have a well-managed plan, we still miss things. If we have a small checklist that can add value.* ***(Senior SQA Engineer)***
>
> *A guideline could be an asset for any tester that can help him do essential things before regression testing. It will reduce the impact of team members leaving and new members being added.* ***(Test Engineer)***
>
> *It will help streamline the practice. However, it should be a short checklist not to hinder the job. We would be interested in using the proposed checklists.* ***(Head QA unit)***

Some form of regression testing checklists are in place in companies (e.g., C3, C4). However, these checklists are application-specific.

> *We are already using the checklists, but our checklists are product-specific. During my experience, I have worked with waterfall, agile, and now DevOps. In waterfall, we had too many checklists, but in our current environment, we have only essential checklists. I think checklists are helpful since they help practitioners not forget to do any essential activity.* ***(Test Lead)***

The participants from these companies voiced the usefulness of generic checklists that can guide practitioners to stick to the plan and not miss any essential steps. Some of the participants highlighted that they do not use any formal checklists. However, they informally follow the lists of essential items. For example, at C2, senior test leaders assess the readiness of their team members through informal chats.

> *We are doing something similar to checklists informally, but we are not using any predefined checklists. We assign regression testing of different modules to the practitioners based on their relevant knowledge of the modules. So informally, we gauge the readiness of the team members. However, we do it using our first-hand knowledge. Being an experienced manager, I know the skills of my team member. When we induct a new member, we provide a chance to get on board and help him gain domain knowledge. We check the readiness through informal chats.* ***(Senior Manager QA)***

Some participants were reluctant to give their opinions before seeing the actual checklists. For example:

> *I would like to see what checklists emerge, and then I will decide about their usefulness. If it is helpful for my context, then it is useful. Generally, I agree that checklists are a useful thing for any environment.* ***(Senior Tester)***

While responding to our question about the types of prospective checklists, most practitioners suggested checklists to assist practitioners before and after regression testing (i.e., pre-regression testing and post-regression testing checklists). There was a divided opinion about the definition of pre-regression testing checklists. The majority of participants suggested having two checklists (one for individual testers and one for team activities), and some suggested using a combined checklist. Table 6 summarizes practitioners' opinions about checklist types.

5.3.1. RT checklists creation and evolution

Checklists creation. Based on the input from the participants about the checklist types presented in Table 6, we decided to opt for the following three checklists.

1. Checklist to track the activities before regression testing (Individual)
2. Checklist to track the activities before regression testing (Team)
3. Checklist to track the activities after regression testing (Exit criteria)

To decide the checklists' items, we considered the regression testing activities identified by practitioners (See Table 4). For each activity, we created relevant checklist items and mapped the individual items to the individual checklists. As a result of this exercise, we created the checklists $CL_{1.0}$, $CL_{2.0}$, & $CL_{3.0}$ presented in Tables 7, 8, & 9. To check the team members' readiness, a test manager can ask them to fill out the checklist ($CL_{1.0}$) provided in Table 7. Later, based on $CL_{1.0}$ results, a test manager and team members can fill out the checklist ($CL_{2.0}$) presented in Table 8 to assess the team's readiness. Finally, while stopping regression tests, a test manager and team members can fill in the checklist ($CL_{3.0}$) presented in Table 9. In every checklist table, we have provided two additional columns "status" and "comments". Using the status column, the stakeholders can report the status

Table 6
Types of regression testing (RT) checklists suggested by participants.

Checklist type	Suggested by
Checklists to track the activities before regression testing (Individual)	C1, C4, C6, C7, C9, C10, C12
Checklists to track the activities before regression testing (Team)	C1, C4, C6, C7, C9, C10, C12
Checklists to track the activities before regression testing (Combined)	C2, C3, C8, C11
Checklists to track the activities after regression testing (Exit criteria)	C1, C2, C3, C4, C7, C8, C11, C12

Table 7
Checklist to access the readiness of testers to be filled by test team members — $CL_{1.0}$.

CLI	Checklist item	Status	Comments
1	Are you aware of the team's regression testing goals?		
2	Do you have essential knowledge of system specifications?		
3	Are you aware of dependencies among the subsystems?		
4	Are you aware of new changes in the system?		
5	Have you analyzed the impact of changes on the unchanged parts of the system?		
6	Are you confident of performing regression testing on your own?		
7	Have you been trained for the tools used for regression testing within the team/organization?		
8	Are you aware of the criticality of the subsystems to be tested?		
9	Do you have access to test data?		

Table 8
Checklist to determine the team's readiness to be filled by test manager — $CL_{2.0}$.

CLI	Checklist item	Status	Comments
1	Have the regression testing goals been defined?		
2	Are the test team members aware of system specifications?		
3	All the changes been checked in?		
4	Have the changes been communicated to the test team?		
5	Have the changes been tested in isolation?		
6	Has the change impact been determined?		
7	Is the regression test suite up to date?		
8	Have the test cases associated with changed parts been identified?		
9	Have the test cases associated with the impacted module been identified?		
10	Has the regression testing scope been determined?		
11	Has the regression testing been incorporated into the test plan?		
12	Has the regression test plan been developed?		
13	Are the required resources available?		
14	Has the decision been taken between manual vs. automated regression testing?		
15	Have clear responsibilities assigned to team members?		
16	Did the testing team agree to start regression testing?		

Table 9
Checklist to determine exit criteria of regression testing to be filled by test manager together with team members — $CL_{3.0}$.

CLI	Checklist item	Status	Comments
1	Have the regression testing test suites been executed completely?		
2	Has the pass rate of regression test suites reached the threshold?		
3	Have all severe /critical defects been resolved?		
4	Have all medium severity defects been closed?		
5	Have all metrics been collected?		
6	Have defined regression testing goals been achieved?		
7	Do the test team members agree to test closure?		

concerning the checklist item, and in the comments column, they can provide the details related to the reported status. For example, for the checklist item *"Are you aware of dependencies among the subsystems?"*, the concerned stakeholder can fill in *"Yes I am aware"*, *"Yes, but not 100%"*, *"No, it is not applicable"*. In the comments column, the stakeholders can further explain the status. If the reported status is yes, then the stakeholders have to reflect on the system dependencies, and if the status is "yes but not 100%", they will report the missing aspects.

Checklists evolution. During the checklists' evolution workshops, the participants were asked to give their opinion on the relevance of checklists and checklist items. They were asked to choose "Yes" for relevant, "No" otherwise, and "Do Not Know" if they were

Table 10
Evolution of checklist to know the readiness of testers to be filled by test team members — $CL_{1.1}$.

CLI	Is the checklist item relevant?	Yes	No	Do not know
1	Are you aware of the team's regression testing goals?	9	1	0
2	Do you have essential knowledge of system specifications?	9	0	1
3	Are you aware of dependencies among the subsystems?	10	0	0
4	Are you aware of new changes in the system?	10	0	0
5	Have you analyzed the impact of changes on the unchanged parts of the system?	8	1	0
6	Are you confident performing regression testing independently?	3	4	3
7	Have you been trained for the tools used for regression testing within the team/organization?	10	0	0
8	Are you aware of the criticality of the subsystems to be tested?	9	0	1
9	Do you have access to test data?	8	0	2

Table 11
Evolution of checklist to know the readiness of test team to be filled by test manager — $CL_{2.1}$.

CLI	Is the checklist item relevant?	Yes	No	Do not know
1	Have the regression testing goals been defined?	9	1	0
2	Are the test team members aware of system specifications?	9	0	1
3	All the changes have been checked in?	8	1	1
4	Have the changes been communicated to the test team?	9	0	1
5	Have the changes been tested in isolation?	7	1	2
6	Has the change impact been determined?	8	0	2
7	Is the regression test suite up to date?	9	0	1
8	Have the test cases associated with changed parts been identified?	10	0	0
9	Have the test cases associated with the impacted module been identified?	9	0	1
10	Has the regression testing scope been determined?	10	0	0
11	Has the regression testing been incorporated into the test plan?	5	3	2
12	Has the regression test plan been developed?	6	2	2
13	Are the required resources available?	9	0	1
14	Has the decision been taken between manual vs. automated regression testing?	8	1	1
15	Have clear responsibilities been assigned to team members?	9	1	0
16	Did the testing team agree to start regression testing?	6	2	2

Table 12
Evolution of checklist to determine exit criteria of regression testing (RT) to be filled by test manager together with team members — $CL_{3.1}$.

CLI	Is the checklist item relevant?	Yes	No	Do not know
1	Have the regression testing test suites been executed completely?	10	0	0
2	Has the pass rate of regression test suites reached the threshold?	9	0	1
3	Have all severe /critical defects been resolved?	10	0	0
4	Have all medium severity defects been closed?	7	1	2
5	Have all metrics been collected?	6	0	4
6	Have defined regression testing goals been achieved?	10	0	0
7	Do the test team members agree to test closure??	9	0	1

undecided about a checklist or checklist item. We provided additional space in the evolution forms for suggestions and reflections from the participants. We shared the initial draft of checklists and evolution forms with the practitioners who participated in the study's first phase. The practitioners evaluated checklist items and provided their feedback. The practitioners who participated as a group in the study's earlier phases provided us with their feedback as a group.

Tables 10, 11, and 12 present the summary of feedback from the study participants. Concerning the relevance of checklists, participants agreed that all checklists are relevant. However, for some of the checklist items, a few participants opted for "No", and "Do not know". We used red, gray, and cyan colors for the checklist items that received fewer recommendations. Red for the checklist items that received less than 50% recommendations, gray for the items that received 50 to 60% recommendations, and cyan for the items that received more than 60% but less than 80% recommendations.

Suggestions by respondents. Test team lead of C4 has reflected on some of the items included in the checklists (i.e., $CL_{1.1}$# 3 & $CL_{1.1}$#4 in Table 10, $CL_{2.1}$# 1 & $CL_{2.1}$#16 in Table 11). The participant voted yes for these items but argued that the inclusion of these items would depend upon the situation. In this regard, the participant provided an example of $CL_{2.1}$#16 in Table 11 and suggested that *"the team should have to be agreed to start regression testing in most cases. Still, there could be exceptions in this regard. If needed for the project, the product owner can decide on an early start"*.

Senior QA lead of C2 suggested the following items be included in checklist $CL_{3.1}$ (Table 12):

- Is the QA sign-off document ready?
- Do the stakeholders agree on QA sign-off?

The test manager of C8 suggested the following items be included in the checklist $CL_{2.1}$ (Table 11):

- Do we need to add new test cases in the regression suite?

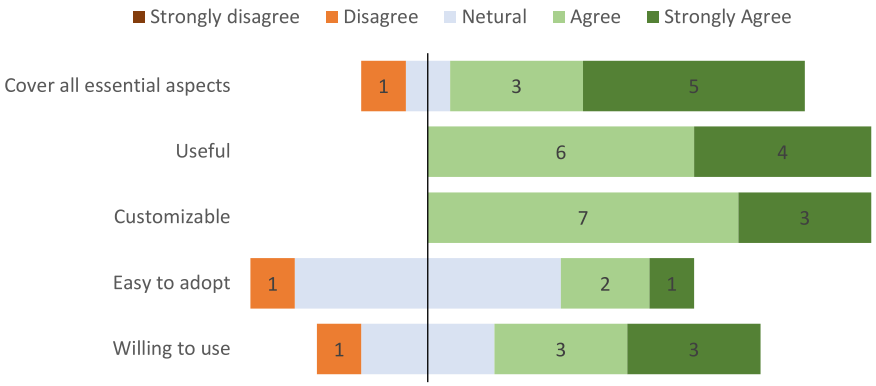

Fig. 4. Evaluation feedback from the participants on the final version of regression testing checklists.

- What is the trade-off between manual vs. automated testing?
- Have we discussed the scope of regression testing in the sprint planning meeting?

The QA unit head of C11 suggested the following two items to be included in the checklists.

- Are there any pending changes that will be deployed during the RT cycle? $CL_{2.1}$ (Table 11)
- Has the regression test report been consolidated and shared? $CL_{3.1}$ (Table 12)

We have reflected on these suggestions in Section 6 (discussion).

5.4. Checklists evaluation ***(RQ3)***

We evolved the checklists in two iterations since the practitioners approved most of the items included in the first version of the checklists except for a couple of cases mentioned in the preceding section. Therefore, we did not iterate the checklists further. We sent the second version (version 1.1) of the checklists and an evaluation questionnaire to the practitioners (see Appendix E in the supplementary data).

Fig. 4, summarizes the results of evaluation feedback. We received responses from 23 (out of 25) participants from 10 (out of 12) companies. The practitioners who participated as a group in the study's earlier phases provided us with their feedback as a group. Therefore we organized the results in a similar way. In Fig. 4, the responses represent the companies, not the individuals. Overall, the feedback was positive, as most participants agreed that the checklists are complete, useful, and customizable. Concerning willingness to use the checklists, a majority showed their willingness. However, there was a divided opinion about the ease of adopting the checklists in the companies. We expected this response because, during the interviews, many participants highlighted that even if they want to adopt the checklists or any other process improvement tool, they may get a negative response from the higher management of the companies.

6. Discussion

We conducted this study to support testing practitioners in structuring and improving the regression testing practice by introducing regression testing checklists. We opted for a multi-step co-design approach involving 25 practitioners from 12 companies in the first two phases and 23 from 10 companies in the last two phases of our study. Most participants represented large-size companies, and the average experience of the participants was 12 years. Therefore, we can say that the findings of this study represent the perspective of senior testing practitioners working in large-scale development environments. We started our process with workshops where we presented our research idea to the prospective participants and discussed the modalities of their participation in the study. In the subsequent steps, we built our understanding of the current state of regression testing practice in the participants' companies, investigated the regression testing activities considered essential by the participants, and investigated their opinion on regression testing checklists.

During the data collection phases, we observed that practitioners not only recognize the significance of checklists, but a few of them also use some form of regression testing checklists. However, they pointed out that their checklists are application-specific and cannot be generalized. They emphasized the need for checklists to help practitioners to keep track of essential regression testing activities.

6.1. Regression testing activities

Using a bottom-up approach, we identified the regression testing activities directly from the practitioners. While investigating the regression testing activities, the aim was to make the findings more representative. Therefore, we took input from the practitioners working in diverse environments and development domains. However, we did not see many variations in this regard as the practitioners working in different companies consider many of the identified regression testing activities equally important. This commonality allowed us to group similar activities under a single label, and we did so using thematic analysis. After identifying the regression testing activities, we classified the activities as "activities considered before regression testing" and "activities considered after regression testing". The activities identified in our study are familiar, as some are available in related studies. We have provided a few examples in Section 5 (Table 5). However, to our knowledge, we did not see a classification of regression testing activities from an applicability perspective.

6.2. Checklists creation and evolution

The checklists proposed in this study would help remind testing practitioners of the essential measures to be taken before and after regression testing. Since we evolved the checklists only for two iterations, we do not claim the comprehensiveness of the proposed checklists. These checklists provide a basis for structuring the regression testing process, and practitioners can improvise the checklist during its use. Practitioners working on domains other than those represented in this study can customize these checklists according to their needs.

During the evolution phase, a few checklist items received fewer recommendations. We highlighted these items in different colors (i.e., red, cyan, and gray). Most participants recommended excluding $CL_{1.1}$#6 from the final checklist (see Table 10, item highlighted in red), and their argument in this regard was that the question is irrelevant because regression testing is a team activity. Nevertheless, since some practitioners recommended this item, we leave it to the practitioners if they want to include $CL_{1.1}$#6 in the final checklist or not. We received suggestions from three participants concerning including a few items in the checklists, and we have presented these suggestions in the results. We did not enforce the inclusion of these items in the final checklists because we consider that these items add further detail to already existing items. For example, items suggested by the senior QA lead of C2 *"Is QA sign-off document ready?" and " Do the stakeholders agree on QA sign-off?"* are the further interpretation of $CL_{3.1}$#7 (Table 12) *"Do the test team members agree to test closure?"*. The items suggested by the test manager of C8 *" Do we need to add new test cases in the regression suite?"* could correspond to the checklist items $CL_{2.1}$#7, 8, & 9, *" What is the trade-off between manual vs automated testing?"* is similar to $CL_{3.1}$# 14, and *"Have we discussed the scope of regression testing in the sprint planning meeting?"* could correspond to items $CL_{2.1}$ # 10& 12 (Table 11). QA unit head of C11 suggested including two items, one in $CL_{2.1}$ *" Are there any pending changes that will be developed during the RT cycle?"* and the other in $CL_{3.1}$ *"Has the regression test report been consolidated and shared?"*. The item suggested for $CL_{2.1}$ is the further interpretation of item $CL_{2.1}$# 3. However, we consider the suggestion of including checklist item *"Has the regression test report been consolidated and shared?"* in $CL_{3.1}$ to be valuable, and we plan to add it to the checklist in future evaluations with more practitioners.

Furthermore, if the respective practitioners consider these items essential for their environment, they can add them to their local checklists.

6.3. Checklists evaluation

We opted for an opinion-based evaluation of the proposed checklists by the study participants. The practitioners' opinion was based on their experience in testing, a trial run of checklists, and discussion among the team members. Two practitioners from companies (C5 & C10), who participated in the study's initial phases, could not participate in the study's evaluation phase. In their feedback, 80% of the respondents think checklists are comprehensive, besides the fact that we only went through two iterations of checklists design and evolution. Considering the communication and cognitive gap between regression testing research and practice a reported fact (Engström et al., 2017; Lin, 2007), we were a little dubious if the proposed checklists are applicable in varying contexts of participating companies. However, the evaluations were affirmative as 100% of our respondents think that the proposed checklists are helpful in their team/organization context, and 100% responded that the checklists could be customized in their team/organization context.

We added a question to ask the participants if they would use these checklists. The participants from six companies showed a willingness to use the checklists on an experimental basis. We did not include the usage-based feedback of practitioners in the current study because practitioners could not give us a definite timeline for providing the usage results of checklists. However, they assured us they would send us their feedback once they completed at least one usage cycle of the checklists. We plan to publish the usage data of checklists and an improved version of the checklists in our future work.

6.4. Implications

This study has implications for regression testing research and practice. In the following, we briefly discuss the implications for regression testing practice and the implications for regression testing research.

6.4.1. Implications for practice

During our interactions with the practitioners for our various studies (e.g., Minhas et al. (2017), bin Ali et al. (2019), Minhas et al. (2020)), we observed that regression testing practice lacks documented structure. Most regression testing decisions are based on expert judgment, and activities are ad-hoc. The practitioners know this fact and realize the need to introduce structure in the regression testing activities.

The checklists proposed in this study are meant to help practitioners to keep track of regression testing activities. These are easy to adopt as the checklists' items represent activities considered essential by the practitioners for regression testing. The proposed checklists will remind practitioners not to miss an activity required for success. These simple checklists will aid the test managers in making necessary decisions concerning regression testing. For example, when to start and when to stop regression testing. Since the checklists are designed in collaboration with senior testing practitioners from varying contexts, therefore, these are scalable to the industry context. Using the feedback loop introduced for the design of the checklists, the practitioners can improvise the checklists by adding, removing, or updating the checklist items.

The proposed checklists will introduce a repeatable process at the team and organizational levels. Practitioners can reflect on the outcomes of adopted regression testing activities. Repetitive use of successful activities would enable practitioners to define and document the regression testing process according to their organizational context, which will be the ultimate step toward improving the regression testing process.

6.4.2. Implications for research

From the research perspective, the study has two kinds of implications (1) Specific implications for regression testing research and (2) Implications for empirical research.

Implications for regression testing research: Regression testing is a well-researched area, and many regression testing techniques have been proposed in the literature (Yoo and Harman, 2012; bin Ali et al., 2019). However, supporting regression testing practice in decision-making is an area overlooked by software engineering researchers. In this study, by incorporating the practitioners' perspectives, we have proposed checklists to support practitioners in decision-making. The study will open up new horizons for regression testing researchers. They can work to support regression testing practice, for example, test management-activities, supporting practitioners in essential regression testing activities, and improving the regression testing process.

Implications for empirical research: The challenging part of our study was to engage the practitioners for a longer period since we needed to involve them, from identifying regression testing activities to the final evaluation of the checklists. Our experience in this regard can be helpful to the software engineering researchers involved in empirical research. The following steps helped us engage practitioners through all phases of our study.

1. Introductory workshops. A practical approach to engaging practitioners in the studies is conducting introduction workshops and convincing them about the worth of the idea for practice.

2. Validate the findings. After the interpretation of the findings, getting validated by the participating practitioners will serve three purposes (i) it will increase the investigators' confidence in

the results, (ii) it will give a sense to practitioners concerning the significance of research for investigators, and (iii) it will increase practitioners' trust in the relevance of the results to their organizational context.

3. Keep them updated. Another way of keeping practitioners' engagement alive is to keep them updated about the progress and results.

4. Communicate the final results. After finalizing the results, communicate these to study participants. Also, discuss the plan of action with them. It will help for future engagements.

7. Conclusion

We conducted a multi-step co-design study to create and evolve regression testing checklists to help practitioners improve the regression testing process by keeping track of essential regression testing activities. Twenty-five practitioners from twelve companies participated in the first two phases of the study (i.e., until checklists creation). In the latter two phases (i.e., checklists evolution and evaluation), twenty-three practitioners from ten companies participated in the study.

As a result of **RQ1**, we identified regression testing activities considered essential in the companies. The identified activities provided a basis for the regression testing checklists.

In the next step, as a result of **RQ2**, we transformed the activities into the respective regression testing checklist. Two primary types for the checklists were finalized (i) two checklists to track the pre-regression testing activities and (ii) one checklist to track the post-regression testing activities. Later, we evolved the checklists based on the feedback of participating practitioners.

Finally, the same practitioners evaluated the checklists and provided us with their feedback after a trial run and discussions among their team members **RQ3**. The practitioner's feedback was positive about the various aspects of the checklists, except one, where we asked them *"Do you think checklists are easy to adopt in your organization's context?"*, 60% of the respondents chose to stay neutral. The reason for not taking a clear stance by the majority was the constraints of getting support from higher management. 60% of the respondent showed their willingness to use the checklists at the team level. This shows the practitioners found checklists helpful in improving their regression testing practice.

We do not include data about the usage of checklists because the checklists still need to be put into practice. However, one company has started adapting checklists to their environment, and two participating companies plan to use checklists for new releases. In the future, we plan to collect the usage data from the participants who are willingly using the checklists. Further, we aim to evaluate the checklists from more practitioners other than the ones who participated in this study. Based on this data, we will see the possibility of improving and generalizing the checklists.

CRediT authorship contribution statement

Nasir Mehmood Minhas: Conceptualization, Implementation, Writing – original draft. **Jürgen Börstler:** Conceptualization, Implementation, Writing – original draft. **Kai Petersen:** Conceptualization, Implementation, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We have provided all the data collection instrument along with the essential data in the appendices of the supplementary data

Acknowledgment

This work has in parts been supported by ELLIIT; the Swedish Strategic Research Area in IT and Mobile Communications. All authors approved the version of the manuscript to be published.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jss.2023.111697.

References

Ammann, Paul, Offutt, Jeff, 2016. Introduction to Software Testing. Cambridge University Press.

bin Ali, Nauman, Engström, Emelie, Taromirad, Masoumeh, Mousavi, Mohammad Reza, Minhas, Nasir Mehmood, Helgesson, Daniel, Kunze, Sebastian, Varshosaz, Mahsa, 2019. On the search for industry-relevant regression testing research. Empir. Softw. Eng. 1–36.

Böhm, Andreas, 2004. Theoretical Coding: Text Analysis in Grounded Theory. Sage London, p. 270.

Brito, Jardelane, Dias-Neto, Arilo Claudio, 2013. Conducting empirical studies to evaluate a technique to inspect software testing artifacts. CLEI Electron. J. 16 (1), 10.

Brykczynski, Bill, 1999. A survey of software inspection checklists. ACM SIGSOFT Softw. Eng. Notes 24 (1), 82.

Chaparro, Alex, Keebler, Joseph R., Lazzara, Elizabeth H., Diamond, Anastasia, 2019. Checklists: A review of their origins, benefits, and current uses as a cognitive aid in medicine. Ergon. Des. 27 (2), 21–26.

Cruzes, Daniela S., Dyba, Tore, 2011. Recommended steps for thematic synthesis in software engineering. In: Proceedings of the International Symposium on Empirical Software Engineering and Measurement. ESEM, pp. 275–284.

Dalal, Sandeep, Sudhir, Solanki, Kamna, 2018. Challenges of regression testing: A pragmatic perspective. Int. J. Adv. Res. Comput. Sci. 9 (1), 499–503.

Dybå, Tore, Dingsøyr, Torgeir, 2008. Strength of evidence in systematic reviews in software engineering. In: Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. pp. 178–187.

Engström, Emelie, Petersen, Kai, Ali, Nauman Bin, Bjarnason, Elizabeth, 2017. SERP-test: a taxonomy for supporting industry-academia communication. Softw. Qual. J. 25 (4), 1269–1305.

Engström, Emelie, Runeson, Per, 2010. A qualitative survey of regression testing practices. In: Proceedings of the International Conference on Product Focused Software Process Improvement. pp. 3–16.

Gawande, Atul, 2010. Checklist Manifesto, the (HB). Penguin Books India.

Hales, Brigette M., Pronovost, Peter J., 2006. The checklist—a tool for error management and performance improvement. J. Crit. Care 21 (3), 231–235.

Hales, Brigette, Terblanche, Marius, Fowler, Robert, Sibbald, William, 2008. Development of medical checklists for improved quality of patient care. Int. J. Qual. Health Care 20 (1), 22–30.

Harrold, Mary Jean, Orso, Alessandro, 2008. Retesting software during development and maintenance. In: Proceedings of the Frontiers of Software Maintenance Conference. pp. 99–108.

Heroux, Michael A., Willenbring, James M., 2009. Barely sufficient software engineering: 10 practices to improve your CSE software. In: 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering. IEEE, pp. 15–21.

Higgins, William Y., Boorman, Daniel J., 2016. An analysis of the effectiveness of checklists when combined with other processes, methods and tools to reduce risk in high hazard activities. Boeing Tech. J.

Host, Martin, Runeson, Per, 2007. Checklists for software engineering case study research. In: First International Symposium on Empirical Software Engineering and Measurement. ESEM 2007, IEEE, pp. 479–481.

Jafrin, Samia, Nandi, Dip, Mahmood, Sharfuddin, 2016. Test case prioritization based on fault dependency. Int. J. Mod. Educ. Comput. Sci. 8 (4), 33.

Kasoju, Abhinaya, Petersen, Kai, Mäntylä, Mika V., 2013. Analyzing an automotive testing process with evidence-based software engineering. Inf. Softw. Technol. 55 (7), 1237–1259.

Kitchenham, Barbara A., Brereton, O. Pearl, Budgen, David, Li, Zhi, 2009. An evaluation of quality checklist proposals-a participant-observer case study. In: 13th International Conference on Evaluation and Assessment in Software Engineering. EASE 13, pp. 1–10.
Kitchenham, Barbara A., Pfleeger, Shari Lawrence, 2002a. Principles of survey research: part 3: constructing a survey instrument. ACM SIGSOFT Softw. Eng. Notes 27 (2), 20–24.
Kitchenham, Barbara, Pfleeger, Shari Lawrence, 2002b. Principles of survey research: part 5: populations and samples. ACM SIGSOFT Softw. Eng. Notes 27 (5), 17–20.
Kitchenham, Barbara, Sjøberg, Dag I.K., Brereton, O. Pearl, Budgen, David, Dybå, Tore, Höst, Martin, Pfahl, Dietmar, Runeson, Per, 2010. Can we evaluate the quality of software engineering experiments? In: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. pp. 1–8.
Lacey, Anne, Luff, Donna, 2001. Qualitative Data Analysis. Trent focus Sheffield.
Lin, Xuan, 2007. Regression Testing in Research and Practice. Technical Report, Lincoln, NE, USA.
Lindgren, Eveliina, Münch, Jürgen, 2016. Raising the odds of success: the current state of experimentation in product development. Inf. Softw. Technol. 77, 80–91.
Madaio, Michael A., Stark, Luke, Wortman Vaughan, Jennifer, Wallach, Hanna, 2020. Co-designing checklists to understand organizational challenges and opportunities around fairness in Ai. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. pp. 1–14.
Minhas, Nasir Mehmood, Iqbal, Javed, 2011. Software process improvement practices–A Pakistani perspective. In: International Workshop on CMMI Based Software Process Improvement in Small and Medium Sized Enterprises. p. 29.
Minhas, Nasir Mehmood, Petersen, Kai, Ali, Nauman Bin, Wnuk, Krzysztof, 2017. Regression testing goals-view of practitioners and researchers. In: 2017 24th Asia-Pacific Software Engineering Conference Workshops. APSECW, IEEE, pp. 25–31.
Minhas, Nasir Mehmood, Petersen, Kai, Börstler, Jürgen, Wnuk, Krzysztof, 2020. Regression testing for large-scale embedded software development–Exploring the state of practice. Inf. Softw. Technol. 120, 106254.
Molléri, Jefferson Seide, Petersen, Kai, Mendes, Emilia, 2020. An empirically evaluated checklist for surveys in software engineering. Inf. Softw. Technol. 119, 106240.
Nayak, Soumen, Kumar, Chiranjeev, Tripathi, Sachin, 2016. Effectiveness of prioritization of test cases based on faults. In: 3rd International Conference on Recent Advances in Information Technology. RAIT, 2016, IEEE, pp. 657–662.
Perry, William E., 2007. Effective Methods for Software Testing: Includes Complete Guidelines, Checklists, and Templates. John Wiley & Sons.
Petersen, Kai, Carlson, Jan, Papatheocharous, Efi, Wnuk, Krzysztof, 2021. Context checklist for industrial software engineering research and practice. Comput. Stand. Interfaces 78, 103541.
Runeson, Per, Höst, Martin, 2009. Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. 14 (2), 131.
Thoring, Katja, Mueller, Roland, Badke-Schaub, Petra, 2020. Workshops as a research method: Guidelines for designing and evaluating artifacts through workshops.
Usman, Muhammad, Petersen, Kai, Börstler, Jürgen, Neto, Pedro Santos, 2018. Developing and using checklists to improve software effort estimation: A multi-case study. J. Syst. Softw. 146, 286–309.
Van de Schoot, Rens, Lugtig, Peter, Hox, Joop, 2012. A checklist for testing measurement invariance. Eur. J. Dev. Psychol. 9 (4), 486–492.
Wohlin, Claes, Runeson, Per, Höst, Martin, Ohlsson, Magnus C., Regnell, Björn, Wesslén, Anders, 2012. Experimentation in Software Engineering. Springer Science & Business Media.
Yoo, Shin, Harman, Mark, 2012. Regression testing minimization, selection and prioritization: a survey. Softw. Test. Verif. Reliab. 22 (2), 67–120.

Nasir Mehmood Minhas is a post-doctoral researcher who completed his Ph.D. in software engineering from the Software Engineering Research Lab (SERL), Blekinge Institute of Technology (BTH), Sweden. Currently, he is working on regression testing for embedded systems in close collaboration with the industry. Dr. Minhas has been working in academia since 2000 and has served in various institutions. His research interests include software testing, requirements engineering, and software process.

Jürgen Börstler is a professor of software engineering at the Department of Software Engineering at Blekinge Institute of Technology (BTH), Sweden. He received his Ph.D. from Aachen University of Technology (RWTH), Germany, in 1993. His research interests are in software process improvement, software quality and measurement, software readability and comprehension, and computer science education.

Kai Petersen is a professor of software engineering at Blekinge Institute of Technology (BTH), Sweden and University of Applied Sciences Flensburg, Germany. He received his Ph.D. from BTH in 2010. His research interests are Agile Software Development, Software Testing, Evidence-Based Software Engineering, and Software Measurement. His research has been conducted in close collaboration with companies and with an empirical focus.